

Wrocławskie Zakłady Elektroniczne **elwro**

Dokumentacja techniczno - ruchowa
maszyna cyfrowa odra 1204

Opis funkcjonalny

1204-I-1

Wrocław 1968

1. WSTĘP
2. ORGANIZACJA WEWNĘTRZNA MASZYNY
 - 2.1. SŁOWA MASZYNOWE
 - 2.1.1. STRUKTURA ROZKAZU
 - 2.1.2. REPREZENTACJA LICZB W MASZYNIE
 - 2.2. MODYFIKACJA ROZKAZU
 - 2.3. LISTA ROZKAZÓW
 - 2.3.1. ROZKAZY GRUPY 02 / $a_i = a \oplus n$ /
 - 2.3.2. ROZKAZY GRUPY 03 / $n_i = a \oplus n$ /
 - 2.3.3. ROZKAZY GRUPY 04 / $a_i = a \oplus N$ /
 - 2.3.4. ROZKAZY ZMIENNOPRZECINKOWE
 - 2.3.5. ROZKAZY DZIAŁAŃ STAŁOPRZECINKOWYCH DŁUGICH
 - 2.3.6. ROZKAZY LOGICZNE
 - 2.3.7. ROZKAZY DZIELENIA I MNOŻENIA STAŁOPRZECINKOWEGO
 - 2.3.8. ROZKAZY PRZESUNIĘĆ
 - 2.3.9. ROZKAZY NA B-REJESTRACH
 - 2.3.10. ROZKAZY STERUJĄCE
 - 2.3.11. ROZKAZY PRZYMUSOWEJ MODYFIKACJI
 - 2.3.12. ROZKAZY BEZADRESOWE
 - 2.3.13. ROZKAZY SZUKANIA DANYCH
 - 2.3.14. ROZKAZY PRZESUWANIA I ZAMIANY PÓL DANYCH
 - 2.3.15. ROZKAZY PROGRAMOWANE
 - 2.4. ORGANIZACJA PAMIĘCI OPERACYJNEJ
 - 2.4.1. PRZEADRESATOR
 - 2.4.2. REJESTRY INDEKSOWE
 - 2.4.3. WYKORZYSTANIE POCZĄTKOWEGO OBSZARU PAMIĘCI OPERACYJNEJ
 - 2.5. PRZERYWANIA PROGRAMOWE
 - 2.5.1. PRZERYWANIA KANAŁU OPERACYJNEGO
 - 2.5.2. PRZERYWANIA ARYTMOMETRU
 - 2.5.3. POWRÓT DO PRZERWANEGO PROGRAMU
 - 2.6. CZASOMIERZ
3. KANAŁ OPERACYJNY
 - 3.1. PULPIT OPERATORA
 - 3.2. ROZKAZY KANAŁU OPERACYJNEGO

- 4. ORGANIZACJA WSPÓLPRACY JEDNOSTKI CENTRALNEJ Z KANAŁAMI
- 4.1. KANAŁ PRZESYŁANIA ZNAKÓW
 - 4.1.1. ROZKAZY PRZYGOTOWUJĄCE REJESTRY KPZ
 - 4.1.2. ROZKAZY TESTOWANIA, PRZYGOTOWYWANIA I ZAINICJOWANIA PRACY URZĄDZEŃ KANAŁU PRZESYŁANIA ZNAKÓW
 - 4.1.3. BLOKOWE PRZEDSTAWIENIE FUNKCJI KPZ
- 4.2. KANAŁ PRZESYŁANIA SŁÓW
 - 4.2.1. ROZKAZY PRZYGOTOWUJĄCE REJESTRY KPS
 - 4.2.2. ZAINICJOWANIE PRACY KANAŁU I URZĄDZEŃ
 - 4.2.3. BLOKOWE PRZEDSTAWIENIE FUNKCJI KPS
- 4.3. OGÓLNA ORGANIZACJA MODUŁÓW URZĄDZEŃ ZEWNĘTRZNYCH
- 5. DODATKI

1. WSTĘP.

Maszyna Cyfrowa ODRA 1204 jest szybką i uniwersalną maszyną przeznaczoną zarówno do obliczeń naukowo-technicznych, jak i do sterowania w czasie rzeczywistym.

Oto krótka charakterystyka funkcjonalna maszyny:

- struktura maszyny bazuje na 24-bitowym słowie /plus 1 bit kontrolny/,
- wbudowane są rozkazy działań na liczbach 24 i 48-bitowych /przedstawionych w arytmetyce uzupełnieniowej/,
- jednoadresowa struktura rozkazu z możliwością:
 - modyfikacji indeksowej /3 rejestry indeksowe w każdym programie/,
 - modyfikacji pośredniej,
 - rekurencyjnej modyfikacji indeksowej i pośredniej,
- każda komórka lub rejestr programowy może być modyfikatorem,
- modyfikacja przez pojedynczą i podwójną zawartość modyfikatora,
- pełna automatyka działań zmiennoprzecinkowych,
- bogata grupa rozkazów szukania danych oraz rozkazów przesuwania i zamiany pól danych,
- 8 rozkazów programowanych w każdym programie, których treść dowolnie ustala programista,
- elastyczny system dołączania urządzeń wprowadzania i wyprowadzania informacji,
- możliwość dołączania do 7-iu kanałów przesyłania informacji /w standardowym wyposażeniu jeden kanał/,
- w każdym kanale można dołączyć do 8-iu urządzeń wprowadzania lub wyprowadzania informacji,
- pełna autonomia przesyłań informacji za pośrednictwem kanałów /zachodzące równocześnie z procesem obliczeń/ i przebiegające z maksymalną szybkością urządzeń zewnętrznych,
- pełna kontrola przesyłań,

- dowolność wyboru urządzeń wprowadzająco-wyprowadzających informacje oraz kodu przedstawienia informacji,
- wbudowany szybki priorytetowy system przerywania programu i wznowienia pracy programu,
- 4 poziomy przerywania programu w każdym kanale,
- wewnętrzne przerywania sygnalizujące ewentualne usterki programów lub złe funkcjonowanie maszyny,
- możliwość wykonania kilku programów jednocześnie z automatycznym zabezpieczeniem przed ich wzajemną interferencją,
- pamięć operacyjna o pojemności 16384 słów 24-bitowych /w specjalnym wykonaniu do 65536 słów/,
- arytmometr maszyny charakteryzuje się dużą prostotą strukturalną dzięki temu, że sterowanie wykonaniem operacji jest zapamiętane w specjalnej, bardzo szybkiej pamięci stałej.

2. ORGANIZACJA WEWNĘTRZNA MASZINY

Jednostka Centralna maszyny składa się z następujących zespołów funkcjonalnych:

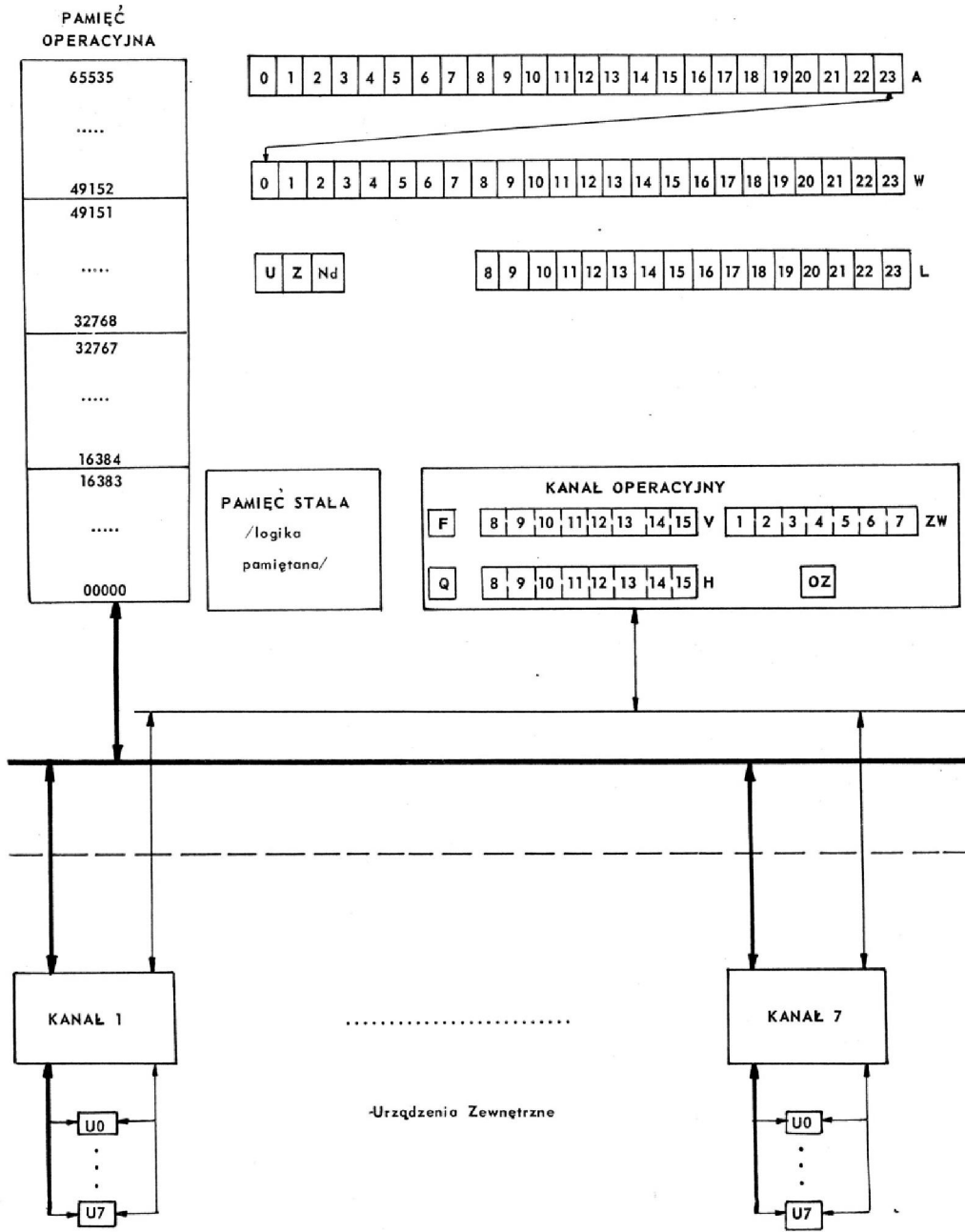
- **A r y t m o m e t r**, w którym są wykonywane operacje arytmetyczne i logiczne. W Arytmometrze wyróżnia się następujące rejestry: /patrz rys. 1/;
 - A - rejestr **A k u m u l a t o r a**, składający się z 24 pozycji binarnych, ponumerowanych od 0-23;
 - W - rejestr, będący prawym **W y d ł u ż e n i e m** Akumulatora, złożony także z 24 pozycji /od 0-23/;
 - L - rejestr **L i c z n i k a** rozkazów, złożony z 16 pozycji ponumerowanych od 8-23, w którym pamięta się adres kolejnego rozkazu do wykonania;
 - U, Z, Nd - jednobitowe rejestry dla zapamiętania pewnych dodatkowych informacji o wyniku operacji;
- **P a m i ę ć O p e r a c y j n a**, składająca się z 24-bitowych komórek /plus 1 bit techniczny/. Pamięć może składać się od 4096 do 65536 słów ponumerowanych jednolicie od 0 do 65535;
- **K a n a ł O p e r a c y j n y**, zawierający pewne rejestry pomocnicze:
 - V i H - złożone z 8 pozycji,
 - ZW - złożony z 7 pozycji,
 - F, Q, OZ - jednobitowe rejestry.

Funkcja rejestrów V i H oraz wskaźników F i Q jest związana ze **w z g l ę d n ą** adresacją Pamięci Operacyjnej oraz kontrolą działania programów /przed ich wzajemnym zniszczeniem/. Rejestr ZW oraz wskaźnik OZ związane są z przyjętym systemem **p r z e r y w a n i a** programu bieżącego.

Rola wyżej wymienionych rejestrów i wskaźników jest dokładnie wyjaśniona w dalszych punktach opisu.

P a m i ę ć S t a ł a, w której zapamiętane jest sterowanie wykonaniem operacji oraz pewne stałe informacje.

Jednostka Centralna współpracuje z Urządzeniami wprowadzania i wyprowadzania informacji za pośrednictwem **K a n a ł ó w**.



Rys. 1. JEDNOSTKA CENTRALNA I JEJ POWIĄZANIA Z KANAŁAMI.

Do Jednostki Centralnej można dołączyć 7 Kanałów, a do każdego Kanału 8 U r z ą d z e ń Z e w n ę t r z n y c h .

Jednoczesną pracę Kanałów i Arytmometru zapewnia tzw. K o o r d y n a t o r K a n a ł ó w .

2.1. SŁOWA MASZYNOWE.

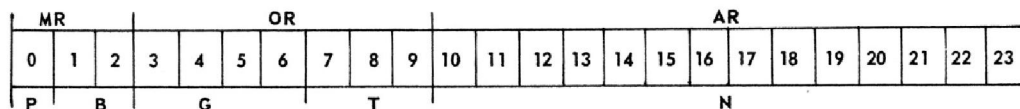
W organizacji rozpatruje się dwa rodzaje słów:

- s ł o w o k r ó t k i e , złożone z 24 bitów, zapamiętane w jednej komórce Pamięci Operacyjnej lub w jednym z rejestrów Arytmometru,
- s ł o w o d ł u g i e , złożone z dwóch kolejnych komórek Pamięci Operacyjnej lub dwóch rejestrów Arytmometru.
Słowo krótkie może przedstawiać r o z k a z lub l i c z b ę /stałoprzecinkową/, zaś słowo długie tylko liczbę /stało- lub zmiennoprzecinkową/.
W słowach zapamiętuje się także z n a k i alfanumeryczne /6 lub 8-bitowe/.

2.1.1. STRUKTURA ROZKAZU.

Rozkaz składa się z trzech części /rys. 2/:

- m o d y f i k a c y j n e j MR ze wskaźnikami:
P - wskaźnik modyfikacji p o ś r e d n i e j /1 bit/,
B - wskaźnik modyfikacji i n d e k s o w e j /2 bity/,
- o p e r a c y j n e j OR /7 bitów/, złożony z dwóch części G i T / g r u p a operacji i t y p operacji/,
- a d r e s o w e j AR /14 bitów/.



Rys. 2. Struktura rozkazu adresowego.

Rysunek 2 przedstawia strukturę rozkazu a d r e s o w e g o z 14-bitowym adresem p i e r w o t n y m .
Bity od 3 do 9 stanowią kod rozkazu /o numerach ósemkowych od 000 do 177/, który określa rodzaj operacji

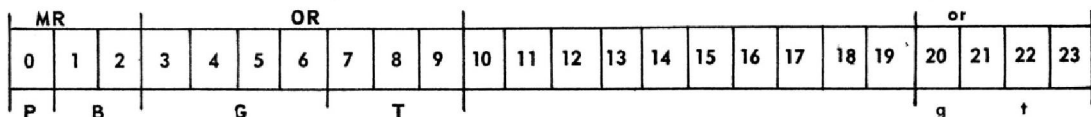
do wykonania.

Bity od 10 do 23 określają 14-bitowy adres pierwotny argumentu operacji /lub parametr/.

Bity od 1 do 2 określają numer rejestru indeksowego. Wartość 0 w tych bitach oznacza brak modyfikacji indeksowej, zaś wartość 1,2,3 oznacza modyfikację indeksową wg stanu komórki o tym numerze.

Jedynka w bicie zerowym oznacza modyfikację pośrednią, a 0 brak takiej modyfikacji.

Natomiast strukturę rozkazu bezadresowego przedstawia rys. 3. W tym rozkazie rozszerzono podstawowy kod operacji /G i T/ o dalsze dwie cyfry ósemkowe /g i t/.



Rys. 3. Struktura rozkazu bezadresowego.

2.1.2. REPREZENTACJA LICZB W MASZYNIE.

W maszynie rozróżnia się /ze względu na zrealizowane rozkazy/ następujące liczby:

- liczba stałoprzecinkowa krótka,
- liczba stałoprzecinkowa długa,
- liczba zmiennoprzecinkowa.

Liczba stałoprzecinkowa krótka, składa się z 24 pozycji binarnych, ponumerowanych od 0 do 23 /z lewej do prawej/.

Znak liczby zapisuje się w pozycji zerowej /jedynek oznacza ujemną liczbę, a zero dodatnią/.

Bity od 1 do 23 są bitami znaczącymi liczby, Dowolna liczba x przedstawiona jest w arytmetyce uzupełnieniowej:

$$x_{uz} = \begin{cases} x & \text{dla } x \geq 0 \\ 2^{n+1} + x, & \text{dla } x < 0, \text{ gdzie } n - \text{skala binarna.} \end{cases}$$

Liczba stałoprzecinkowa długa, złożona jest z 48 pozycji ponumerowanych od 0 do 47.

W pozycji 0 zapisuje się znak liczby, zaś w pozycjach od 1 do 47 bity znaczące liczby, /w kodzie uzupełnieniowym/. W Arytmometrze bity od 0 do 23 są kolejnymi bitami Akumulatora /od 0 do 23/, zaś bity od 24 do 47 to bity rejestru W /od 0 do 23/.

Liczba długa w Pamięci Operacyjnej składa się z dwóch komórek o adresach N-1 i N. W komórce N-1 znajdują się bity od 0 do 23, zaś w komórce N bity od 24 do 47 liczby długiej.

Liczba zmiennoprzecinkowa zajmuje, podobnie jak liczba stałoprzecinkowa długa, dwa rejestry Arytmometru /A i W/ lub dwie kolejne komórki Pamięci Operacyjnej o adresach N-1 i N. Liczba zmiennoprzecinkowa przedstawiona jest w maszynie przy pomocy pary liczb: m i c /m a n t y s a i c e c h a liczby/.

Wartość liczby x określa zależność:

$$x = m \cdot 2^c = /m; c/.$$

Mantysa m zajmuje 38 bitów /bity od 0 do 37/ słowa długiego, a cecha c 10 bitów /bity od 38 do 47/ słowa długiego. Mantysa i cecha są przedstawione także w arytmetyce uzupełnieniowej według wzoru:

$$m_{uz} = \begin{cases} m & \text{dla } m \geq 0, \\ 2 + m & \text{dla } m < 0, \end{cases}$$

$$c_{uz} = c + 512, \text{ dla } -511 \leq c \leq +511$$

Czyli mantysa przyjmuje wartości z przedziału od -1 do +1 - 2^{-37} , a cecha maszynowa /przesunięta/ od +1 do 1023.

Zero zmiennoprzecinkowe ma mantysę i cechę równą zeru, tzn. $m_{uz} = 0$ i $c_{uz} = 0$.

Z przyjętej struktury liczby zmiennoprzecinkowej wynika, że jej mantysa może składać się z 11 dziesiętnych cyfr znaczących, a jej wartość bezwzględna mieści się w przedziale od 10^{-153} do 10^{+153} .

2.2. MODYFIKACJA ROZKAZU.

Każdy rozkaz programu, pobrany do wykonania z Pamięci Opera-

cyjnej do Arytmometru, może być z m o d y f i k o w a n y przed wykonaniem. Modyfikacja rozkazu polega ogólnie mówiąc, na dodawaniu do jego części adresowej pojedynczej /lub podwójnej/ liczby zawartej we wskazanym rejestrze /lub komórce/. Tak otrzymaną nową część adresową nazywa się adresem r o b o c z y m rozkazu i oznacza się go także przez N. Rejestrami modyfikującymi mogą być:

- wyróżnione trzy komórki B1, B2, B3,
- dowolna komórka Pamięci Operacyjnej,
- rejestry programowe A i W.

B-modyfikacja zachodzi tylko wówczas, gdy wskaźnik B rozkazu modyfikowanego jest różny od zera.

B-modyfikacja polega na dodawaniu do adresu pierwotnego /14-bitowego/ lub do roboczego /16-bitowego/ 16 prawych bitów komórki /tj. bitów od 8 do 23/ Pamięci Operacyjnej o adresie określonym wskaźnikiem B rozkazu /tj. komórki 1,2 lub 3/. Czyli adres roboczy N równy jest

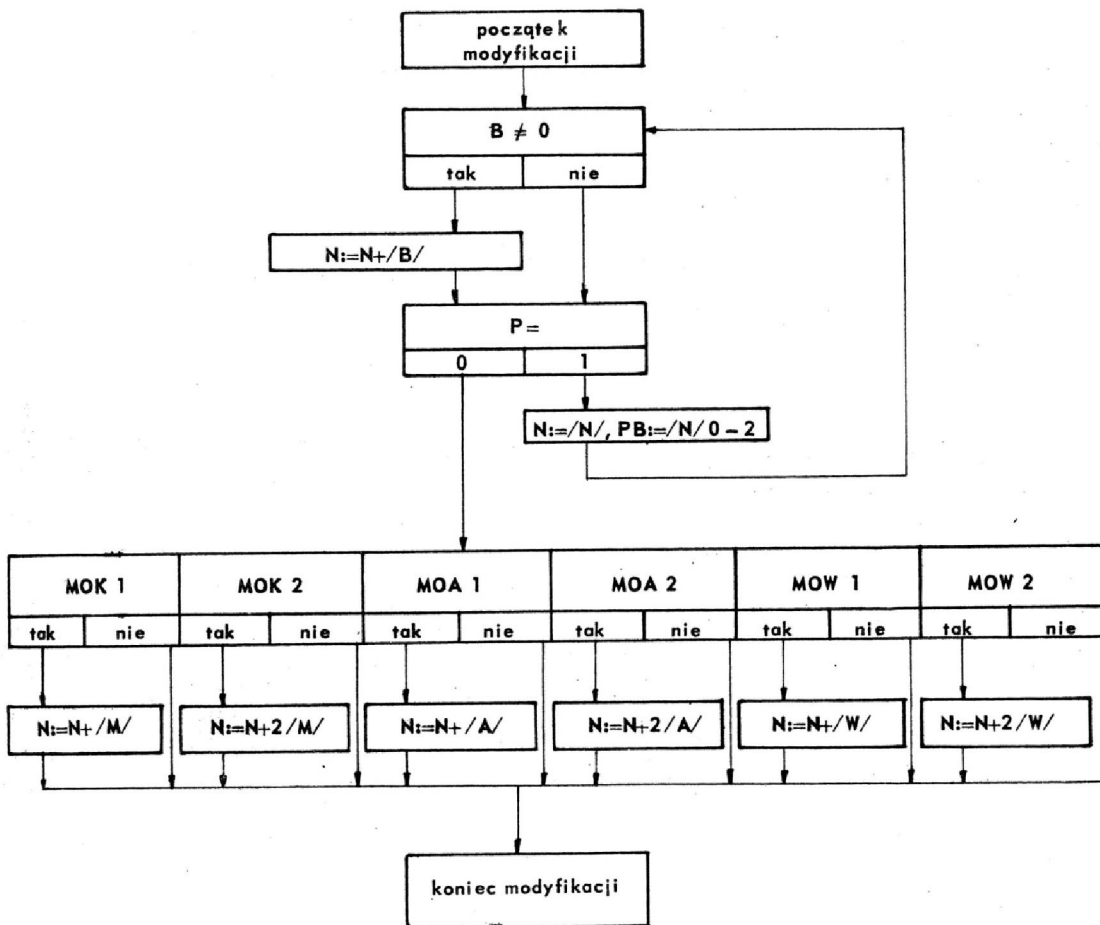
$N = N + B$, gdy B-modyfikacja zachodzi jako pierwsza, to występujące po prawej stronie równości N ma 14 bitów,

gdy ta modyfikacja zachodzi po innych modyfikacjach, to N ma 16 bitów.

Dokładny przebieg modyfikacji rozkazu pokazuje rys. 4.

P-modyfikacja, zachodzi tylko wówczas, gdy wskaźnik P rozkazu równa się 1.

P-modyfikacja polega na tym, że przyjmuje się za nowy adres roboczy liczbę zapisaną w komórce /tj. w bitach od 8 do 23/ określonej wartością adresu roboczego /dotychczas obliczonego/. Ponadto odnawia się wskaźniki P i B modyfikowanego rozkazu /w Arytmometrze!/. Dokładnie: nowy wskaźnik P przyjmuje wartość zerowej pozycji komórki określonej adresem roboczym, zaś wskaźnik B wartość pozycji 1-szej i 2-ej komórki.



N - adres rozkazu modyfikowanego
M - adres skuteczny rozkazów
MOK 1, MOK 2
/N/0-2⁻ oznacza wartość bitów
0-2 słowa o adresie N.
/ / - oznacza zawartość rejestru,
którego symbol podano w na-
wiasie.

Rys. 4. Modyfikacja rozkazu.

Czyli $N := /N/_{8-23}$ $P := /N/_0$ $B := /N/_{1-2}$

Kolejność wykonywania modyfikacji jest następująca:
B-modyfikacja,
P-modyfikacja,

B-modyfikacja i P-modyfikacja mogą być wykonane wielokrotnie przy ustalaniu adresu rozkazu /lecz nie więcej niż 6 razy/.

Przykład 1. Użycie B-modyfikacji.

Przypuśćmy, że chcemy wykonać rozkaz, który powoduje przepisanie stanu Komórki /określonej skutecznym adresem/ do Akumulatora. Rozkaz ten ma kod ósemkowy 020 i zapisany jest w naszym przykładzie w Komórkach 00100 i 00101 /wszystkie wielkości w tym przykładzie zapisane są oktalnie/.

Rozkaz zapisany w Komórce 00101 zawiera B-modyfikację przez B3, zaś rozkaz Komórki 00100 nie zawiera B-modyfikacji.

Fragment pamięci zawiera informacje:

<u>Adres komórki</u>	<u>Zawartość komórki Bity</u> 0-2 3-9 10-23	<u>Adres po modyfikacji</u>	<u>Efekt operacji</u>
00100	0 020 00105	00105	0 000 00012 → A
00101	3 020 00105	00105+/00003/=	
		=00105+00211=	
		=00316	0 000 05367 → A
00105	0 000 00012		
00003	0 000 00211		
00316	0 000 05367		

Przykład 2. Użycie B-modyfikacji i P-modyfikacji.

B-modyfikacja, która używa rejestr B2 oraz P-modyfikacja zadana jest przez cyfrę 6 w bitach 0-2 rozkazu.

Rozkaz z komórki 00200 zawiera tylko B-modyfikację przez rejestr B2, zaś rozkaz z komórki 00201 także i P-modyfikację.

<u>Adres komórki</u>	<u>Zawartość komórki Bity</u> 0-2 3-9 10-23	<u>Adres po modyfikacji</u>	<u>Efekt operacji</u>
00200	2 020 00500	00525	4 000 13007 → A
00201	6 020 00500	//00500+00025//=	
		= / 13007/ =	
		= <u>06035</u>	0 000 00701 → A
00002	0 000 00025		
00525	4 000 13007		
13007	0 000 06035		
06035	0 000 00701		

Modyfikacja przymusowa.

Jeśli omawiany rozkaz zostanie poprzedzony jednym z rozkazów: MOK1, MOK2, /patrz lista rozkazów/, to adres roboczy tego rozkazu powiększa się odpowiednio o pojedynczą lub podwójną zawartość komórki określonej adresem rozkazu MOK1, /MOK2/.
Jeśli zaś rozkaz zostanie poprzedzony jednym z rozkazów: MOA1, MOA2, MOW1, MOW2, to do adresu roboczego tego rozkazu dodaje się pojedynczą lub podwójną zawartość rejestru A lub W.
Modyfikacja przymusowa zachodzi jednokrotnie i po wykonaniu wszystkich modyfikacji typu B i P.
Adres roboczy otrzymany po wykonaniu wszystkich modyfikacji nazywa się adresem skutecznym i oznacza się go także literą N.

Przykład 3. Użycie modyfikacji przymusowej.

Przypuśćmy, że poznany wcześniej rozkaz o kodzie 020 jest poprzedzony rozkazem Modyfikacji przymusowej przez pojedynczą zawartość Komórki /MOK1/, który ma kod 105:

<u>Adres komórki</u>	<u>Zawartość komórki Bity</u>	<u>Adres po modyfikacji</u>	<u>Efekt operacji</u>
	0-2 3-9 10-23		
00300	0 105 00666	00666	
00301	0 020 01005	01005+/00666/=	
		= 01005+00011 =	
		= 01016	111125555 → A
00666	7 000 00011		
01016	1 111 25555		

Ograniczenia modyfikacji.

Ilość rozkazów typu MOK, MOA, MOW poprzedzających rozkaz modyfikowany oraz ilość P-modyfikacji jest ograniczona.

Mianowicie:

liczba P-modyfikacji modyfikowanego rozkazu + liczba rozkazów typu MOK, MOA, MOW poprzedzających modyfikowany rozkaz + liczba P-modyfikacji występujących w rozkazach MOK, MOA, MOW; musi być mniejsza od 7 /patrz poniższą uwagę/.

W przeciwnym przypadku wykonuje się rozkaz o kodzie 000 /BŁĄD/.

który powoduje przerwanie techniczne związane z komórką 63.

Uwaga: gdy sam modyfikowany rozkaz nie zawiera P-modyfikacji, to powyższa suma ograniczenia musi być mniejszą od 6.

2.3. LISTA ROZKAZÓW.

- Skróty użyte w liście rozkazów:

A - Akumulator /24 bity/,

W - Rejestr W /24 bity/,

AW - rejestr długi /48 bitów/ złożony z rejestrów A i W,

N - adres skuteczny rozkazu omawianego.

Małe litery: a,w,aw,n, oznaczają odpowiednio zawartości rejestrów A,W,AW i komórki o adresie skutecznym N /bądź komórek o adresach N-1 i N/.

Znak:= oznacza operację podstawiania. Dokładnie: zawartość rejestru, którego symbol napisano po lewej stronie znaku, staje się równa wartości wyrażenia stojącego po prawej stronie znaku.

- W maszynie istnieją trzy jednobitowe rejestry o nazwach: U,Z,Nd, w których pamięta się dodatkowe informacje o wyniku wykonanej operacji. Są to tzw. rejestry w a r u n k ó w. Do rejestru U /odpowiednio Z, Nd/ wpisuje się 1, gdy wynik wykonanej operacji jest Ujemny /odpowiednio równy Zero, z Nadmiarem/. W przeciwnym przypadku do rejestru U /odpowiednio Z, Nd/ wpisuje się 0.
- Wszystkie czynności związane z pobieraniem rozkazu z Pamięci Operacyjnej oraz jego wykonaniem tworzą tzw. c y k l r o z k a z o w y.
Cykl każdego rozkazu ma następujące etapy:
 - pobieranie rozkazu z Pamięci Operacyjnej, którego adres określa zawartość Licznika rozkazów,
 - ustalenie skutecznego adresu tego rozkazu przez wykonanie wszystkich zadanych modyfikacji /indeksowej, pośredniej i przymusowej/,
 - wykonanie pobranego rozkazu, tj. wykonania operacji określonej kodem tego rozkazu,

- powiększenie zawartości Licznika rozkazów o 1 /i przejście do kolejnego cyklu rozkazowego/.

W przypadku, gdy omawiany rozkaz jest rozkazem sterującym, to Licznik rozkazów może przyjąć dowolny stan /określony treścią tego rozkazu/.

Przy omawianiu listy rozkazów będziemy opisywali tylko trzeci etap cyklu rozkazowego w przypadku rozkazów niesterujących, a w przypadku rozkazów sterujących trzeci i czwarty etap cyklu.

Dwa pierwsze etapy cyklu rozkazowego przebiegają jednakowo dla wszystkich rozkazów.

- Wykonanie rozkazu powoduje na ogół zmianę stanu określonej komórki, rejestru lub wskaźnika. Należy przyjąć, że zachodzą tylko takie zmiany, jakie wynikają z opisu funkcji tego rozkazu.
- Siedmiobitowy podstawowy kod rozkazu podzielono na dwie części G i T /4 i 3 bity; patrz rys. 2/. Część G określa grupę rozkazową, zaś część T określa rodzaj rozkazu w danej grupie. Część T przyjmuje wartości ósemkowe od 0 do 7, a część G wartości od 00 do 17. Zatem podstawowy kod rozkazu może przyjmować wartości ósemkowe od 000 do 177. Części g i t rozkazu bezadresowego przyjmują wartości: g - 0 lub 1, t - 0 do 7. Pełna lista rozkazów, z kodami numerycznymi i skrótami literowymi, podana jest w DODATKU A.

3.1. ROZKAZY GRUPY 02 ($a:=a\oplus n$).

Rozkazy tej grupy charakteryzują się ogólnie tym, że powodują wykonanie operacji pomiędzy zawartością Akumulatora i Komórki o adresie skutecznym N.

Czyli argumentami tej operacji są słowa: a i n.

Wynik operacji umieszcza się w Akumulatorze.

Rozkazy te powodują wyzerowanie rejestrów warunków U, Z i Nd przed wykonaniem operacji, po operacji zaś wpisują do nich nową wartość zgodnie z relacjami wyniku operacji.

Opis poszczególnych rozkazów przebiega wg schematu:

- Skrót literowy rozkazu.
- Pełna nazwa rozkazu i kod ósemkowy rozkazu /podany w nawiasie/.
- Słowny opis funkcji rozkazu.
- Zapis symboliczny funkcji rozkazu.

USAK Umieść Stałoprzecinkowo w Akumulatorze Komórkę /020/.

- Rozkaz USAK o adresie skutecznym N powoduje umieszczenie 24-bitowej zawartości komórki o adresie N w Akumulatorze, zerując poprzedni jego stan. Ponadto zachodzą dodatkowe funkcje:
 - jeżeli nowy stan Akumulatora jest liczbą ujemną /jedynka w pozycji zerowej rejestru A/, to wpisuje się jedynkę do rejestru U, w przeciwnym wypadku wpisuje się 0,
 - jeżeli nowy stan Akumulatora jest zerem /zera we wszystkich pozycjach rejestru A/, to wpisuje się 1 do rejestru Z; w przeciwnym wypadku wpisuje się 0,
 - wpisuje się 0 do rejestru Nd
- $a:=n$; Jeśli $a < 0$, to $U:=1, Z:=0, Nd:=0$
 jeśli $a = 0$, to $U:=0, Z:=1, Nd:=0$
 jeśli $a > 0$ to $U:=0, Z:=0, Nd:=0$

U w a g a: Sposób przygotowania warunków U, Z, Nd, opisany przy tym rozkazie nazywa się schematem 1 i oznacza się go skrótem WAR /a,1/.

Skrót WAR /a,1/ czyta się - "warunki według stanu Akumulatora, schemat 1".

Skrót ten będzie używany przy dalszych opisach treści rozkazów maszyny.

SSAK Sumuj Stałoprzecinkowo Akumulator i Komórkę /021/.

- Rozkaz SSAK powoduje obliczenie sumy stałoprzecinkowej komórki o Adresie N i Akumulatora oraz umieszczenie tej sumy w Akumulatorze. Ponadto:
 - jeśli wynik operacji jest ujemny, to wpisuje się 1 do

rejestrze U; w przeciwnym przypadku wpisuje się 0 do rejestru U,

- jeśli wynik jest 0, to wpisuje się 1 do rejestru Z; w przeciwnym przypadku wpisuje się 0 do rejestru Z,
- jeśli wynik jest liczbą z nadmiarem, tj. liczbą nie należącą do przedziału $[-1, 1-2^{-23}]^*$, to wpisuje się 1 do rejestru Nd, jeśli zaś wynik operacji należy do tego przedziału, to wpisuje się 0 do rejestru Nd.

- $a := a+n$; jeśli $a < 0$, to $U:=1, Z:=0$,
jeśli $a = 0$, to $U:=0, Z:=1$,
jeśli $a > 0$, to $U:=0, Z:=0$,

jeśli wynik należy do przedziału $[-1, 1-2^{-23}]$, to $Nd:=0$

jeśli wynik nie należy do przedziału $[-1, 1-2^{-23}]$, to $Nd:=1$

* / Jest to przedział liczb stałoprzecinkowych krótkich występujących w zerowej skali binarnej /tj. w skali maszynowej/.

U w a g a: Sposób przygotowania warunków U, Z i Nd w zależności od wyniku operacji, opisany przy tym rozkazie, nazywa się schematem 2 i oznacza się go skrótem WAR /a,2/.
Skrót WAR /a,2/ czyta się - "warunki według wyniku pozostawionego w Akumulatorze, schemat 2".

OSAK Odejmij Stałoprzecinkowo od Akumulatora Komórkę /022/.

- Rozkaz OSAK powoduje odjęcie od Akumulatora zawartości komórki o adresie N. Przygotowanie warunków U, Z, Nd przebiega jak przy rozkazie SSAK.

- $a := a-n$; WAR /a,2/.

RSAK Oblicz różnicę Stałoprzecinkową Akumulatora i Komórki /023/.

- Rozkaz RSAK powoduje zmianę znaku Akumulatora i dodanie do niego zawartości komórki o adresie N.

- $a := -a+n$; WAR /a,2/

MLAK Pomnóż logicznie Akumulator przez Komórkę /024/.

- Rozkaz MLAK powoduje obliczenie iloczynu logicznego zawartości Akumulatora i Komórki. Wynik umieszcza się w Akumulatorze. Poszczególne bity w wyniku operacji przyjmują wartość w zależności od odpowiednich bitów argumentów tej operacji zgodnie z tablicą:

<u>/A/_i</u>	<u>/N/_i</u>	<u>/A/_i po operacji</u>
0	0	0
0	1	0
1	0	0
1	1	1

$i = 0, 1, \dots, 23$

Przykład: /binarna reprezentacja 24-bitowych słów/:

Przed operacją	Po operacji
/A/=001110...0	000110...0
/N/=010111...1	010111...1

- $a := a \wedge n$; WAR /a, 1/

ZMAK Zamień Akumulator z Komórką /025/.

- Rozkaz ZMAK powoduje zamianę zawartości Akumulatora i komórki o adresie N. Warunki ustala się według nowego stanu Akumulatora.

- $a := n$
 $n := a$; WAR /a, 1/

UAZK Umieść w Akumulatorze Zerując Komórkę /026/.

- Rozkaz UAZK powoduje przepisanie zawartości komórki o adresie N do Akumulatora, po czym wpisuje się zero do tej komórki:

- $a := n$,
 $n := 0$, WAR /a, 1/

PSAK Porównaj Stałoprzecinkowo Akumulator i Komórkę /027/.

- Liczby zapisane w Akumulatorze i w komórce są traktowane jako 24-bitowe liczby względne /znak w pozycji zerowej/ w arytmetyce uzupełnieniowej. Rozkaz PSAK powoduje względne porównanie zawartości Akumulatora i komórki o adresie N; po czym w zależności od relacji ustawia warunki U i Z. •
Dokładnie:
 - jeśli względna zawartość Akumulatora jest mniejsza od względnej zawartości komórki, to wpisuje się 1 do rejestru U, w przeciwnym przypadku wpisuje się 0 do rejestru U,
 - jeśli zawartości Akumulatora i komórki są identyczne, to wpisuje się 1 do rejestru Z, w przeciwnym przypadku wpisuje się 0 do rejestru Z.
- $a \leq n$; jeśli $a < n$, to $U:=1, Z:=0, Nd:=0$
 jeśli $a = n$, to $U:=0, Z:=1, Nd:=0$
 jeśli $a > n$, to $U:=0, Z:=0, Nd:=0$

Przykłady /binarna reprezentacja 24-bitowych słów/.

	<u>Przed i po operacji</u>	<u>Warunki U i Z po operacji</u>
I	/A/=00111...1 /N/=01011...1	U:=1, Z:=0
II	/A/=01000...0 /N/=11000...0	U:=0, Z:=0
III	/A/=10111...1 /N/=10111...1	U:=0, Z:=1

2.3.2. ROZKAZY GRUPY 03 ($n:=a \oplus n$).

Rozkazy tej grupy charakteryzują się ogólnie tym, że powodują wykonanie operacji pomiędzy zawartością Akumulatora i komórki o adresie skutecznym N /omawianego rozkazu/. Wynik operacji zapisuje się w komórce, a warunki U, Z, Nd opisują stan komórki po operacji.

PSKA Pamiętaj Stażoprzecinkowo w Komórce Akumulator /030/

- Rozkaz PSKA powoduje przepisanie zawartości Akumulatora do komórki o adresie N. Akumulator nie ulega zmianie. Ponadto:
 - jeśli nowy stan komórki jest liczbą ujemną, to $U:=1$;
w przeciwnym przypadku $U:=0$,
 - jeśli zaś nowy stan komórki jest zerem, to $Z:=1$;
w przeciwnym przypadku $Z:=0$,
 - wpisuje się 0 do rejestru Nd.
- $n:=a$; jeśli $n < 0$, to $U:=1, Z:=0, Nd:=0$
jeśli $n = 0$, to $U:=0, Z:=1, Nd:=0$
jeśli $n > 0$, to $U:=0, Z:=0, Nd:=0$

U w a g a: Sposób przygotowania warunków U, Z, Nd opisany przy tym rozkazie oznacza się skrótem WAR /n,1/.

SSKA Sumuj Stażoprzecinkowo w Komórce Akumulator /031/.

- Rozkaz SSKA powoduje obliczenie sumy stażoprzecinkowej komórki o adresie N i Akumulatora oraz umieszczenie tej sumy w Komórce. Ponadto:
 - jeśli wynik operacji jest ujemny, to wpisuje się 1 do rejestru U; w przeciwnym przypadku wpisuje się 0 do rejestru U,
 - jeśli wynik jest zerem, to wpisuje się 1 do rejestru Z; w przeciwnym przypadku wpisuje się 0 do rejestru Z,
 - jeśli wynik jest liczbą z nadmiarem, tj. liczbą nienależącą do przedziału $[-1, 1-2^{-23}]$, to wpisuje się 1 do rejestru Nd, jeśli zaś wynik operacji należy do tego przedziału, wtedy wpisuje się 0 do rejestru Nd.
- $n:=a+n$; jeśli $n < 0$, to $U:=1, Z:=0$
jeśli $n = 0$, to $U:=0, Z:=1$
jeśli $n > 0$, to $U:=0, Z:=0$

jeśli wynik należy do przedziału $[-1, 1-2^{-23}]$, to $Nd:=0$

jeśli wynik nie należy do przedziału $[-1, 1-2^{-23}]$, to $Nd:=1$

U w a g a: Sposób przygotowania warunków U, Z, Nd opisany przy tym rozkazie oznacza się skrótem WAR /n,2/.

OSKA Odejmij Stałoprzecinkowo Komórkę od Akumulatora /032/.

- Rozkaz OSKA powoduje odjęcie stałoprzecinkowe od zawartości Akumulatora stanu komórki o adresie N i zapamiętanie wyniku w komórce N.
- $n:=a-n$; WAR /n,2/

RSKA Oblicz Różnicę Stałoprzecinkową Komórki i Akumulatora /033/.

- Rozkaz RSKA powoduje odjęcie stałoprzecinkowe od komórki o adresie N zawartości Akumulatora.
- $n:=n-a$; WAR /n,2/

MLKA PoMnóż Logicznie Komórkę przez Akumulator /034/.

- Rozkaz MLKA powoduje obliczenie iloczynu logicznego zawartości komórki o adresie N i Akumulatora. Wynik umieszcza się w komórce.
- $n:=a \wedge n$; WAR /n,1/

DOKJ DOdaj do Komórki Jedynkę /035/.

- Rozkaz DOKJ powoduje dodanie do zawartości komórki o adresie N jedynki na pozycję 23 /czyli w skali binarnej 23/.
Jeśli przed dodaniem jedynki w 24-bitowej komórce była wartość binarna: 0111...1, to dodanie jedynki powoduje powstanie nadmiaru /Nd:=1/, a stan komórki po dodaniu ma postać: 1000...0; w przeciwnych wypadkach Nd:=0
- $n:=n+1$; WAR /n,2/.

PKZA Pamiętaj w Komórce Zerując Akumulator /036/.

- Rozkaz PKZA powoduje zapamiętanie zawartości Akumulatora w komórce o adresie N; po czym następuje wyzerowanie Akumulatora.
- $n:=a$,
 $a:=0$; WAR /n,1/

ODKJ Odejmij od Komórki Jedynekę /037/.

- Rozkaz ODKJ powoduje odjęcie od zawartości komórki o adresie N, jedynki na pozycji 23. Jeśli początkowy stan 24-bitowej komórki ma postać binarną: 1000...0, to odjęcie jedynki powoduje powstanie nadmiaru /Nd:=1/, a stan komórki po odjęciu jest w postaci: 0111...1.
- $n:=n-1$; WAR /n,2/.

2.3.3. ROZKAZY GRUPY 04 ($\alpha:=\alpha \oplus N$).

Większość rozkazów tej grupy powoduje wykonanie operacji pomiędzy zawartością Akumulatora i s t a łą określoną adresem skutecznym N tego rozkazu /dodatnia 16-bitowa wielkość/.

Wynik operacji zapisuje się w Akumulatorze. Warunki U, Z, Nd, opisują wynik operacji.

USAN Umieść Stałoprzecinkowo w Akumulatorze adres N /040/.

- Rozkaz USAN powoduje umieszczenie 16-bitowego adresu skutecznego N w Akumulatorze /w pozycjach 8-23 Akumulatora zaś w pozycjach 0-7 wpisuje się zera/.
- $a:=N$; WAR /a,1/.

SSAN Sumuj Stałoprzecinkowo Akumulator i adres N /041/.

- Rozkaz SSAN powoduje dodanie do zawartości Akumulatora 16-bitowego adresu skutecznego N.
- $a:=a+N$; WAR /a,2/.

OSAN Odejmij Stałoprzecinkowo od Akumulatora adres N /042/.

- Rozkaz OSAN powoduje odjęcie od Akumulatora 16-bitowego adresu skutecznego N.
- $a := a - N$; WAR /a,2/.

RSAN Oblicz Różnicę Stałoprzecinkową Akumulatora i adresu N /043/.

- Rozkaz RSAN powoduje odjęcie od adresu skutecznego N wartości Akumulatora i umieszczenie tej różnicy w Akumulatorze.
- $a := N - a$; WAR /a,2/.

MLAN Pomnóż Logicznie Akumulator przez adres N /044/.

- Rozkaz MLAN powoduje umieszczenie w Akumulatorze wyniku iloczynu logicznego Akumulatora i 16-bitowego adresu skutecznego N,
- $a := a \wedge N$; WAR /a,1/.

ZERK ZERuj Komórkę /045/.

- Rozkaz ZERK powoduje wyzerowanie komórki o adresie N. Stan warunków U, Z i Nd nie ulega zmianie w czasie wykonania tego rozkazu.
- $n := 0$.

PSKW Pamiętaj Stałoprzecinkowo w Komórce rejestr W /046/.

- Rozkaz PSKW powoduje przepisanie stanu rejestru W do komórki o adresie N. Warunki U, Z, Nd ustalają się z nowym stanem komórki.
- $n := w$; WAR /n,1/.

USWK Umieść Stałoprzecinkowo w rejestrze W Komórkę /047/.

- Rozkaz USWK powoduje przepisanie zawartości komórki o adresie N do rejestru W.

oraz dokonanie takiej korekty cechy liczby x , by nie została zmieniona wartość liczby zmiennoprzecinkowej.

Operacja zaokrąglenia logicznego mantysy, która następuje zawsze po normalizacji, polega na dodaniu w sposób logiczny /bez skutków "przeniesienia"/ jedynek /lub zera/ do pozycji 37 mantysy wyniku, gdy na pozycjach 38-47 mantysy wyniku jest przynajmniej jedna jedynka /są same zera/.

Operacje normalizacji i zaokrąglenia logicznego mantysy stosowane są do 48-bitowej mantysy "pośredniej" wyniku. Finalna mantysa wyniku ma 38 bitów i umieszczona zostaje w pozycjach od 0-37 Akumulatora zmiennoprzecinkowego, zaś finalna cecha wyniku /po korektach/ ma 10 bitów i ulokowana jest w pozycjach 38-47 Akumulatora zmiennoprzecinkowego.

Po rozkazach zmiennoprzecinkowych warunki U i Z opisują relację mantysy wyniku, a warunek Nd jest na ogół obojętny w tych operacjach.

Finalna cecha wyniku operacji zmiennoprzecinkowej może być większa od liczby 1023 i dlatego nie da się zapisać w 10 bitach.

Wówczas powstaje przerwanie /patrz punkt 2.5./, który informuje o tym fakcie.

UZAK Umieść Zmiennoprzecinkowo w Akumulatorze Komórki /050/.

- Rozkaz UZAK powoduje przepisanie zawartości komórek o adresach $N-1$ i N do Akumulatora zmiennoprzecinkowego.
Dokładnie: zawartość komórki $N-1$ do rejestru A , zaś zawartość komórki N do rejestru W .
Warunki U , Z , Nd według nowego stanu rejestru A . Normalizacja i zaokrąglenie nie zachodzą po tym rozkazie:

• $aw:=n$ lub $a:=/N-1/$, $w:=/N/$; $WAR /a,1/$

SZAK Sumuj Zmiennoprzecinkowo Akumulator i Komórki /051/.

- Rozkaz SZAK powoduje dodanie do zawartości Akumulatora zmiennoprzecinkowego zawartość zmiennoprzecinkową komórek o adresach $N-1$ i N . Ponadto wynik pośredni zostaje znor-

malizowany i zaokrąglony logicznie. Warunki po operacji opisują stan mantysy wyniku.

- $aw := aw + n$; WAR /a,2/.

OZAK Odejmij Zmiennoprzecinkowo od Akumulatora Komórki
/052/.

- Rozkaz OZAK powoduje odjęcie od Akumulatora zmiennoprzecinkowego zawartości komórek N-1 i N. Wynik zostaje ponadto znormalizowany i zaokrąglony logicznie.
- $aw := aw - n$; WAR /a,2/

RZAK Oblicz Różnicę Zmiennoprzecinkową Akumulatora
i Komórek /053/.

- Rozkaz RZAK oblicza różnicę zmiennoprzecinkową rejestru AW i komórek N-1 i N. Wynik znormalizowany i zaokrąglony logicznie pozostawia się w rejestrze AW.
- $aw := n - aw$; WAR /a,2/

PZKA Pamiętaj Zmiennoprzecinkowo w Komórkach Akumulator
/054/.

- Rozkaz PZKA powoduje przepisanie stanu rejestru AW do komórek N-1 i N. Operacje normalizacji i zaokrąglenia logicznego nie zachodzą.
- $n := aw$ lub $/N-1/ := a$, $/N/ := w$; WAR /a,1/

MZAK PoMnóż Zmiennoprzecinkowo Akumulator przez Komórki
/056/.

- Rozkaz MZAK powoduje pomnożenie zmiennoprzecinkowe rejestru AW przez zawartość komórek o adresach N-1 i N. Znormalizowany i zaokrąglony iloczyn pozostawia się w rejestrze AW.
- $aw := aw \cdot n$; WAR /a,2/

DZAK PoDziel Zmiennoprzecinkowo Akumulator przez Komórki
/057/.

- Rozkaz DZAK powoduje dzielenie zmiennoprzecinkowe rejestru AW przez zawartość komórek o adresach N-1 i N. Znormalizowany i zaokrąglony iloraz pozostawia się w rejestrze AW.

• aw:=aw/n ; WAR/a,2/

U w a g a: Rozkaz ten wykorzystuje zerową komórkę /BO/ programu jako roboczą. Zatem niszczy się poprzedni stan tej komórki.

3.5. ROZKAZY DZIAŁAŃ STAŁOPRZECINKOWYCH DŁUGICH.

Rozkazy tej grupy powodują wykonanie działań na liczbach stałoprzecinkowych długich. Jednym z argumentów jest rejestr długi AW /od 0 do 47/, a drugim zawartość komórek N-1 i N /komórka długa/. Warunki związane są z lewą częścią wyniku /bity od 0-23/ operacji długiej.

SDAN Sumuj Długo Akumulator i Adres N /060/.

- Rozkaz SDAN dodaje 16-bitowy adres skuteczny N do długiego stałoprzecinkowego Akumulatora /AW/ w skali 47. Oznacza to, że adres ten zostaje dodany do rejestru W z uwzględnieniem przeniesienia z rejestru W do A. Dodawanie długie powoduje nadmiar /Nd:=1/, gdy rezultat sumy jest większy od $1-2^{-47}$ /traktowany w zerowej skali binarnej/.

• aw:=aw+N ; WAR/a,2/

SDAK Sumuj Długo Akumulator i Komórkę /061/.

- Rozkaz SDAK dodaje do liczby stałoprzecinkowej długiej zapisanej w rejestrze AW liczbę stałoprzecinkową długą zapisaną w komórkach o adresach N-1 i N. Jeśli wynik nie należy do przedziału $[-1, 1-2^{-47}]$, to powstaje nadmiar /Nd:=1/.

- $aw:=aw+n$; lub $w:=w+/N/$ i $a:=a+/N-1/+C$; WAR/a,2/,
gdzie przez C oznaczono wartość przeniesienia binarnego
powstałego z sumy w $+/N/$.

ODAK Odejmij Długo od Akumulatora Komórkę /062/.

- Rozkaz ODAK powoduje odjęcie od liczby stałoprzecinkowej
długiej zapisanej w rejestrze AW, liczby stałoprzecin-
kowej długiej zapisanej w komórkach o adresach N-1 i N.
Jeśli wynik tej operacji nie należy do przedziału
 $/-1, 1-2^{-47}/$, to powstaje nadmiar /Nd:=1/.
- $aw:=aw-n$; lub $w:=w-/N/$ i $a:=a-/N-1/ + C$,
gdzie C oznacza przeniesienie w powstałe z różnicy
 $w-/N/$.

RDAK Oblicz Różnicę Długą Akumulatora i Komórki /063/.

- Rozkaz RDAK powoduje odjęcie od liczby stałoprzecinkowej
długiej zapisanej w komórkach o adresach N-1 i N, liczby
stałoprzecinkowej długiej zapisanej w rejestrze AW.
Wynik różnicy zapisuje się w rejestrze AW.
Jeśli wynik tej operacji nie należy do przedziału
 $/-1, 1-2^{-47}/$, to powstaje nadmiar /Nd:=1/.
- $aw:=n-aw$; lub $w:=/N/-w$ i $a:=/N-1/-a+C$; WAR/a,2/,
gdzie C oznacza przeniesienie z różnicy $/N/-w$.

2.3.6. ROZKAZY LOGICZNE.

Do grupy tej należą rozkazy:

- iloczynu logicznego /MLAK, MLKA, MLAN/,
- sumy logicznej /SLAK/,
- różnicy symetrycznej /OŁAK/.

Rozkazy iloczynu logicznego zostały wcześniej opisane.
Tutaj opiszemy rozkazy sumy i różnicy logicznej.

SLAK Sumuj Logicznie do Akumulatora Komórkę /064/:

- Rozkaz SLAK powoduje wyznaczenie sumy logicznej Akumula-
tora i komórki o adresie N. Wynik zapisuje się w A.

- $a := a \vee n$; WAR/a,1/
- poszczególne bity wyniku tej operacji przyjmują wartości zgodne z tabelą:

<u>/A/_i</u>	<u>/N/_i</u>	<u>/A/_i po operacji</u>
0	0	0
0	1	1
1	0	1
1	1	1

gdzie $i = 0, 1, \dots, 23$

OLAK Odejmij Logicznie od Akumulatora Komórkę /065/.

- Rozkaz OLAK oblicza logiczną różnicę symetryczną Akumulatora i komórki o adresie N. Wynik zapisuje się w A.
- $a := a \oplus n$; WAR/a,1/
- Poszczególne bity w wyniku tej operacji przyjmują wartość zgodnie z tablicą:

<u>/A/_i</u>	<u>/N/_i</u>	<u>/A/_i po operacji</u>
0	0	0
0	1	1
1	0	1
1	1	0

gdzie $i = 0, 1, \dots, 23$

Do grupy rozkazów logicznych zaliczamy także rozkaz porównania l o g i c z n e g o :

PLAK Porównaj Logicznie Akumulator i Komórkę /077/.

- Rozkaz PLAK służy do porównania 24-bitowej zawartości Akumulatora i 24-bitowej zawartości komórki o adresie N. Zawartości te są traktowane jako 24-bitowe ciągi binarne, gdzie wszystkie bity są kolejnymi bitami rozwinięcia /od 0-23/. Czyli bit 0 słowa jest traktowany jako bit rozwinięcia o najwyższej wartości /a nie jako pozycja

znakowa/, a bit 23 o najniższej wartości tego rozwinięcia. Rozkaz PLAK powoduje porównanie arytmetyczne tak traktowanych ciągów a i n , po czym w zależności od relacji porównania ustawia warunki U i Z . Dokładnie:

- jeśli wielkość a jest mniejsza od n ; tzn. jeśli w pierwszej pozycji /idąc od pozycji 0 do 23/, w której bity ciągu a i n są różne, jest 0 w ciągu a , to wpisyje się 1 do rejestru U .
 - jeśli wielkość a jest identyczna z wielkością n , to $Z:=1$,
 - jeśli wielkość a jest większa od n , to $U:=Z:=0$.
- $a \geq n$; jeśli $a < n$, to $U:=1, Z:=0, Nd:=0$
 jeśli $a = n$, to $U:=0, Z:=1, Nd:=0$
 jeśli $a > n$, to $U:=0, Z:=0, Nd:=0$.

Przykłady /binarna postać 24-bitowych słów/.

	<u>Przed i po operacji</u>	<u>Warunki U i Z po operacji</u>
I	/A/=010011...1 /N/=011111...1	U:=1, Z:=0
II	/A/=010011...1 /N/=111011...1	U:=1, Z:=0
III	/A/=101110...0 /N/=101110...0	U:=0, Z:=1
IV	/A/=100111...1 /N/=010111...1	U:=0, Z:=0
V	/A/=01000...01 /N/=01000...00	U:=0, Z:=0

2.3.7. ROZKAZY DZIELENIA I MNOŻENIA STAŁOPRZECINKOWEGO.

Rozkazy te działają na liczbach stałoprzecinkowych krótkich /tzn. zajmujących pozycje od 0 do 23 słowa/ lub na liczbach stałoprzecinkowych długich /tzn. zajmujących pozycje od 0 do 47 słowa długiego/. Argumentami tych rozkazów są wielko-

ci a i n lub aw i n, zaś wynik operacji zostawia się w Arytmometrze.

MSAK PoMnóż Stałoprzecinkowo Akumulator przez Komórkę /066/.

- Rozkaz MSAK powoduje pomnożenie stałoprzecinkowe zawartości Akumulatora i komórki o adresie N. Iloczyn /48-bitowy/ umieszcza się w rejestrze AW. Jeśli czynniki iloczynu a i n występują w skali binarnej S, to wynik jest w skali S+S+1. Zatem, jeśli czynniki są w skali 23 /skala liczb całkowitych/, to iloczyn jest w skali 47. Przyjęta skala iloczynu / przesunięta o 1/ pozwala na pomnożenie dwóch dowolnych wielkości /stałoprzecinkowych krótkich/ bez obawy powstania nadmiaru w wyniku operacji. Po rozkazie MSAK warunki ustalają się według części iloczynu zapisanego w A. Dokładnie:
 - jeśli iloczyn jest ujemny /jedynka w pozycji zerowej Akumulatora/, to U:=1, w przeciwnym przypadku U:=0,
 - jeśli pierwsze 24-bity iloczynu /czyli bity Akumulatora/ są zerami, to Z:=1, w przeciwnym przypadku Z:=0,
 - rejestr Nd przyjmuje 0.
- aw=a.n ; WAR/a,1/.

Przykłady mnożenia /postać binarna słów/:

	<u>Przed mnożeniem</u>	<u>Po mnożeniu</u>
I	/N/=000...0010 /A/=000...0011 /W/=XXX...XXXX	000...0010 000...0000 000...0110
	<u>U w a g a:</u> stan rejestru .W przed mnożeniem jest obojętny.	
II	/N/=010...0000 /A/=000...0011 /W/=XXX...XXXX	010...0000 000...0000 110...0000
III	/N/=100...0000 /A/=100...0000 /W/=XXX...XXXX	100...0000 010...0000 000...0000

DSAK PoDziel Stałoprzecinkowo Akumulator przez Komórkę /06

- Rozkaz DSAK powoduje dzielenie 48-bitowej zawartości rejestru AW przez zawartość komórki o adresy N. Iloraz /24-bitowy/ umieszcza się w akumulatorze. Jeśli 48-bitowa dzielna traktowana jest w skali 47 /a dzielnik w skali 23 to iloraz jest w skali 23 /47-1-23/.

Rozkaz DSAK zakłada, że dzielenie jest wykonalne.

Oznacza to, że zachodzą warunki:

1. $|aw| < n$, dla jednakowych znaków dzielnej i dzielnika,
2. $|aw| \leq n$, dla różnych znaków dzielnej i dzielnika.

Powyższe warunki wykonalności dzielenia 1 i 2 nie są sprawdzane technicznie przez maszynę, zaś dzielenie jest zawsze wykonane przez maszynę.

Po operacji dzielenia warunki opisują iloraz:

- jeśli iloraz jest ujemny, to $U:=1$; w przeciwnym przypadku $U:=0$,
 - jeśli iloraz jest równy zeru, to $Z:=1$; w przeciwnym przypadku $Z:=0$,
 - rejestr Nd przyjmuje 0.
- $a:=aw/n$; $WAR/a,1/$.

Reszta z dzielenia jest umieszczona w rejestrze W i ma ten sam znak jak dzielnik. Dokładna postać reszty z dzielenia jest następująca: reszta = dzielna - iloraz · dzielnik.

Przykłady dzielenia /postać binarna słów/:

	<u>Przed dzieleniem</u>	<u>Po dzieleniu</u>
I	/N/=0000...0010	0000...0010
	/A/=0000...0000	0000...0011
	/W/=0000...0110	0000...0000
II	/N/=0100...0000	0100...0000
	/A/=0001...1110	0111...1011
	/W/=1110...0001	0010...0001

DDAK PoDziel Długo Akumulator przez Komórkę /107/.

- Rozkaz DDAK powoduje dzielenie stałoprzecinkowe 48-bitowej zawartości rejestru AW oraz zawartość komórki o adresie N.

Długi iloraz /48-bitowy/ umieszcza się w rejestrze AW.

Warunki wykonalności dzielenia oraz stan rejestrów U,Z,Nd jak przy rozkazie DSAK.

- $aw:=aw/n$; $WAR/a,1/$

Reszta z dzielenia nie jest zachowana w maszynie.

Przykład dzielenia długiego:

<u>Przed dzieleniem</u>	<u>Po dzieleniu</u>
/N/=0100...0000	0100...0000
/A/=0001...1110	0111...1011
/W/=1110...0001	1000...0100

2.3.8. ROZKAZY PRZESUNIĘĆ.

Są to rozkazy powodujące przesuwania w prawo lub w lewo Akumulatora /Krótkiego lub Długiego/. Przesunięcia mogą być a r y t m e t y c z n e, n a t u r a l n e lub c y k l i c z n e. Przesuwanie w prawo lub w lewo 24-bitowego Akumulatora nazywa się k r ó t k i m, natomiast przesuwanie rejestru AW /traktowanego jako jeden rejestr/ nazywa się d ł u g i m. W czasie przesunięcia długiego w prawo bity z pozycji 23 Akumulatora przesuwa się do pozycji 0 rejestru W. Natomiast w czasie przesunięcia długiego w lewo bity z pozycji 0 rejestru W przesuwa się do pozycji 23 Akumulatora.

W procesie przesunięcia a r y t m e t y c z n e g o pozycja 0 Akumulatora nie ulega zmianie, natomiast w czasie przesunięcia n a t u r a l n e g o zmienia się, przyjmując wartość pozycji sąsiedniej /wartość pozycji pierwszej w przypadku przesuwania w lewo, bądź zero w przypadku przesuwania w prawo/. W czasie przesunięcia c y k l i c z n e g o w prawo Akumulatora bity z pozycji 23 przepisuje się do pozycji 0. W rozkazach przesuwania liczba przesunięć jest określona wartością adresu skutecznego N /przez 6 prawych

bitów adresu/. Liczba ta może przyjmować wartości dziesiętne od 0 do 63.

Rozkazy z liczbą przesunięć 0 nie wywołują żadnego skutku w rejestrach A i W, zaś wskaźniki U, Z opisują stan Akumulatora /wskaźnik Nd:=0/.

PAK Przesuń w Prawo Arytmetycznie Krótko /071/.

- Rozkaz PAK powoduje przesunięcie w prawo Akumulatora o N miejsc /należy przez to rozumieć liczbę zapisaną w bitach 18-23 adresu/. Pozycja zerowa Akumulatora nie ulega zmianie, zaś jej wartość zostaje kopiowana /do pozycji pierwszej/.

Warunki U i Z po tym rozkazie opisują stan przesuniętego Akumulatora, a warunek Nd=0.

- $a:=a \cdot 2^{-N}$; WAR/a,1/.

Przykłady przesunięcia arytmetycznego Akumulatora w prawo o 2 pozycje:

<u>Akumulator przed przesunięciem</u>	<u>Akumulator po przesunięciu</u>	<u>Warunki U,Z,Nd po przesunięciu</u>
I /A/=1000...0000	1110...0000	U:=1, Z=Nd=0
II /A/=0111...1101	0001...1111	U=Z=Nd=0

LNK Przesuń w Lewo Naturalnie Krótko /072/.

- Rozkaz LNK powoduje przesunięcie w lewo Akumulatora o N miejsc. Wszystkie pozycje Akumulatora ulegają przesunięciu, przy czym, jeśli w czasie przesunięcia wpisano do pozycji zerowej Akumulatora wartość różną od jej początkowej wartości, to powstaje nadmiar /Nd:=1/.

Warunki U i Z opisują stan przesuniętego Akumulatora:

- jeśli jest jedynka w pozycji zerowej Akumulatora, to U:=1; w przeciwnym przypadku U:=0,
- jeśli są zera we wszystkich pozycjach A /0 + 23/, to Z:=1, w przeciwnym przypadku Z:=0.

Wyżej opisany sposób przygotowania warunków U, Z i Nd notujemy: WAR/a,3/.

- $a:=a \cdot 2_N$; WAR/a,3/.

PNK Przesuń w Prawo Naturalnie Krótko /073/.

- Rozkaz PNK powoduje przesunięcie w prawo Akumulatora o N miejsc. W czasie przesunięcia wpisuje się zera do pozycji 0 Akumulatora, bity z pozycji 23 gubi się.
- $a := a \cdot 2_{-N}$; WAR/a,1/.

PAD Przesuń w Prawo Arytmetycznie Długo /074/.

- Rozkaz PAD powoduje przesunięcie arytmetyczne w prawo rejestru AW o N pozycji,
- $aw := aw \cdot 2^{-N}$; WAR/a,1/.

Przykłady przesunięcia arytmetycznego w prawo rejestru AW o 1 pozycję:

	Rejestry A i W przed przesunięciem	Rejestry A i W po przesunięciu	Warunki U,Z,Nd
I	/A/=0100...0011 /W/=0111...1001	0010...0001 1011...1100	U=Z=0, Nd=0
II	/A/=1000...0100 /W/=1011...1011	1100...0010 0101...1101	U=1, Z=Nd=0

LND Przesuń w Lewo Naturalnie Długo /075/.

- Rozkaz LND powoduje przesunięcie w lewo naturalnie zawartości rejestru AW. Warunki U,Z i Nd jak przy rozkazie LNK.
- $aw := aw \cdot 2_N$; WAR/a,3/.

Przykłady długiego przesunięcia naturalnego w lewo o 2 pozycje.

	Rejestry A i W przed przesunięciem	Rejestry A i W po przesunięciu	Warunki U,Z,ND
I	/A/=0010...0000 /W/=1011...1110	1000...0010 1111...1000	U=Nd=1, Z=0

II /A/=0100...0000 0000...0000 U=0, Z=Nd=1
 /W/=0010...0001 1000...0100

PCK Przesuń w Prawo Cyklicznie Krótko /076/.

- Rozkaz PCK powoduje przesunięcie w prawo cyklicznie zawartości Akumulatora o N miejsc.

Warunki U i Z opisują stan przesuniętego Akumulatora, zaś warunki Nd=0.

- $a := a * 2^N$; WAR/a,1/.

Przykłady działania rozkazu PCK z liczbą przesunięć 2.

	<u>Rejestr A</u> <u>przed przesunięciem</u>	<u>Rejestr A</u> <u>po przesunięciu</u>	<u>Warunki U,Z,Nd</u> <u>po przesunięciu</u>
I	/A/=0000...0010	1000...0000	U=1, Z=Nd=0
II	/A/=1000...0100	0010...0001	U=Z=Nd=0
III	/A/=0000...0111	1100...0001	U=1, Z=Nd=0

2.3.9. ROZKAZY NA B-REJESTRACH.

Rozkazy tej grupy pozwalają na:

- wpisywanie wartości skutecznego adresu N do dowolnego B - rejestru,
- przepisywanie zawartości komórki o adresie N do dowolnego B - rejestru,
- zapamiętywanie stanu dowolnego B - rejestru w komórce o adresie N,
- zamianę zawartości dwóch dowolnych B - rejestrów,
- powiększenie stanu dowolnego B - rejestru o dowolną stałą N.

W czasie wykonania tych rozkazów nie ulegają zmianie stany warunków U,Z,Nd.

UB1N Umieść w B1 adres N /115/.

- Rozkaz UB1N powoduje wpisanie 16-bitowego adresu skutecznego N do rejestru B1 /w pozycjach 8-23, zaś w pozycjach

0 do 7 wpisuje się zera/. Przy ustalaniu adresu skutecznego N tego rozkazu można używać dowolnych modyfikacji maszyny, zatem i B - modyfikacji.

- /B1/:=N

Przykłady użycia rozkazu UB1N /kod 115/.

W komórce 00400 znajduje się rozkaz o kodzie 115 bez żadnych modyfikacji, natomiast rozkaz z komórki 00401 zawiera B - modyfikację przez rejestr B3:

<u>Adres komórki</u>	<u>Zawartość komórki bity</u>	<u>Adres po modyfikacji</u>	<u>Efekt operacji</u>
	0-2 3-9 10-23		
00400	0 115 07777	07777	07777 → B1
00401	3 115 00000	00000+/00003/=	
		=01025	01025 → B1
00003	0 000 01025		

UB2N Umieść w B2 adres N /116/.

- Rozkaz UB2N powoduje wpisanie 16-bitowego adresu skutecznego N do rejestru B2.
- /B2/:=N.

UB3N Umieść w B3 adres N /117/.

- Rozkaz UB3N powoduje wpisanie 16-bitowego adresu skutecznego N do rejestru B3.
- /B3/:=N.

UB1K Umieść w B1 Komórkę /111/.

- Rozkaz UB1K powoduje przepisanie zawartości komórki o adresie N do rejestru B1.
- /B1/:=n.

UB2K Umieść w B2 Komórkę /112/.

- Rozkaz UB2K powoduje przepisanie zawartości komórki o adresie N do rejestru B2.
- /B2/:=n.

UB3K Umieść w B3 Komórkę /113/.

- Rozkaz UB3K powoduje przepisanie zawartości komórki o adresie N do rejestru B3.
- /B3/:=n.

PKB1 Pamiętaj w Komórce B1 /101/.

- Rozkaz PKB1 powoduje zapamiętanie zawartości rejestru B w komórce o adresie N.
- n:=/B1/.

PKB2 Pamiętaj w Komórce B2 /102/.

- Rozkaz PKB2 powoduje zapamiętanie zawartości rejestru B w komórce o adresie N.
- n:=/B2/.

PKB3 Pamiętaj w Komórce B3 /103/.

- Rozkaz PKB3 powoduje zapamiętanie zawartości rejestru B w komórce o adresie N.
- n:=/B3/.

2.3.10. ROZKAZY STERUJĄCE.

Rozkazów s t e r u j ą c y c h /warunkowych lub bezwarunkowych/ używa się do zmiany sekwencyjnej kolejności wykonania programu. Zmianę tę uzyskuje się przez wpisanie do Licznika rozkazów dowolnego stanu, określonego jednoznacznie treścią rozkazu sterującego. Rozkazy sterujące charakteryzują się także tym, że nie zmieniają stanu warunków U, Z i Nd.

SKB SKocz Bezwarunkowo /017/.

- Rozkaz SKB powoduje umieszczenie adresu skutecznego N /16-bitowego/ do Licznika rozkazów /L/.
Oznacza to, że sterowanie maszyny weźmie, w kolejnym cyklu rozkazowym, rozkaz z komórki o adresie N.
- $l:=N$

SKU SKocz przy U /014/.

- Rozkaz SKU powoduje wpisanie do Licznika rozkazów adresu skutecznego N, gdy warunek $U=1$; w przeciwnym przypadku /tzn., gdy $U=0$ / zachodzi normalna zmiana Licznika rozkazów /polegająca na powiększeniu jego stanu poprzedniego o 1/.
- $l:=N$, gdy $U=1$;
 $l:=l+1$, gdy $U=0$

SNU Skocz przy Nie U /013/.

- Rozkaz SNU powoduje wpisanie do L adresu N, gdy $U=0$; w przeciwnym przypadku zachodzi normalna zmiana Licznika rozkazów.
- $l:=N$, gdy $U=0$
 $l:=l+1$, gdy $U=1$.

SKZ SKocz przy Z /012/.

- Rozkaz SKZ powoduje skok, gdy warunek $Z=1$.
- $l:=N$, gdy $Z=1$
 $l:=l+1$, gdy $Z=0$

SNZ Skocz przy Nie Z /015/.

- Rozkaz SNZ powoduje skok, gdy warunek $Z=0$.
- $l:=N$, gdy $Z=0$;
 $l:=l+1$, gdy $Z=1$.

SKN SKocz przy Nd /010/.

- Rozkaz SKN powoduje skok, gdy $N_d=1$.
- $l:=N$, gdy $N_d=1$;
 $l:=l+1$, gdy $N_d=0$.

SKD SKocz przy Dodatniej wartości /011/.

- Rozkaz SKD powoduje skok, gdy $U=0$ oraz $Z=0$; natomiast skok nie zachodzi, gdy $U=1$ lub $Z=1$.
- $l:=N$, gdy $U=Z=0$
 $l:=l+1$, gdy $U=1$ lub $Z=1$.

SND Skocz przy Nie Dodatniej wartości /016/.

- Rozkaz SND powoduje skok, gdy $U=1$ lub $Z=1$.
- $l:=N$, gdy $U=1$ lub $Z=1$;
 $l:=l+1$, gdy $U=Z=0$.

SKP SKocz do Podprogramu /007/.

- Rozkaz SKP powoduje zapamiętanie stanu Licznika rozkazów plus jeden w komórce o adresie N ; po czym ładuje się $N+1$ do Licznika rozkazów /czyli następuje skok do rozkazu z komórki $N+1$ /.
Dokładnie: pamięta się wielkość $l+1$ w pozycjach od 8 do 23 komórki N ; zaś w pozycjach od 3 do 5 pamięta się kolejno wartości rejestrów U , Z i N_d ; a w pozostałych pozycjach tej komórki wpisuje się zera.
- $n:=l+1+WAR$
 $l:=N+1$

PLR Pamiętaj Licznik Rozkazów /002/.

- Rozkaz PLR zapamiętuje stan Licznika rozkazów plus dwa w komórce o adresie N .
- $n:=l+2+WAR$;
 $l:=l+1$.

SKocz i Wróć /004/.

Rozkaz SKW powoduje chwilowy skok celem wykonania rozkazu zapisanego w komórce o adresie N. Po wykonaniu rozkazu z komórki N powraca się do rozkazu następującego po rozkazie SKW.

Jeśli jednak rozkaz komórki N jest efektywnym rozkazem skokowym, to zostanie on wykonany w sposób normalny, /tzn. zmieni się stan licznika rozkazów LR/ i wówczas nie nastąpi powrót do początkowej sekwencji rozkazów. Rozkaz SKW sam nie zmienia wskaźników U, Z, Nd, lecz rozkaz z komórki N może zmienić te warunki /zgodnie z jego funkcją/.

Wykonaj rozkaz n.

SKS SKocz do Systemu /006/.

- Rozkaz SKS zapamiętuje stan Licznika rozkazów plus jeden /z warunkami/ w komórce o adresie rzeczywistym 16; po czym ładuje się do Licznika rozkazów zawartość komórki o adresie 16+N.
- $F:=0$, $/16/:=1+1+WAR$, $l:=/16+N/$.

U w a g a: wskaźnik $F=0$ pozwala na kontaktowanie się z komórkami Pamięci Operacyjnej, których adres rzeczywisty określa N /patrz pkt 2.4./.

PWR PoWRót z podprogramu /162/.

- Rozkaz PWR umieszcza zawartość komórki o adresie N w Liczniku rozkazów; po czym zeruje się komórkę N oraz wpisuje się 1 we wskaźniku F .
- $l:=/N/$, $/N/:=0$, $F:=1$.

STOP STOPuj maszynę /160/.

- Legalne pojawienie się rozkazu STOP powoduje przerwanie pracy maszyny. Przerwanie to następuje po wpisaniu adresu skutecznego N do Licznika rozkazów tak, że po ponownym uruchomieniu programu nastąpi skok do Komórki N ,
- $l:=N$ i STOP

U w a g a: rozkaz STOP jest uważany za n i e l e g a l-
n y, jeśli pojawi się, gdy $F=Q=1$ /patrz pkt 2.4./.

PGW PrzyGotuj Warunki /005/.

- Rozkaz PGW wpisuje:
 - jedynekę do warunku U , gdy zawartość komórki o adresie N jest ujemna; w przeciwnym przypadku wpisuje się zero do U ,
 - jedynekę do warunku Z , gdy zawartość komórki N jest równa zero; w przeciwnym przypadku wpisuje się zero do Z ,

- zero do Nd.

- U:=1 i Z:=0, gdy $n < 0$;
U:=0 i Z:=1, gdy $n = 0$;
U:=0 i Z:=0, gdy $n > 0$;
oraz Nd:=0.

U w a g a: rozkaz PGW tylko przygotowuje warunki według stanu komórki, lecz korzystanie z tych warunków odbywa się za pomocą rozkazów: SKU, SNU, SKZ, SNZ.

NIC nie rób /003/.

- Rozkaz NIC nie wywołuje żadnego skutku w maszynie. Proces obliczeń zostaje kontynuowany.

l:=l+1.

2.3.11. ROZKAZY PRZYMUSOWEJ MODYFIKACJI.

Rozkazy przymusowej modyfikacji powodują wykonanie dodatkowej modyfikacji na adresie następnego rozkazu. Modyfikację przymusową zadaje się przy pomocy jednego z rozkazów: MOK1, MOK2, MOA1, MOA2, MOW1, MOW2. Modyfikacja ta polega na dodaniu do części adresowej następnego rozkazu, pojedynczej lub podwójnej zawartości komórki /bity 8-23/ określonej skutecznym adresem N rozkazów MOK1 lub odpowiednio MOK2.

W ten sposób dowolna komórka Pamięci Operacyjnej może być zastosowana jako m o d y f i k a t o r.

Modyfikacja przymusowa zadana rozkazem MOA1 lub MOA2 /odpowiednio MOW1 lub MOW2/ polega na dodaniu do części adresowej następnego rozkazu /przed jego wykonaniem/, zawartości pojedynczej lub podwójnej Akumulatora /odpowiednio zawartości rejestru W/.

Rozkazy tej grupy są traktowane przez sterowanie maszyny jako część integralna rozkazu, który poprzedzają.

Dlatego po tych rozkazach wykonuje się zawsze i następny rozkaz. Oznacza to, że po wykonaniu cyklu rozkazowego,

rozkażu przymusowej modyfikacji, nie może zająć przerwy programu /patrz pkt 2.5/ lub stopowanie maszyny zadane klawiszem z pulpitu operatora /patrz pkt 3/. Jeśli dany rozkaz, podlegający przymusowej modyfikacji, ma także inne modyfikacje /tzn. B - modyfikacje lub P - modyfikacje/, to kolejność zastosowania tych modyfikacji jest następująca:

- wykonuje się wszystkie B - i P - modyfikacje, po czym
 - wykonuje się zadaną modyfikację przymusową /patrz rys.4/.
- Należy również podkreślić, że rozkazy przymusowej modyfikacji nie zmieniają stanu warunków U, Z i Nd.

MOK1 Modyfikuj przez Komórkę raz /105/.

- Zmodyfikuj rozkaz następny przez pojedynczą zawartość komórki o adresie skutecznym N.
- $N_n := N_n + N /$: gdzie N_n oznaczono adres następnego rozkazu /po ewentualnych B - i P - modyfikacjach/.

MOK2 Modyfikuj przez Komórkę dwa razy /106/.

- Zmodyfikuj rozkaz następny przez podwójną zawartość komórki o adresie N.
- $N_n := N_n + 2 / N /$.

MOA1 Modyfikuj przez A raz /00114/.

- Zmodyfikuj rozkaz następny przez pojedynczą zawartość Akumulatora.
- $N_n := N_n + A /$.

U w a g a: rozkaz MOA1 /podobnie jak i inne rozkazy bez-adresowe/ ma 5-cyfrowy ósemkowy kod; przy czym trzy pierwsze cyfry kodu /001/ to kod podsta w o w y rozkazu /część OR/, zaś ostatnie dwie cyfry /14/ kodu, to kod d o d a t k o w y rozkazu /część or/.

MOA2 Modyfikuj przez A dwa razy /00115/.

- Zmodyfikuj rozkaz następny przez podwójną zawartość Akumulatora.
- $N_n := N_n + 2/A/$.

MOW1 Modyfikuj przez W raz /00116/.

- Zmodyfikuj następny rozkaz przez pojedynczą zawartość rejestru W.
- $N_n := N_n + W/$.

MOW2 Modyfikuj przez W dwa razy /00117/.

- Zmodyfikuj następny rozkaz przez podwójną zawartość rejestru W.
- $N_n := N_n + 2/W/$.

2.3.12. ROZKAZY BEZADRESOWE.

Rozkazy tej grupy charakteryzują się tym, że posiadają ko część o p e r a c y j n ą, a nie posiadają części a d r e s o w e j.

5-cyfrowy kod rozkazu jest określony częścią OR rozkazu która przyjmuje wartość 001 dla rozkazów bezadresowych i częścią or rozkazu o wartościach od 00 do 17 /w omawianych rozkazach/. Niektóre z rozkazów bezadresowych zostały wcześniej omówione /np. MOA1, MOW1/; w tym paragrafie zajmujemy się pozostałymi rozkazami.

USAW Umieść Stałoprzecinkowo w Akumulatorze rejestr W /00110/.

- Rozkaz USAW powoduje przepisanie stanu rejestru W do Akumulatora; po czym ustawia warunki U i Z według nowego stanu Akumulatora /Nd:=0/.
- $a := w ; \text{ WAR}/a, 1/$.

PSWA Pamiętaj Stałoprzecinkowo w W Akumulator /00111/.

- Rozkaz PSWA przepisuje stan Akumulatora do rejestru W. Warunki według Akumulatora.
- $w:=a$; $WAR/a,1/$.

ZMAW Zamień A i W /00112/.

- Rozkaz ZMAW dokonuje zamian zawartości rejestru W i Akumulatora, po czym ustala warunki U i Z według nowego stanu Akumulatora $/Nd:=0/$.
- $w:=a$
 $a:=w$; $WAR/a,1/$.

ZEAW ZEruj A i W /00113/.

- Rozkaz ZEAW wpisuje zera w rejestrze W i Akumulatorze. Warunki U, Z i Nd nie ulegają zmianie.
- $a:=0$, $w:=0$.

MINA MINus Akumulator /00100/.

- Rozkaz MINA powoduje zmianę znaku Akumulatora zmiennoprzecinkowego /rejestry A i W/ z ewentualną korektą /w górę lub w dół/ jego cechy oraz normalizacją i zaokrągleniem mantysy. Warunki po operacji wg stanu A.
- $aw:=-aw$; $WAR/a,1/$.

Przykłady działania rozkazu MINA;

	<u>Rejestry A i W przed operacją</u>	<u>Rejestry A i W po operacji</u>
I	$/A/=0010...0000$ $/W/=0000...0011$	1000...0000 0000...0001
II	$/A/=1000...0000$ $/W/=0000...0101$	0100...0000 0000...0110

ABSA ABSolutna wartość Akumulatora /00101/.

- Rozkaz ABSA powoduje obliczenie bezwzględnej wartości Akumulatora zmiennoprzecinkowego, z ewentualną korektą cechy oraz normalizacją i zaokrągleniem jego mantysy.
- $aw := |aw|$; $WAR/a, 1/$.

ENTA Weź ENTier Akumulatora /00103/.

- Rozkaz ENTA oblicza część całkowitą liczby zmiennoprzecinkowej, zapisanej w rejestrze AW; po czym umieszcza się tę liczbę w rejestrze A w skali 23.
Jeśli liczba całkowita $E/aw/$ mieści się w przedziale $/-2^{23}, +2^{23}-1/$, to przeskakuje się następny rozkaz; jeśli zaś nie należy do tego przedziału, to wykonuje się rozkaz następny.

$a := E/aw/$; $WAR/a, 1/$;

$l := l+2$, gdy $-2^{23} \leq E/aw/ \leq 2^{23}-1$,

$l := l+1$, gdy $E/aw/ \geq 2^{23}$ lub $E/aw/ < -2^{23}$.

U w a g a 1: jeśli wielkość $E/aw/$ nie należy do przedziału $/-2^{23}, +2^{23}-1/$, to w rejestrze AW pozostawia się oryginalną liczbę zmiennoprzecinkową.

U w a g a 2: jeśli wielkość $E/aw/$ należy do przedziału $/-2^{23}, 2^{23}-1/$, to po rozkazie ENTA w rejestrze W znajduje się przypadkowa wielkość.

STND STaNDaryzacja /00104/.

- Rozkaz STND zamienia liczbę stałoprzecinkową długą /w skali 47/, zapisaną w rejestrze AW, na standardową postać zmiennoprzecinkową /znormalizowana i zaokrąglona/. Liczba zmiennoprzecinkowa zostaje także zapisana w rejestrze AW.
- $aw := S/aw/$; $WAR/a, 1/$.

DECA Mnożenie DECymalne A /00105/.

- Rozkaz DECA powoduje pomnożenie stałoprzecinkowego Akumulatora długiego /traktowanego w skali 47/ przez 10. Iloczyn pozostawia się także w rejestrze AW i w tej samej skali, tj. w skali 47.
- $aw:=aw \cdot 10$; $WAR/a,1/$.

U w a g a: rozkaz DECA wykonuje działania:
 $aw \cdot 2^3 + aw \cdot 2$

CZKL Czytaj Klawiaturę /00107/.

- Rozkaz CZKL powoduje przeczytanie 24-bitowej klawiatury /o klawiszach stabilnych/, znajdującej się na pulpicie operatora /patrz pkt 3.1/, do Akumulatora; po czym ustala warunki U i Z według nowego stanu A.
- $a:=KL$; $WAR/a,1/$.

ROZKAZY SZUKANIA DANYCH.

Rozkazy tej grupy pozwalają na szukanie wielkości 24-bitowej w określonym polu Pamięci Operacyjnej.

Postać szukanej wielkości /wzór/ znajduje się w rejestrze W, czyli szuka się w pewnym polu Pamięci Operacyjnej liczbę równą /lub mniejszą/ zawartości rejestru W. Przy szukaniu można przeglądać wszystkie kolejne komórki danego pola /tj. z k r o k i e m 1/ lub co drugie komórki /krok 2/, co trzecie komórki /krok 3/ itd.

Fakt znalezienia szukanej wielkości w danym polu jest sygnalizowany sposobem zmiany Licznika rozkazu przez rozpatrywany rozkaz szukania:

- przyjmuje się $l:=l+2$, gdy znaleziona została szukana wielkość lub
- przyjmuje się $l:=l+1$, gdy nie znaleziono szukanej wielkości.

Pole Pamięci Operacyjnej, w którym szukamy słowa spełniającego określoną relację /równy, mniejszy od/, jest zadane

jednolicie dla wszystkich rozkazów tej grupy. Mianowicie początek tego obszaru /adres/ jest określony przez zawartość rejestru B1, zaś koniec przez wartość adresu skutecznego N omawianego rozkazu szukania. Zatem komórki, których zawartość jest porównywana ze stanem rejestru W i z krokiem a, mają adresy:

$/B1/, /B1/+a, /B1/+2a, /B1/+3a, \dots, N.$

Przyjmujemy tutaj, że zmienna x reprezentuje kolejno zawartość komórek o powyższych adresach.

Ponadto przyjmujemy dodatkowe umowne oznaczenia:

- $\forall x=w$ - zastępuje zdanie: istnieje w rozpatrywanym polu pamięci taka wielkość /słowo/ x , że $x=w$
- $\forall x < w$ - zastępuje zdanie: istnieje w rozpatrywanym polu pamięci taka wielkość /słowo/ x , że $x < w$
- $\wedge x \neq w$ - zastępuje zdanie: wszystkie wielkości /słowa/ x , z rozpatrywanego pola pamięci, są różne od w ,
- $\wedge x \geq w$ - zastępuje zdanie: wszystkie wielkości /słowa/ x , z rozpatrywanego pola pamięci, są nie mniejsze od w .

Rozkazy szukania według relacji "mniejszy od" traktują porównywane słowa w dwojaki sposób:

- względny,
- logiczny.

W pierwszym przypadku zakłada się, że porównywane słowa są 24-bitowymi względnymi liczbami stałoprzecinkowymi /jak przy rozkazie PSAK/, zaś w drugim przypadku są tylko 24-bitowymi ciągami binarnymi, gdzie pozycja zerowa słowa nie jest pozycją znakową, lecz pozycją o najwyższej wartości tego ciągu /jak przy rozkazie PLAK/.

SZKR SZuKaj słowo Równe /130/.

- Rozkaz SZKR szuka w polu Pamięci Operacyjnej, o adresach brzegowych $/B1/$ i N , słowa równego zawartości rejestru W i z krokiem zapisanym w Akumulatorze $/w$ skali 23/. Przy

czym, jeśli w tym polu znajduje się szukane słowo, to sterowanie maszyny zapamiętuje jego adres w rejestrze B1 oraz przeskakuje następny rozkaz; w przeciwnym przypadku wykonuje się kolejny rozkaz programu.

- Jeśli $\vee x=w$, to $l:=l+2$ oraz $/B1/:=\text{adres } x$.
Jeśli $\wedge x \neq w$, to $l:=l+1$.
- W stosunku do rozkazu SZKR oraz rozkazów SZKM, SZMR, SZMM, SZKL, SZML stosuje się uwagi:

U w a g a 1: rejestry A i W nie ulegają zmianie w czasie wykonywania tego rozkazu.

U w a g a 2: po wykonaniu tego rozkazu warunki U i Z opisują stan Akumulatora $/Nd:=0/$.

U w a g a 3: proces szukania przebiega w kolejności rosnącej, tj. od adresu $/B1/$ do N, a kończy się po znalezieniu pierwszego słowa pola równego zawartości rejestru W.

U w a g a 4: jeśli adres N jest mniejszy od zawartości rejestru B1, to następuje przerwanie programu związane z komórką 63 /patrz punkt 2.5/.

SZKM SZUKaj słowo Mniejsze /131/.

- Rozkaz SZKM szuka w polu Pamięci Operacyjnej, o adresach brzegowych $/B1/$ i N, słowa mniejszego /ostro/ od zawartości rejestru W z krokiem zapisanym w A. Przy czym, jeśli w tym polu znajduje się szukane słowo, to sterowanie maszyny zapamiętuje adres tego słowa w rejestrze B1 oraz przeskakuje rozkaz następny; w przeciwnym przypadku wykonuje się kolejny rozkaz programu.
- Jeśli $\vee x < w$, to $l:=l+2$ oraz $/B1/:=\text{adres } x$.
Jeśli $\wedge x \geq w$, to $l:=l+1$.

SZMR SZukaj Maskując słowo Równe /132/.

- Rozkaz SZMR szuka w polu Pamięci Operacyjnej, o adresach brzegowych $/B1/$ i N, słowa równoważnego zawartości rejestru

ru W i z krokiem zapisanym w Akumulatorze /w skali 23/. Na wynik porównania słów x i w mają wpływ jedynie te pozycje słów, w których są jedynki w słowie będącym zawartością rejestru B2. Czyli zawartość rejestru B2 jest maską tego rozkazu. Przy czym, jeśli w tym polu znajduje się szukane słowo, to sterowanie maszyny zapamiętuje jego adres w rejestrze B1 oraz przeskakuje następny rozkaz; w przeciwnym przypadku wykonuje się kolejny rozkaz programu.

- Jeśli $\vee x=w$, to $l:=l+2$ oraz $/B1/:=\text{adres } x$.
Jeśli $\wedge x \neq w$, to $l:=l+1$.

SZMM Szukaj Maskując słowo Mniejsze /133/.

- Rozkaz SZMM szuka w polu Pamięci Operacyjnej, o adresach brzegowych $/B1/$ i N , słowa mniejszego /ostro/ od zawartości rejestru W z krokiem zapisanym w A. Na wynik porównania słów x i w mają wpływ jedynie te pozycje słów, w których są jedynki w słowie będącym zawartością rejestru B2. Przy czym, jeśli w tym polu znajduje się szukane słowo, to sterowanie maszyny zapamiętuje adres tego słowa w rejestrze B1 oraz przeskakuje rozkaz następny; w przeciwnym przypadku wykonuje się kolejny rozkaz programu.
- Jeśli $\vee x < w$, to $l:=l+2$ oraz $/B1/:=\text{adres } x$.
Jeśli $\wedge x \geq w$, to $l:=l+1$.

SZKL Szukaj Logicznie /136/.

- Rozkaz SZKL szuka w polu Pamięci Operacyjnej, o adresach brzegowych $/B1/$ i N , słowa mniejszego /ostro/ od zawartości rejestru W z krokiem zapisanym w A; przy czym, jeśli w tym polu znajduje się szukane słowo, to sterowanie maszyny zapamiętuje adres tego słowa w rejestrze B1 oraz przeskakuje rozkaz następny; w przeciwnym przypadku wykonuje się kolejny rozkaz programu.
- Jeśli $\vee x < w$, to $l:=l+2$ oraz $/B1/:=\text{adres } x$.
Jeśli $\wedge x \geq w$, to $l:=l+1$.

U w a g a: porównane słowa są traktowane jak przy rozkazie PLAK.

SZML SZukaj Maskując Logicznie /137/.

- Rozkaz SZML szuka w polu Pamięci Operacyjnej, o adresach brzegowych /B1/ i N, słowa mniejszego /ostro/ od zawartości rejestru W z krokiem zapisanym w A. Na wynik porównania słów x i w mają wpływ jedynie te pozycje słów, w których są jedyneki w słowie będącym zawartością rejestru B2.

Przy czym, jeśli w tym polu znajduje się szukane słowo, to sterowanie maszyny zapamiętuje adres tego słowa w rejestrze B1 oraz przeskakuje rozkaz następny; w przeciwnym przypadku wykonuje się kolejny rozkaz programu.

- Jeśli $\vee x < w$, to $l:=l+2$ oraz $/B1/:=\text{adres } x$
Jeśli $\wedge x \geq w$, to $l:=l+1$.

U w a g a: Zawartość rejestru W, po rozkazach SZMR /132/, SZMM /133/, i SZML /137/ ulega zmianie i równa się $W:=/B2/ \wedge W$.

2.3.14. **ROZKAZY PRZESUWANIA I ZMIANY PÓL DANYCH.**

Rozkazy tej grupy przepisują /przesuwają/ zawartość jednego pola Pamięci Operacyjnej na drugie pole /tej samej długości/ lub zamieniają zawartości dwóch różnych pól.

PSPS PrzeSuń Pole Słów /134/.

- Rozkaz PSPS przepisuje N słów z pola, którego początek określa zawartość rejestru B1, do pola, którego początek określa stan rejestru B3.

- $$\begin{aligned} & /B3/:=/B1/ \\ & /B3/+1:=/B1/+1 \\ & \dots\dots\dots \\ & /B3/+N-1:=/B1/+N-1 \end{aligned}$$

U w a g a: stan rejestrów A i W bez zmian.

U w a g a 2: warunki U i Z po operacji według A
/Nd:=0/.

U w a g a 3: gdy N=0, to omawiany rozkaz nie wywołuje
żadnego efektu.

ZMPS Zamień Pola Słów /135/.

- Rozkaz ZMPS zamienia zawartości pola, którego początek określa stan rejestru B1, z polem, którego początek określa stan rejestru B3. Długość pola pierwszego i drugiego jest określona adresem N.

/B3/:=:/B1/
/B3/+1:=:/B1/+1
.....
/B3/+N-1:=:/B1/+N-1

Do rozkazu ZMPS stosuje się uwagi z poprzedniego rozkazu.

2.3.15. ROZKAZY PROGRAMOWANE.

Są to rozkazy adresowe, których treść programową ustala programista dowolnie. Zaletą tych rozkazów wobec tradycyjnego sposobu wywoływania podprogramów jest to, że używa się ich w programie tak jak rozkazów posiadających realizację techniczną. W odniesieniu do adresu rozkazów programowanych stosuje się wszystkie modyfikacje /indeksową, pośrednią, przymusową/. Rozkazy programowane związane są technicznie z początkowymi komórkami obszaru programu. Dokładnie: z komórkami o numerach dziesiętnych od 4 do 13 względnego bloku zerowego programu /patrz punkt 2.4/.

Każdy program może używać 8 różnych typów rozkazów programowanych o kodach literowych: ROP0, ROP1, ROP2, ..., ROP7 /ogólnie ROPT/ i o kodach ósemkowych:

120, 121, 122, ..., 127 /ogólny kod 12T/.

Sterowanie maszyny "wykonuje" dowolny rozkaz programowany o kodzie 12T i adresie skutecznym N, w sposób następujący:

/4/:=N,
/5/:=1+1+WAR,
1 :=/6+T/.

2.4. ORGANIZACJA PAMIĘCI OPERACYJNEJ.

Pamięć Operacyjna składa się z 65536 komórek 24-bitowych, mających a d r e s y r z e c z y w i s t e od 0 do 65535. Organizacyjnie pamięć dzieli się na 256 b l o k ó w, każdy po 256 słów. Bloki mają n u m e r y rzeczywiste od 0 do 255. Numer bloku, do którego należy dana komórka, określony jest przez 8 lewych bitów adresu tej komórki /bity od 8 do 15/.

Programy maszyny ODRA 1204 są tak opracowane, że zajmują pewien obszar Pamięci Operacyjnej, którego komórki /i bloki/ mają kolejne w z g l ę d n e numery: 0,1,2,..., . Wielkość obszaru Pamięci Operacyjnej, zajętego przez program /wraz z danymi/ jest liczona z dokładnością do bloku. Programy z adresami względnymi funkcjonują sprawnie po wprowadzeniu ich w dowolny obszar Pamięci Operacyjnej.

W e w n ę t r z n a w z g l ę d n o ś ć adresów jest zapewniona dzięki tak zwanemu P r z e a d r e s a t o r o w i, który zamienia automatycznie numery względne bloków na rzeczywiste w czasie czytania /lub pisania/ z Pamięci Operacyjnej. Zamiana ta polega na zwiększaniu numerów względnych bloków o pewną stałą wielkość, będącą numerem rzeczywistym zerowego bloku programu.

P r z e a d r e s o w a n i e /powiększanie/ numerów bloków zachodzi tylko przy k o n t a k t a c h z Pamięcią Operacyjną /czytanie lub pisanie/.

Np. skuteczny adres biorący udział w działaniach Arytmometru jako argument operacji nie zostaje powiększony.

2.4.1. PRZEADRESATOR.

Przeadresator jest organizacyjnie częścią Kanału Operacyjnego i składa się z następujących rejestrów:

V-złożony z 8 pozycji, w którym przechowuje się numer rzeczywisty zerowego bloku programu czynnego,

H-złożony z 8 pozycji, w którym przechowuje się liczbę bloków programu czynnego,

F, Q - jednobitowe rejestry.

Rejestr V jest używany do przeadresowania numerów bloków programu. Proces przeadresowania numerów bloków zachodzi wówczas, gdy rejestry F i Q są równe 1 /niezależnie od postaci rozkazów/, w przeciwnym przypadku przeadresowanie nie zachodzi.

Zatem adres rzeczywisty R komórki, z którą kontaktuje się Arytmometr, ustala się następująco:

$$R = \begin{cases} N+V \cdot 2^8 & \text{gdy } F=Q=1 \\ N & \text{; gdy } F=0 \text{ lub } Q=0. \end{cases}$$

Gdzie N oznacza adres skuteczny argumentu operacji lub wartość Licznika rozkazów /w którym jest adres następnego rozkazu do wykonania/.

Wyżej opisany proces przeadresowania bloków nazywa się także V-modyfikacją.

Dotychczas przy opisywaniu listy rozkazów, mówiliśmy o adresie skutecznym omawianego rozkazu lub o adresie będącym zawartością Licznika rozkazów, nie zaznaczając bliżej, czy chodzi o adres wewnętrznie bezwzględny, czy też o adres wewnętrznie względny.

Przez adres w e w n ę t r z n i e b e z w z g l ę d n y należy rozumieć taki adres, który jest równy adresowi rzeczywistemu. Czyli są to adresy, które nie podlegają V-modyfikacji. Jeśli na przykład adres 00003 jest wewnętrznie bezwzględny, to adres ten odnosi się do komórki o adresie rzeczywistym /fizycznym/ równym 00003.

Przez adres w e w n ę t r z n i e w z g l ę d n y należy rozumieć taki adres, który odnosi się do komórki o adresie rzeczywistym, będącym sumą tego adresu względnego i zawartości rejestru V /dodanie w skali 15/. Jeśli zatem w rejestrze V jest np. liczba 4 /numer bloku/, to adres względny 00003 odnosi się do komórki o adresie rzeczywistym 02003 /oktalny zapis/. Innymi słowy jest to adres trzeciej komórki czwartego bloku. Jak już wspomniano V-modyfikacja zachodzi tylko wówczas, gdy we wskaźnikach F i Q jest jeden. Zatem adres skuteczny danego rozkazu może być wewnętrznie względny /gdy F=Q=1/ lub bezwzględny /gdy F=0 lub Q=0/.

Np. przy opisie treści rozkazów programowanych podajemy, że rozkazy te są związane z komórkami o numerach od 4 do 13, lecz numery te mogą być względne lub bezwzględne. Pełny zapis treści tych rozkazów ma postać:

$$\begin{aligned} /4+/V/ \cdot 2^8 / &:= N, \\ /5+/V/ \cdot 2^8 / &:= 1+1+WAR, \\ 1 &:= /6+T+/V/ \cdot 2^8 /, \quad \text{dla } F=Q=1; \end{aligned}$$

albo też:

$$\begin{aligned} /4/ &:= N, \\ /5/ &:= 1+1+WAR, \\ 1 &:= /6+T/, \quad \text{dla } F=0 \text{ lub } Q=0. \end{aligned}$$

Widać zatem, że można mieć 8 różnych rozkazów programowanych w każdym programie. Przyjmujemy tutaj oczywiście, że podczas pracy programów normalnych /użytkowych/ zachodzi warunek $F=Q=1$.

Natomiast dla programów organizacyjnych, jeden ze wskaźników F lub Q jest równy 0.

Wskaźniki F i Q równe 1 oraz rejestr V uniemożliwiają kontakt z komórkami bloków o niższych numerach rzeczywistych, niż stan rejestru V . Czyli niewłaściwe działanie danego programu nie może zniszczyć zapisu innych programów, z zapisanych przed nim. Natomiast w celu ochrony zapisów programów, ulokowanych w blokach o większych numerach rzeczywistych, niż program bieżący, /czynny/ wprowadzono rejestr H , w którym pamięta się liczbę bloków programu czynnego. Ostatnia kontrola /i ochrona/ przebiega następująco: Każda próba kontaktu z Pamięcią Operacyjną, gdy we wskaźnikach F i Q są jedynki, jest poprzedzona kontrolą, czy numer bloku adresu względnego /zapisany w bitach 8-15 tego adresu/ jest mniejszy /ostro!/ od numeru zapisanego w rejestrze H , czy też nie. Jeśli numer ten jest mniejszy od stanu rejestru H , to kontakt z Pamięcią Operacyjną zachodzi; w przeciwnym przypadku kontakt nie zachodzi, a sterowanie maszyny zapoczątkowuje przerwanie programowe /patrz punkt 2.5/.

2.4.2. REJESTRY INDEKSOWE.

Przez rejestry indeksowe danego programu tzw. B-rejestry, rozumie się trzy początkowe komórki zerowego bloku tego programu. Są to komórki względnego bloku zerowego o numerach: 1,2,3. Zawartości tych komórek wykorzystuje się w modyfikacji indeksowej rozkazu /patrz punkt 2.2./, a także do organizacji bardzo sprawnych pętli programowych /ze względu na wyróżnione rozkazy dotyczące B-rejestrów/.

2.4.3. WYKORZYSTANIE POCZĄTKOWEGO OBSZARU PAMIĘCI OPERACYJNEJ.

Blok Pamięci Operacyjnej o numerze rzeczywistym 0 jest wyróżniony w organizacji maszyny. Blok ten został związany z systemem przerwania programowych /wewnętrznych i zewnętrznych/ w maszynie. Komórki tego bloku zostały wykorzystane w sposób następujący:

- komórki od 1 do 3 są rejestrami indeksowymi /programu organizacyjnego/,
- komórki od 4 do 13 są wykorzystane przez rozkazy programowe /patrz punkt 2.3.13/ programu organizacyjnego,
- komórki od 16 do 59 są użyte jako wejścia dla wewnętrznych przerywań programowych, spowodowanych rozkazem SKS /patrz punkt 2.3.10/,
- komórki od 60 do 63 są związane z przerywaniami błędów, wykrytych przez Arytmometr /patrz punkt 2.5.2/,
- komórki od 64 do 95 są związane przez przerywanie zewnętrzne, spowodowane sygnałami nadawanymi przez Kanały /patrz punkt 2.5/.

Pozostałe komórki bezwzględnego bloku zerowego nie są wyróżnione przez organizację maszyny i dlatego mogą być użyte jako normalne komórki programowe.

2.5. PRZERYWANIA PROGRAMOWE.

Program pracujący w maszynie może być przerywany. Przerywania programowe spowodowane są sygnałami, które mogą

pojawić się w dowolnym momencie pracy Urządzeń Zewnętrznych i Jednostki Centralnej. Przerwanie programu polega, ogólnie mówiąc, na zawieszeniu działalności programu aktualnie pracującego i powołaniu do pracy innego programu /związanego z sygnałem, który spowodował dane przerwanie/ oraz zapamiętaniu miejsca przerwania.

Sygnały przerywające programy są zarejestrowane w specjalnych rejestrach Kanałów. Każdy Kanał może przekazywać do 4 sygnałów przerywających program. Zatem do Jednostki Centralnej przychodzą 32 sygnały zgrupowane w 8 grupach /Kanałów ponumerowanych od 0 do 7/. Z punktu widzenia Jednostki Centralnej obowiązuje kolejność priorytetowa przyjęcia sygnałów, według numerów Kanałów, do których należą zgłoszone sygnały. Zgłoszone sygnały Kanałów o niższych numerach są załatwiane w pierwszej kolejności. Kolejność przyjęcia sygnałów jest również programowo uregulowana za pomocą specjalnej maski. Z przyjęciem sygnałów przerywania związane są rejestry ZW i OZ, które organizacyjnie należą do Kanału Operacyjnego /Kanał 0/.

Rejestr ZW jest 7-bitowym rejestrem, zaś OZ - jednobitowym. Poszczególne pozycje rejestru ZW /ponumerowane od 1-7/ zezwalają /1/ lub zakazują /zero/ załatwienia zgłoszenia związanego z Kanałem o tym samym numerze.

Zatem Przerwanie Programu zachodzi wówczas, jeśli:

- istnieje zgłoszenie z Kanału,
- w odpowiedniej pozycji rejestru ZW jest jedynka,
- w rejestrze OZ jest jedynka.

Stany rejestrów OZ i ZW są ustawiane programowo.

Poniżej podaje się funkcje, jakie zachodzą w przypadku Przerwania Programu:

Q:=OZ:=0,
p:=a,
p+1:=w,
p+2:=l+1 /lub p+2:=N/,
l:=p+3,
ZG:=0.

Gdzie przez p oznaczono adres przerwania, określony zawartością komórki o adresie dziesiętnym $64+4K+z$ / K -numer Kanału, z którego pochodzi zgłoszenie; z - numer zgłoszenia w danym Kanale/.

Przez ZG oznaczono rejestr, w którym zarejestrowane jest omawiane /załatwiane/ zgłoszenie.

Zatem w komórkach o adresach bezwzględnych od 64 do 95 pamięta się adresy początkowe obszarów pamięci /4-komórkowe/, w których są pamiętane kolejno:

- zawartość Akumulatora,
- zawartość rejestru W,
- adres następnego rozkazu do wykonania /w pozycjach od 8 do 23 /oraz stan wskaźników U, Z, Nd /dokładnie: U w 3-ej pozycji komórki, Z w 4-ej, Nd w 5-ej, w pozostałych pozycjach wpisuje się zero/,
- pierwszy rozkaz do wykonania po przerwaniu.

Ponadto zeruje się wskaźniki OZ i Q, a także załatwioną pozycję rejestru ZGłoszeń.

2.5.1. PRZERYWANIA KANAŁU OPERACYJNEGO.

Kanał Operacyjny przerywa bieżący program, gdy wskaźnik OZ jest równy jedynce, jeśli wystąpi jeden z następujących powodów:

- próba zapisu /lub odczytu/ do komórki nie należącej do obszaru danego programu. Dokładnie: gdy numer względny bloku tej komórki jest większy lub równy zawartości rejestru H; przerwanie to ma numer 3 w tym Kanale i związane jest z komórką 67;
- rejestr c z a s o m i e r z a programowego zgłosił przerwanie /patrz punkt 2.6/; przerwanie to związane jest z komórką 66;
- na pulpicie operatora naciśnięto klawisz PP /przerywanie programu/; przerywanie to związane jest z komórką 65.

2.5.2. PRZERYWANIA ARYTMOMETRU.

Arytmometr maszyny może wykryć niektóre błędy programu. Błędy te są sygnalizowane w formie przerywań programowych. Arytmometr przerywa program, jeśli:

- w czasie realizacji rozkazów zmiennoprzecinkowych powstanie nadmiar /tzn. cecha wyniku jest większa od 1023/. Przerwanie to związane jest z komórką 62;
- w Arytmometrze pojawił się rozkaz do wykonania z niedozwolonym kodem operacji. Kodami niedozwolonymi są kody rozkazów zarezerwowanych /do przyszłego użytku/ oraz kod BŁĘDU /000/. Przerwanie to związane jest z komórką 63. Zainicjowane jest ono również wówczas, gdy pojawi się nielegalny rozkaz. Przez pojawienie się nielegalnego rozkazu, rozumie się próbę wykonania jednego z rozkazów:
 - organizacyjnych STOP, UKON, DOKC;
 - wejścia-wyjścia WYB, TES, PGT, WYK, UKN, UKK;w czasie, gdy we wskaźnikach F i Q są jedyńki.

2.5.3. POWRÓT DO PRZERWANEGO PROGRAMU.

Wznowienie pracy przerywanego programu odbywa się za pomocą specjalnego rozkazu o kodzie 151 /skrót literowy: OSA/. Rozkaz ten realizuje funkcje przeciwne niż przerywanie programowe. Dokładnie:

a:=/N/,
w:=/N+1/,
l:=/N+2/,
OZ:=1, Q:=1.

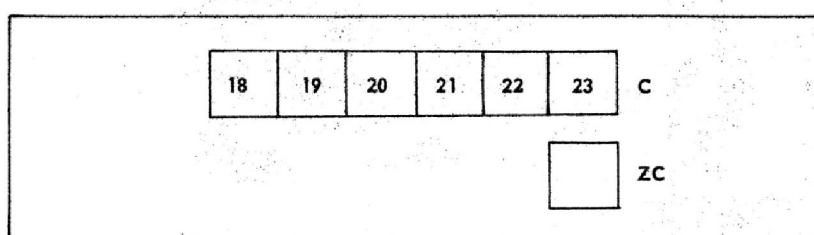
gdzie N - adres skuteczny rozkazu OSA.

Czyli następuje "odtworzenie" stanu Akumulatora, rejestru W i Licznika rozkazów /z warunkami U, Z, Nd/ przerywanego programu.

Ponadto wpisuje się jedynekę do rejestrów OZ i Q.

2.6. CZASOMIERZ.

Czasomierz jest 6-bitowym rejestrem, który mierzy czas rzeczywisty. Czas rzeczywisty jest mierzony przez ten układ w pewnych maszynowych jednostkach zwanych *chwilami*. Czas jednej chwili maszynowej wynosi 20 msek. Czasomierz maszyny działa w ten sposób, że po upływie czasu jednej chwili dodaje jedynkę do rejestru C czasomierza /w pozycji 23/.



Rejestr i wskaźnik czasomierza.

Dodawanie jedynki do rejestru C, może być programowo uregulowane przez wpisanie 0 lub 1 do jednobitowego rejestru ZC. Dokładnie: do rejestru C, dodaje się 1, po upływie czasu jednej chwili, gdy we wskaźniku Zezwolenie Czasomierza ZC jest jedynka; w przeciwnym przypadku dodawanie to nie zachodzi i stan rejestru C nie ulega zwiększaniu. Wpisanie 0 lub 1 do rejestru ZC odbywa się za pomocą rozkazu UKON /punkt 3.2/.

Przerwanie czasomierza

Ze struktury rejestru C wynika, że może on przechowywać wielkości od 0 do 63. Zatem można mierzyć odcinki czasowe nie dłuższe niż 0,64 sekundy. W celu mierzenia dowolnie dużych przedziałów czasu akumuluje się w określonej komórce pamięci operacyjnej częściowe stany rejestru C. Aby ułatwić akumulację czasu w komórce, czasomierz wysyła sygnał zwany *przerwanieniem czasomierza*.

Sygnał pojawia się, gdy czasomierz osiągnie wartość 32 czyli, gdy pojawi się jedynka na pozycji 18-tej czasomierza.

Sygnal ten powoduje normalne przerwanie programu i związany jest z komórką 66. Oczywiście po pojawieniu się sygnału przerwania, proces akumulacji czasu w rejestrze C jest kontynuowany.

DOKC DODAJ DO KOMÓRKI CZAS /167/.

- Rozkaz DOKC powoduje dodanie do komórki o adresie N zawartości rejestru C czasomierza /w skali 23/; po czym odejmuje się liczbę 32 od rejestru C /czyli zeruje się pozycję C18/.
- $n := n + c$ i $C = C - 32$.
- przygotowuje warunki zgodnie z wynikiem $n+c$.

UKON UMIEŚĆ W KANALE OPERACYJNYM ADRES N /177/.

- Rozkaz UKON powoduje wpisanie do rejestru Czasomierza bity 18-23 adresu skutecznego N tego rozkazu /patrz punkt 3.2/.
- $c := N_{18-23}$

U w a g a 1. Rozkaz UKON ułatwia pomiar odcinków czasowych dowolnie małych /będących całkowitą wielokrotnością chwili maszynowej/. Jeśli np. wpisze się liczbę 24 do rejestru C, to po upływie 8 chwil /tj. 160 msek/ czasomierz spowoduje przerwanie programu, sygnalizując w ten sposób, że już minęło 8 chwil maszynowych.

U w a g a 2. Rozkazy DOKC i UKON są traktowane jako nielegalne przez sterowanie maszyny, jeśli pojawią się, gdy $F=1$ i $Q=1$.

KANAŁ OPERACYJNY.

Rola i funkcja poszczególnych zespołów tego kanału zostały opisane w poprzednich paragrafach.

Poniżej będą opisane możliwości pulpitu operatora maszyny oraz rozkazy obowiązujące w Kanale Operacyjnym. Pulpit ten jest przeznaczony głównie dla personelu technicznego maszyny, lecz może on służyć także jako pomoc przy uruchomieniu trud-

niejszych programów organizacyjnych.

3.1. PULPIT OPERATORA.

Na pulpicie operatora wyróżnia się:

- rząd 24 lampek /SG/ do sygnalizowania zawartości wybranego rejestru /A,W,L,AF,P,C,Z,/. Sygnalizowanie stanu wybranego rejestru zachodzi tylko wówczas, gdy maszyna jest w stanie STOP-u. Jest to tak zwany STOP dynamiczny. Oznacza to, że maszyna wpisuje do rejestru sygnalizacyjnego SG zawartość określonego rejestru. Sygnalizowany rejestr jest wybrany stanem trzech klawiszy: KS4, KS2, KS1.

Przyjęto następujące przyporządkowanie:

<u>KS4</u>	<u>KS2</u>	<u>KS1</u>	<u>Rejestr sygnalizowany</u>
0	0	0	nie zmienia się stan rejestru SG
0	0	1	do rejestru SG wpisuje się stan rejestru A
0	1	0	do rej. SG wpisuje się stan rejestru W
0	1	1	do rej. SG wpisuje się stan rejestru L+WAR
1	0	0	do rej. SG wpisuje się stan rejestru AF
1	0	1	do rej. SG wpisuje się stan rejestru P
1	1	0	do rej. SG wpisuje się stan rejestru C
1	1	1	do rej. SG wpisuje się stan rejestru Z

Rejestry P, C i Z są rejestrami pomocniczymi Arytmometru. Stan rejestrów C i Z jest przypadkowy /zależnie od rodzaju wykonanej operacji/. Natomiast w rejestrze P znajduje się zawsze kolejny rozkaz do wykonania /nie zmodyfikowany; zatem rozkaz ten ma taką postać, jak w Pamięci Operacyjnej/.

- rząd 24 klawiszy stabilnych /KL/, których stan może być wprowadzony do komórki Pamięci Operacyjnej lub do rejestrów programowych /ręcznie lub programowo/,
- rząd 10 lampek do sygnalizowania zawartości rejestru AS /adres pamięci stałej/,
- rząd 10 klawiszy stabilnych /KS/ do ustawienia adresu mikro-rozkazu,

- lampki U, Z, Nd do sygnalizacji stanu warunków: U, Z, Nd,
- osiem lampek: 0,1,2,3,4,5,6,7 do sygnalizacji zajętości odpowiedniego Kanału,
- klawisz OPERACJA /niestabilny/ do wykonania ciągu mikrooperacji, którego początek określa klawiatura KS,
- klawisz START /niestabilny/ do rozpoczęcia wykonania ciągu mikrooperacji, którego początek określa zawartość rejestru AS,
- klawisz PK MIKRO /stabilny/ dla określenia Pracy Krokowej /klawisz wciśnięty/ lub pracy ciągłej /klawisz zwolniony/ przy wykonywaniu ciągu mikrooperacji,
- klawisz PK MAKRO /stabilny/ dla określenia Pracy Krokowej /klawisz wciśnięty/ lub pracy ciągłej /klawisz zwolniony/ przy wykonywaniu ciągu rozkazów /programu/.
- klawisz PP /niestabilny/ do przerywania programu /tj. przerywanie o numerze 3 w Kanale zerowym, wejście przez komórkę 67/.

Przy pomocy klawiszy, znajdujących się na pulpicie, można wykonać pewne operacje zwane o p e r a c j a m i p u l p i t o w y m i.

Treść operacji pulpitowej określa jej numer nastawiony na klawiszach KS. Natomiast wykonanie tej operacji następuje po naciśnięciu klawisza OPERACJA.

Poniżej podaje się numery operacji pulpitowych oraz przypisaną im funkcję:

Numer Oktalny
Operacji

Treść Operacji pulpitowej

100

Umieszczenie w rejestrze sygnalizacji SG zawartości komórki, której adres określa rejestr AF oraz dodanie jedynek do rejestru AF i stop.

101

Wykonanie sekwencji rozkazów, której początek określa adres ustawiony na klawiszach KL.

- 102 Zapamiętanie stanu klawiatury KL w komórce Pamięci Operacyjnej, której adres określa rejestr AF oraz dodanie jedynek do rejestru AF i stop.
- 103 Wykonanie w Kanale Operacyjnym rozkazu określonego przez stan klawiatury KL /przez klawisze o numerach od 10 do 23, patrz punkt 3.2/ i stop.
- 104 Ładowanie adresu nastawionego na klawiaturze KL do rejestru AF /jest to rejestr adresu Pamięci Operacyjnej/ i stop.
- 105 Zapamiętanie stanu klawiatury KL w komórce Pamięci Operacyjnej, której adres określa rejestr AF oraz dodanie jedynek do rejestru AF i powtórzenie czynności tej operacji /dla nowego stanu AF/.
- Uwaga: po zapisaniu KL do danej komórki następuje odczyt z tej komórki i sprawdzenie poprawności cechy parzystości. Jeśli wykryje się błąd, to maszyna zatrzymuje się, a w rejestrze AF znajduje się adres komórki następnej.
- Przy pomocy tego rozkazu można:
- wyzerować dowolny obszar pamięci,
 - wpisać same jedynek w dowolny obszar pamięci,
 - wpisać wybrany rozkaz w dany obszar testujący pracę Pamięci Operacyjnej.

Uwaga: w czasie stopu w rejestrze SG jest sygnalizowany stan jednego z rejestrów: A, W, L, AF, P, C, Z.

Czyli po operacji pulpituwej:

- 100 - sygnalizuje się stan komórki,
- 101 - sygnalizuje się stan Akumulatora,
- 102 - sygnalizuje się stan rejestru W,
- 103 - sygnalizuje się stan L+WAR
- 104 - sygnalizuje się stan AF₈₋₂₃

U 1 2 3 4 5 6 7
 T SIEĆ
 U Z Nd F Q OZ BP
 R ZW

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 SG
 KL

Przerwanie programu operacja start
 Makro Mikro
 512 256 128 64 32 16 8 4 2 1
 AS
 KS

PULPIT OPERATORA

LAMPKA

KLAWISZ NIESTABILNY

KLAWISZ STABILNY

3.2. ROZKAZY KANAŁU OPERACYJNEGO.

Jeśli w Kanale Operacyjnym pojawi się rozkaz UKON /kod ósemkowy 177/, to adres skuteczny tego rozkazu /bity 10-23/ zostaje zinterpretowany przez Kanał Operacyjny jako rozkaz. Rozkaz ten, obowiązujący tylko w Kanale Operacyjnym, został wykorzystany w sposób następujący:

- bity od 10 do 14 adresu N /czyli N_{10-14} / określają rodzaj operacji, która ma być wykonana w Kanale,
- bity od 15 do 23 określają parametr, którego wartość zostaje wpisana do jednego z rejestrów:
V, H, ZW, C, ZC, OZ, F, Q, Kanału Operacyjnego /zależnie od rodzaju operacji/.

Poniższa tabela pokazuje treść poszczególnych operacji Kanału Operacyjnego.

<u>Wartość ósemkowa</u>	<u>Operacja określona wielkością N_{10-11}</u>	<u>Operacja określona wielkością N_{12-14}</u>
0	-	-
1	$F:=N_{15}$	$V_{8-15}:=N_{16-23}$
2	$OZ:=N_{15}$	$H_{8-15}:=N_{16-23}$
3	$C_{18-23}:=N_{18-23}$	$ZW_{1-7}:=N_{17-23}$ $ZC:=N_{16}$
4	-	$Q:=N_{15}$
5	-	$V:=N_{16-23}$, $Q:=N_{15}$
6	-	$H:=N_{16-23}$, $Q:=N_{15}$
7	-	$ZW:=N_{17-23}$, $Q:=N_{15}$ $ZC:=N_{16}$

Przykład: jeśli część adresowa rozkazu UKON /Umieść w Kanale Operacyjnym adres N/ ma postać binarną:

$\frac{N_{10-11}}{10}$	$\frac{N_{12-14}}{111}$	$\frac{N_{15-23}}{011111111}$
------------------------	-------------------------	-------------------------------

to wykonanie tego rozkazu powoduje:

- wyzerowanie wskaźników OZ i Q /gdyż $N_{15}=0/$,
- wpisanie jedynek do pozycji 1-7 rejestru ZW /Zezwolenie WE-WY/,
- wpisanie jedynki do wskaźnika ZC /jest to wskaźnik Zezwolenia Czasomierza, patrz punkt 2.6/.

4. ORGANIZACJA WSPÓŁPRACY JEDNOSTKI CENTRALNEJ Z KANAŁAMI.

Informacje z U r z ą d z e ń Z e w n ę t r z n y c h są wprowadzane /lub wyprowadzane/ do /lub z/ Pamięci Operacyjnej za pośrednictwem K a n a ł ó w. Do Jednostki Centralnej można dołączyć 7 kanałów przesyłania informacji do /lub z/ Pamięci Operacyjnej, zaś do każdego kanału można dołączyć 8 Urządzeń Zewnętrznych wprowadzania lub wyprowadzania informacji.

Kanały podłączone do Jednostki Centralnej pracują równocześnie z Arytmometrem.

Jednoczesną pracą kanałów i Arytmometru steruje tzw.

K o o r d y n a t o r Kanałów, który wstrzymuje kontakty Arytmometru z Pamięcią Operacyjną na czas pisania /lub czytania/ jednego słowa z Kanału do Pamięci. Czas ten jest bliski czasowi cyklu pamięci operacyjnej. W tym czasie także Arytmometr na ogół pracuje, wykonując rozpoczęty ciąg mikrooperacji /np. mnożenia, dzielenia/.

Powyższa zasada współpracy Arytmometru z Kanałami nosi nazwę w s t r z y m a ń p r o g r a m o w y c h.

W kanałach są układy, które pozwalają określić miejsce przepisanych informacji oraz ilość słów /lub znaków/ do przepisania.

Do zapoczątkowania operacji przesyłania informacji używa się specjalnych rozkazów przygotowujących rejestry Kanałów oraz określających sposób przesyłania.

Fakty zakończenia operacji przesyłania informacji są sygnalizowane przez p r z e r y w a n i a p r o g r a m o w e Kanału /cztery przerywania w każdym Kanale/.

Kanały łączy się z Centralną Jednostką za pomocą ujednolico-

nych /technicznie i organizacyjnie/ p o ł ą c z e ń.
Dzięki temu można w sposób elastyczny dołączyć żądane typy i ilości kanałów. Ujednolicony jest również sposób łączenia Urządzeń Zewnętrznych z Kanałami danego typu. Wewnętrzna struktura Kanału /oraz lista operacji obowiązująca w Kanale/ może być odmienna w różnych typach kanałów.

Z punktu widzenia wewnętrznej struktury Kanałów wyróżnia się następujące typy Kanałów Urządzeń Zewnętrznych:

- Kanały Przesyłania Znaków /KPZ/,
- Kanały Przesyłania Słów /KPS/.

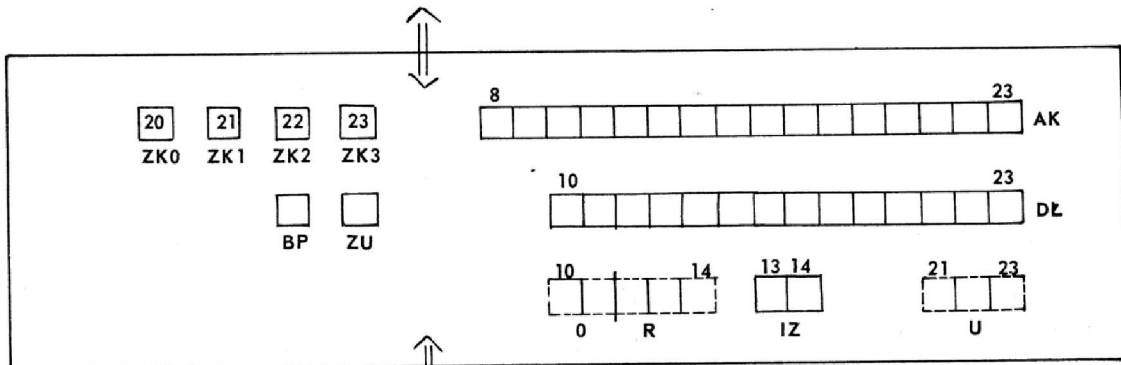
4.1. KANAŁ PRZESYŁANIA ZNAKÓW.

Kanał ten umożliwia przesyłanie bloku znaków /do 24-bitowych/ pomiędzy Pamięcią Operacyjną a wybranym Urządzeniem Zewnętrznym /np. Czytnikiem lub Perforatorem Taśmy Papierowej, Pamięcią Bębnową/.

W strukturze tego Kanału wyróżnia następujące rejestry:

- rejestr AK /16 bitów/, gdzie przechowuje się adres komórki Pamięci Operacyjnej, do której zapisuje się /lub z której czyta się/ znaki w trakcie przesyłania,
- rejestr DŁ /14 bitów/, gdzie pamięta się ilość znaków do przesyłania pomiędzy Pamięcią Operacyjną a wybranym Urządzeniem Zewnętrznym,
- rejestry ZK0, ZK1, ZK2, ZK3 /1-bitowe/ - dla zasygnalizowania czterech różnych zgłoszeń Kanału,
- rejestr BP, dla zasygnalizowania Błędu Parzystości przy operacjach przesyłań,
- rejestr ZU, dla zasygnalizowania Zgłoszenia Urządzenia,
- rejestry O, R, U dla określenia numeru: Operacji, Rejestru /wskaźnika/ i Urządzenia podłączonego do Kanału,
- rejestr IZ /2 bity/ dla określenia długości znaku przesyłanego, zero w IZ oznacza znaki 8-bitowe, zaś 3 w tym rejestrze oznacza znaki 24-bitowe.

DO lub Z PAMIĘCI OPERACYJNEJ



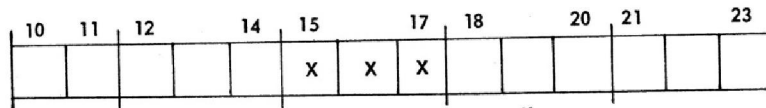
DO lub Z URZĄDZEŃ 16 - 23 lub 0 ÷ 23

4.1.1. ROZKAZY PRZYGOTOWUJĄCE REJESTRY KPZ.

Do przygotowania rejestrów AK i DŁ służą rozkazy o kodach:

- 170 N /skrót literowy: WYB N/
- 174 N /skrót literowy: UKN N/
- 175 N /skrót literowy: UKK N/

Rozkaz WYB N /170 N/ dokonuje wyboru Kanału, którego rejestry chcemy przygotować. Numer omawianego kanału /K/ jest określony przez adres N tego rozkazu w następujący sposób:



czyli numer wybranego Kanału określają^K bity 18 - 20 adresu N. W pozycjach 15 - 17 należy pisać zera, zaś w pozycjach 13-14 długość znaku wprowadzanego, a pozostałe pozycje adresu N są obojętne w tym rozkazie.

Rozkaz UKN N /174 N/ powoduje Umieszczenie w wybranym Kanale /za pomocą rozkazu WYB/ wartości adresu N.

Dokładnie: umieszcza się 16-bitowy adres N w rejestrze AK wybranego kanału, przesuając jednocześnie poprzednią zawartość rejestru AK do rejestru DŁ.

Rozkaz UKK N /175 N/ powoduje Umieszczenie w rejestrze AK wybranego Kanału /za pomocą rozkazu WYB/ zawartości komórki o adresie skutecznym N, przesuując jednocześnie poprzednią zawartość rejestru AK do rejestru DŁ.

U w a g a 1. Rozkazy UKN N /174 N/ i UKK N /175 N/ muszą być użyte bezpośrednio po rozkazie WYB N /170 N/, w żądanej kolejności i ilości, gdyż rozkaz WYB określa numer Kanału dla rozkazów UKN i UKK.

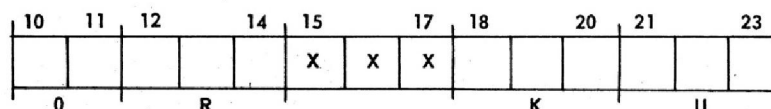
U w a g a 2. Po rozkazach: WYB, UKN i UKK nie może nastąpić przerwanie programu; czyli nastąpi także wykonanie kolejnego rozkazu programu mimo, że mogą zachodzić warunki przerwania programu.

4.1.2. ROZKAZY TESTOWANIA, PRZYGOTOWYWANIA I ZAINICJOWANIA PRACY URZĄDZEŃ KANAŁU PRZESYŁANIA ZNAKÓW.

Powyższe czynności wykonuje się odpowiednio za pomocą rozkazów:

- 171 N /TES N; TESTuj Urządzenie/
- 172 N /PGT N; PrzyGotuj Urządzenie/
- 173 N /WYK N; WYKonaj operacje/

Adres N tych rozkazów określa Urządzenie w Kanale, do którego odnosi się dany rozkaz, w następujący sposób:



- bity od 21 do 23 określają numer Urządzenia /U/, do którego odnosi się odpowiedni rozkaz,
- bity od 18 do 20 określają numer Kanału /K/, do którego należy dane urządzenie,
- bity od 12 do 14 określają rejestr /R/ danego urządzenia, do którego odnosi się odpowiedni rozkaz,
- bity od 10 do 11 określają rodzaj operacji /O/, jaka ma być wykonana przez określone urządzenie danego kanału,
- w bitach od 15 do 17 należy pisać zera /są one zarezerwowane do przyszłego użytku/.

Rozkaz PGT N. Rozkaz ten jest wprowadzony dla włączenia /lub wyłączenia/ Urządzeń do pracy, dla zapalenia /lub gaszenia/ określonych przerzutników urządzenia /częścią R adresu N/. Bliższa interpretacja sygnału "Przygotuj Urządzenie" zależy od struktury elektroniki danego modułu urządzenia i jest podana w opisie tego modułu.

Rozkaz TES N. W elektronice urządzeń znajdują się pewne rejestry 1-bitowe, zw. w s k a ź n i k a m i.
Wskaźniki te mają numery od 0 do 7 /a w razie potrzeby do 37/.

Zatem wskaźniki oznacza się przez W0, W1, ..., W7.

Część R adresu N /rozszerzona w razie potrzeby przez część 0 określa numer wskaźnika danego urządzenia.

Rozkaz TES N jest wprowadzony w celu sprawdzenia stanu wskaźników danego urządzenia. Najczęściej chodzi o testowanie czy dane urządzenie jest gotowe do pracy, tzn. czy może przyjąć z Pamięci Operacyjnej lub przekazać do niej kolejną informację. Pojawienie się tej operacji powoduje załadowanie 1 /patrz p. 4.3/ do przerzutnika TEST, gdy stan badanego wskaźnika urządzenia jest jedynką; po czym zeruje się jego stan; w przeciwnym przypadku ładuje się 0 do przerzutnika TEST. Jak wspomniano, badany wskaźnik urządzenia jest określany częścią R adresu tego rozkazu. Sposób wykorzystania części R zależy od struktury modułu danego urządzenia i jest podany każdorazowo w opisie tego modułu. Niemniej ustala się, że wartość $R=0$, $R=1$, $R=2$, czyli wskaźniki W0, W1, W2 mają tę samą interpretację we wszystkich urządzeniach podłączonych do Kanału Przesyłania Znaków:

- przyjmuje się, że W1 jest wskaźnikiem zwykłej gotowości urządzenia do przyjęcia lub przekazania kolejnej informacji,
- wskaźnik W2 związany jest z klawiszem ZGŁOSZENIE OPERATORA,
- wskaźnik W0 związany jest ze Zgłoszeniem Urządzenia i jest sumą logiczną wskaźników W1, W2, W3, ...

Stan przerzutnika TEST jest sprawdzany przy pomocy dwóch specjalnych rozkazów sterujących:

SKTO N /163 N/ i SKT1 /164 N/.

Rozkaz SKTO N powoduje skok do rozkazu z komórki N, gdy przerzutnik TEST=0; w przeciwnym przypadku nie następuje skok /czyli wykonuje się kolejny rozkaz programu/.

Rozkaz SKT1 N powoduje skok do rozkazu z komórki N, gdy przerzutnik TEST=1; w przeciwnym przypadku skok nie następuje.

Zatem w celu zbadania stanu określonego wskaźnika urządzenia należy zastosować rozkaz: TES N /Testuj Urządzenie/, a po nim jeden z rozkazów: SKTO N lub SKT1, zależnie od tego czy chce się skoczyć przy niezapalonym wskaźniku /0/, czy też przy zapalonym /1/.

U w a g a 1. Po rozkazie TES N nie zachodzi przerwanie programu, niezależnie od tego czy istnieją warunki przerwania programu, czy też nie.

U w a g a 2. Użycie rozkazu TES N lub PGT N w przypadku, gdy sam Kanał Przesyłania Znaków jest zajęty, /np. przesyłaniem blokowym/ jest dozwolone, przy czym rozkazy te nie zakłócają przebiegu transmisji w Kanale. Zatem można w czasie transmisji testować wskaźniki dowolnego modułu urządzenia podłączonego do Kanału, w którym zachodzi transmisja. Można też zapalać lub gasić określone przerzutniki dowolnego modułu.

Rozkaz WYK N /173 N/.

Rozkaz ten został wprowadzony celem zainicjowania przesyłania informacji z Pamięci Operacyjnej do określonego Urządzenia /lub z Urządzenia do Pamięci Operacyjnej/.

W Kanale Przesyłania Znaków zrealizowano następujące typy przesyłań zależnie od wartości bitów od 10 do 11 adresu N tego rozkazu /część 0/:

Wartość ósemkowa bitów 10-11	Skrót literowy operacji	Treść operacji przesyłania
0	CJZ	Czytaj Jeden Znak z Urządzenia U /podłączonego do Kanału K/ do komórki Pamięci Operacyjnej.wskazanej przez rejestr AK, /niezależnie od stanu rejestru DŁ/.
1	PJZ	Pisz Jeden Znak w Urządzeniu U, z komórki Pamięci Operacyjnej.
2	CBZ	Czytaj Blok Znaków z Urządzenia U, do pola Pamięci Operacyjnej,którego początek określa zawartość rejestru AK. Długość d bloku określa zawartość rejestru DŁ.
3	PBZ	Pisz Blok Znaków w Urządzeniu U z pola Pamięci Operacyjnej.

W czasie realizacji dwóch pierwszych typów przesyłań zawartość rejestru AK nie ulega zmianie, co pozwala na kontaktowanie się z ustaloną komórką bez powtórnego przygotowania Kanału, zaś zawartość rejestru DŁ jest obojętna.

Natomiast w czasie realizacji przesyłań 2 lub 3, zawartość rejestru AK zostaje powiększona o jeden po każdym przeczytaniu lub pisaniu znaku, a stan rejestru DŁ zostaje pomniejszony o jeden.

Fakty zakończenia operacji przesyłania informacji, sygnalizuje się przez wpisanie 1 do któregoś z rejestrów zgłoszeń: ZK0, ZK1, ZK2. Znaczenie poszczególnych zgłoszeń Kanału jest następujące:

- Zgłoszenie ZK0=1 oznacza zakończenie operacji przesyłania informacji spowodowane wyzerowaniem licznika DŁ, przy czym w czasie przesyłania nie powstał błąd ani na drogach przesyłania /pomiędzy Kanałem a Urządzeniem/, ani w samym Urządzeniu, ani też nie napotkano ZNAKU KOŃCA w czasie przesyłania ciągu znaków;

- Zgłoszenie ZK1=1 oznacza zakończenie przesyłania, spowodowane pojawieniem się ZNAKU KOŃCA. Pojawienie się tego znaku powoduje zakończenie przesyłania niezależnie od stanu Licznika DE. Sygnał ZNAKU KOŃCA nadają elektroniki modułów Urządzeń Zewnętrznych;
- Zgłoszenie ZK2=1 oznacza także zakończenie przesyłania, lecz w czasie przesyłania powstał Błąd Parzystości na drodze pomiędzy Kanalem a Urządzeniem. Testowanie cechy parzystości przesyłanego znaku odbywa się w Kanale /przez rejestr BP/ dla operacji czytania z Urządzeń, zaś dla operacji pisania do Urządzeń w elektronice Urządzenia.
Natomiast przygotowanie cechy parzystości przebiega odwrotnie;

Oprócz zgłoszeń ZK0, ZK1, ZK2, które są związane z faktami zakończenia transmisji, znajduje się także w Kanale zgłoszenie ZK3, które jest sumą logiczną wszystkich zgłoszeń modułów Urządzeń Zewnętrznych /sygnały ZU/.

Sygnały ZU mogą pojawiać się w dowolnym momencie pracy Kanalu i są natychmiast załatwiane /tzn. następuje przerwanie związane z drogą ZK3/ niezależnie od przebiegu ewentualnej transmisji informacji.

Pojawienie się sygnału ZU w module Urządzenia świadczyć może np., że została zakończona w module jakaś zlecona jemu praca lub pojawił się błąd w module określonego typu.

Każdy moduł Urządzenia Zewnętrznego może nadawać sygnał ZGŁOSZENIE URZĄDZENIA /ZU/, który jest sumą logiczną określonych wskaźników modułu: W1, W2, W3,...

Sumaryczny sygnał ZU jest formalnie traktowany w module jako wskaźnik WO /patrz punkt 4.3./. Zatem przy pomocy rozkazu TEST N można sprawdzić, czy dany moduł nadał sygnał ZU /WO/. Jeśli tak, to testując kolejno wskaźniki można określić zdarzenie, które spowodowało to zgłoszenie.

Zgłoszenia ZK0, ZK1, ZK3 mogą pojawić się w Kanale jednocześnie i dlatego obowiązuje następująca kolejność załatwienia tych zgłoszeń: ZK3, ZK2, ZK1, ZK0.

Sposób komunikowania się Kanalu Przesyłania Znaków z podłączonymi do niego Urządzeniami jest jednolity niezależnie od

rodzaju Urządzeń. Numeracja Urządzeń Zewnętrznych jest dowolna i może być ustalona przy pomocy stabilnych klawiszy umieszczonych przy elektronice danego Urządzenia. Natomiast numeracja Kanałów jest zadana sposobem podłączenia Kanału do Jednostki Centralnej.

U w a g a: Niektóre Urządzenia wprowadzają lub wyprowadzają znaki nie 8-bitowe, lecz np. 5,6 lub 7-bitowe. W takich przypadkach wartości znaków zajmują odpowiednią ilość prawych pozycji słowa, a pozostałe pozycje przyjmują wartość równą zeru /w elektronice Urządzenia/.

Testowanie wskaźników Kanału.

Przy pomocy rozkazu TES N oraz rozkazu SKTO N lub SKT1 N można zbadać nie tylko stan wskaźników elektroniki Urządzeń, lecz także wskaźniki elektroniki samego Kanału.

W Kanale bada się programowo przy pomocy rozkazu TES N następujące wskaźniki: ZKO, ZK1, ZK2 i ZK3.

Powyższe wskaźniki Kanału mają odpowiednio numery ósemkowe: 20, 21, 22 i 23. Numer ten jest określany przez część 0 /bity 10-11/ i R /bity 12-14/ adresu N rozkazu TES N.

Innymi słowy, jeśli numer zapisany w bitach 10-14 adresu N jest mniejszy od 20, to rozkaz ten testuje wskaźnik określonego modułu; jeśli zaś jest większy /lub równy/ od 20, to testowany jest wskaźnik należący do Kanału.

U w a g i:

I Wprowadzanie i wyprowadzanie znaków, przez Kanał Przesyłania Znaków, odbywa się za pomocą rozkazów: PJZ, CJZ, PBZ, CBZ; przy czym po wykonaniu treści rozkazów PBZ i CBZ zostaje zapalony wskaźnik Kanału ZKO lub ZK1 /stosownie do sposobu zakończenia transmisji/.

Natomiast po wykonaniu rozkazu PJZ lub CJZ wskaźniki Kanału ZKO i ZK1 nie zostają zapalone /czyli nie następuje przerwanie związane z drogą zerową i pierwszą/.

Uwaga: Wskaźniki ZK2 i ZK3 zapalają się /zgodnie z zachodzącymi faktami/ zarówno w czasie wykonania rozka-

zów PBZ i CBZ, jak i PJZ i CJZ.

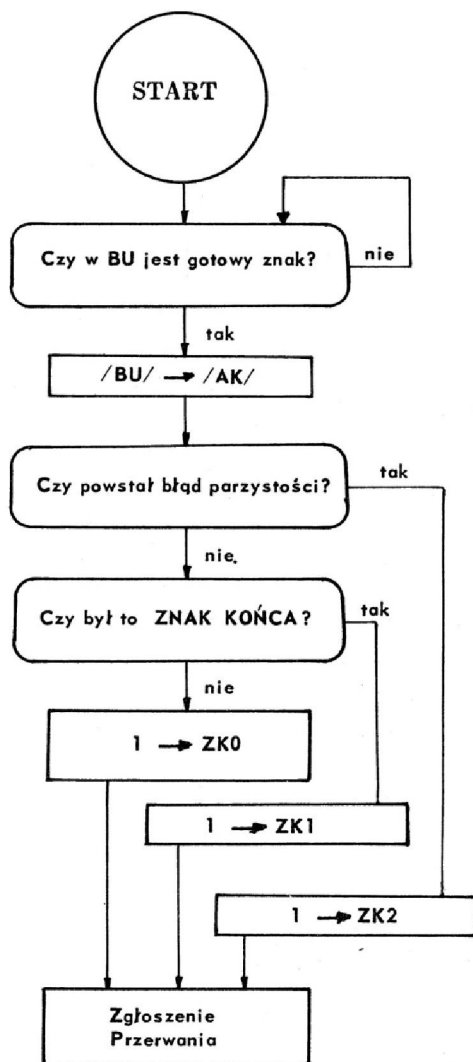
- II Rozkaz bezpośrednio następujący w programie po rozkazie PJZ /lub CJZ/, nie może dotyczyć komórki, której stan został wyprowadzony /wprowadzony/ za pomocą tego rozkazu.
- III Rozkaz bezpośrednio następujący w programie po rozkazie PJZ /lub CJZ/ nie może dotyczyć tego samego Kanału co rozkaz PJZ /CJZ/. Chyba, że jest to rozkaz typu PGT lub TES.

4.1.3. BLOKOWE PRZEDSTAWIENIE FUNKCJI KPZ.

Poniższe 4 rysunki pokazują operacje Kanału Przesyłania Znaków, które wykonuje się przy:

- czytaniu jednego znaku z Urządzenia do Pamięci Operacyjnej,
- blokowym czytaniu znaków z Urządzenia do Pamięci Operacyjnej,
- pisaniu jednego znaku z Pamięci Operacyjnej do Urządzenia,
- pisaniu blokowym znaków z Pamięci Operacyjnej do Urządzenia.

Czytanie jednego znaku.



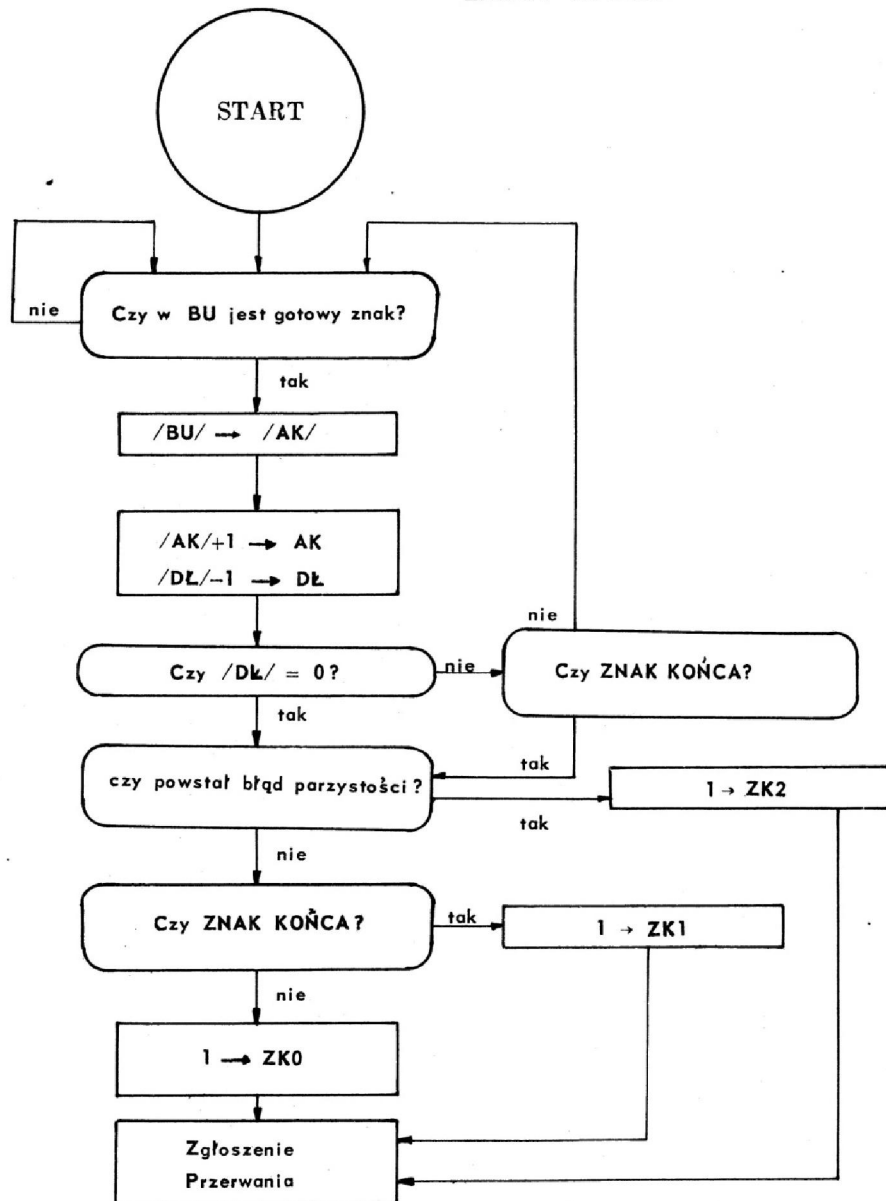
Wyjaśnienia:

- START. Sygnał ten, zainicjowany rozkazem WYK N, powoduje załadowanie kolejnego znaku z nośnika informacji do BU,
 - BU oznacza Bufor Urządzenia, w którym pamięta się gotowy znak do przekazania,
 - Zgłoszenie Przerwania oznacza przerwanie programu związane z komórką:
 - . 64+4K+0, gdy ZK0=1
 - . 64+4K+1, gdy ZK1=1
 - . 64+4K+2, gdy ZK2=1
- gdzie K oznacza numer Kanału.

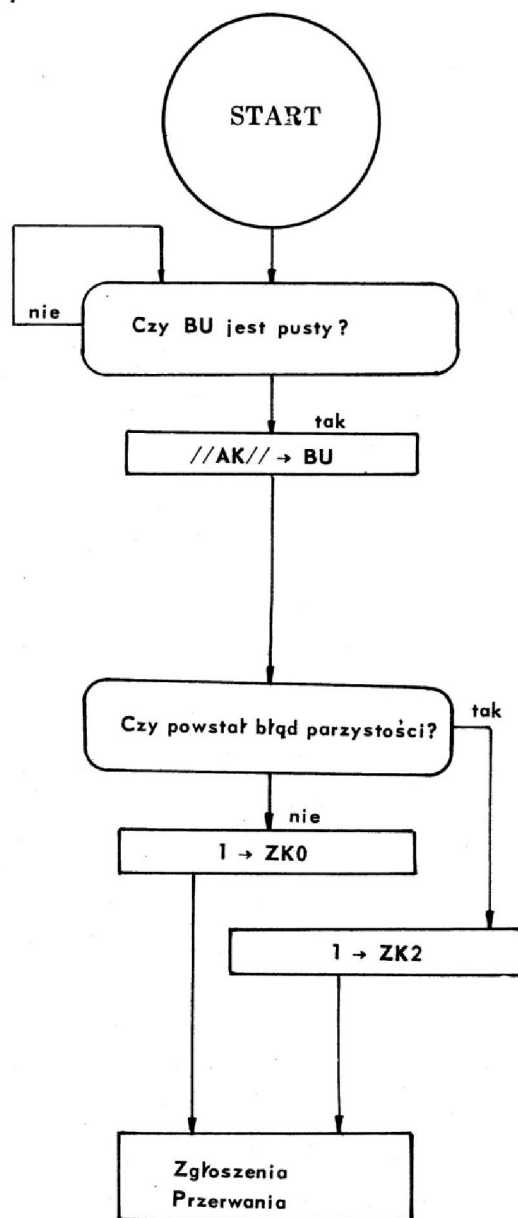
Czytanie blokowe znaków.

Uwaga 1. Sygnał START, zainicjowany WYK N, powoduje załadowanie znaku do Buforu Urządzenia, gdy jest pusty.

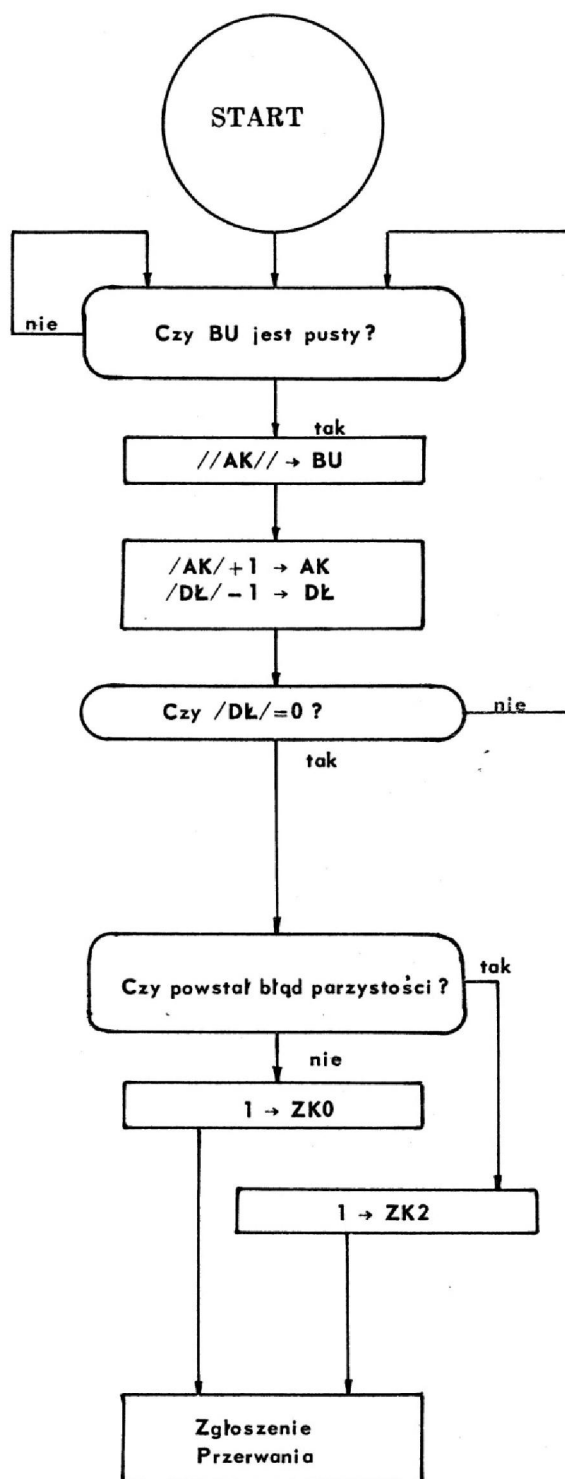
Uwaga 2. Sygnał START trwa aż do momentu wyzerowania się rejestru DŁ lub pojawienia się ZNAKU KOŃCA.



Pisanie jednego znaku



Blokowe pisanie znaków

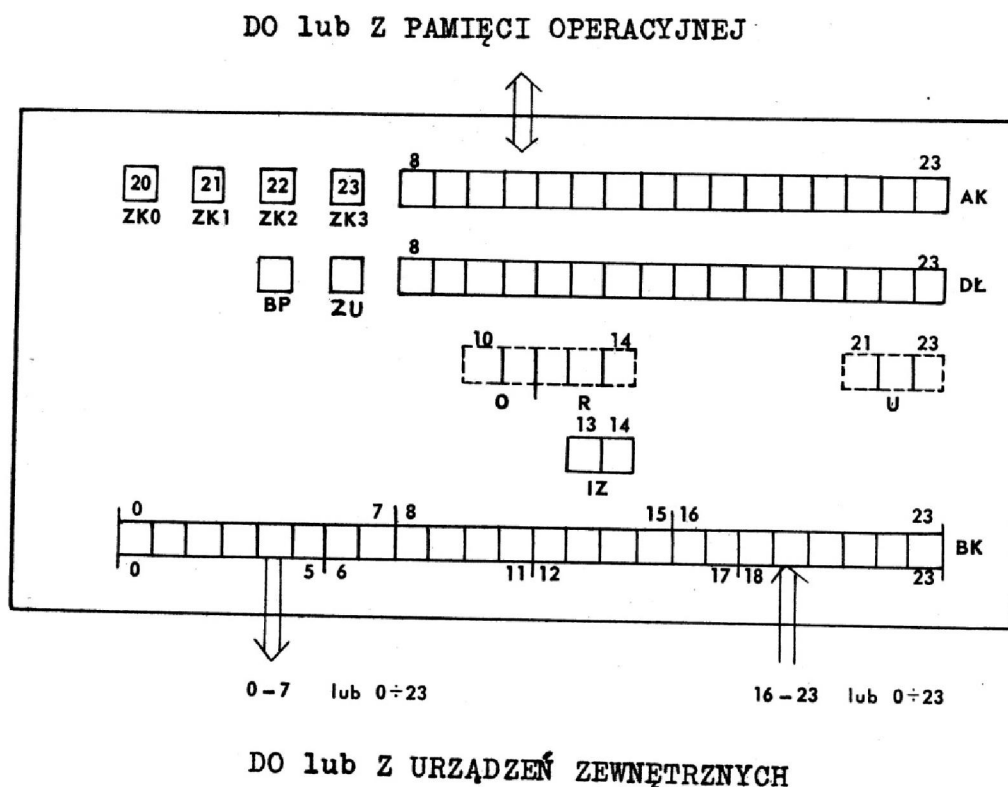


4.2. KANAŁ PRZESYŁANIA SŁÓW.

Kanał Przesyłania Słów /KPS/ umożliwia przesyłanie bloku słów /24-bitowych/ lub znaków /do 8 bitów/ pomiędzy Pamięcią Operacyjną a wybranym Urządzeniem Zewnętrznym. Do każdego Kanału Przesyłania Słów można dołączyć do 8-iu Urządzeń Zewnętrznych, takich jak:

- Czytnik lub Perforator Taśmy Papierowej,
- Monitor,
- Czytnik Kart Perforowanych,
- Drukarka Wierszowa,
- Dziurkarka Kart,
- Pamięć Bębnowa,
- Pamięć Taśmy Magnetycznej oraz inne dowolne urządzenia przekazujące lub odbierające informacje w postaci ciągu znaków /ewentualnie łączonych w słowa/.

Struktura Kanału Przesyłania Słów jest pokazana na poniższym rysunku:



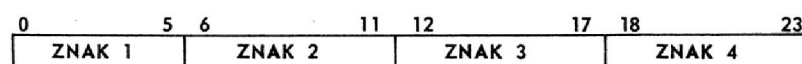
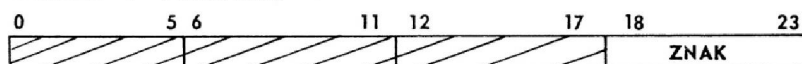
W strukturze Kanału Przesyłania Słów wyróżnia się:

- rejestr AK /16 bitów/, gdzie przechowuje się adres komórki Pamięci Operacyjnej, do której zapisuje się /lub z której czyta się/ znaki w trakcie przesyłania,
- rejestr DL /16 bitów/, gdzie pamięta się ilość słów do przesyłania pomiędzy pamięcią operacyjną a urządzeniem zewnętrznym,
- rejestry ZK0, ZK1, ZK2, ZK3 /1-bitowe/ - dla zasygnalizowania czterech różnych zgłoszeń Kanału,
- rejestr BP, dla zasygnalizowania Błędu Parzystości przy operacjach przesyłań,
- rejestry O,R,U dla określenia numeru: Operacji, Rejestru /wskaźnika/ i Urządzenia podłączonego do Kanału,
- rejestr IZ /2 bity/ dla określenia ilości znaków słowa przesyłanego,
- rejestr BK /24 bity/ spełniający rolę Bufora Kanału, w którym kompletuje się słowa ze znaków /przy wprowadzaniu z urządzenia/ lub dekompletuje się słowa na znaki /przy wyprowadzaniu na urządzenie/.

Z punktu widzenia struktury kanału wyróżnia się znaki 6-bitowe i 8-bitowe. Znaki o mniejszej ilości bitów są uzupełniane zerami /z lewej strony/ do znaku odpowiednio 6 lub 8-bitowego.

Przykłady słów zawierających:

- jeden znak 6-bitowy,
- jeden znak 8-bitowy,
- cztery znaki 6-bitowe,
- trzy znaki 8-bitowe.



U w a g a: zakreskowany obszar słów pokazanych na poprzednich rysunkach zawiera zera przy wprowadzaniu z urządzeń do maszyny, a przy wyprowadzaniu jest obojętny.

Struktura Kanału Przesyłania Słów umożliwia przesyłania słów zawierających:

- jeden znak 8-bitowy,
- trzy znaki 8-bitowe,
- cztery znaki 6-bitowe,
- jeden znak 24-bitowy.

O tym, ile znaków i jakie zawiera słowo przesyłane, decyduje wielkość rejestru IZ w następujący sposób:

<u>IZ</u>	<u>Ilość znaków słowa</u>
0	Słowo wprowadzane lub wyprowadzane zawiera jeden znak 8-bitowy /raczej do 8-bitów, zależnie od typu urządzenia zewnętrznego/. Znak ten zapisuje się w pozycjach 16-23 słowa.
1	Słowo zawiera trzy znaki 8-bitowe: <ul style="list-style-type: none">- pierwszy znak w pozycjach 0-7,- drugi znak w pozycjach 8-15,- trzeci znak w pozycjach 16-23.
2	Słowo zawiera cztery znaki 6-bitowe: <ul style="list-style-type: none">- pierwszy znak w pozycjach 0-5,- drugi znak w pozycjach 6-11,- trzeci znak w pozycjach 12-17,- czwarty znak w pozycjach 18-23.
3	Wprowadza /wyprowadza/ się pełne słowo, czyli 24-bitowy znak.

4.2.1. ROZKAZY PRZYGOTOWUJĄCE REJESTRY KPS.

Rejestry AK i DŁ przygotowuje się za pomocą rozkazów: UKN N i UKK N. Funkcja tych rozkazów oraz struktura adresu w odniesieniu do Kanału Przesyłania Słów jest identyczna

- z funkcją tych rozkazów w odniesieniu do Kanału Przesyłania Znaków /patrz pkt 4.1.1./.

Natomiast rozkaz WYB N wybiera nie tylko Kanał, którego rejestry przygotowujemy /za pomocą rozkazów UKN N i UKK N/, lecz także przygotowuje stan rejestru IZ. Dokładnie: Część R adresu N /bity 13-14/ rozkazu WYB N zostaje wpisana do rejestru IZ danego Kanału Przesyłania Słów.

Przygotowanie i testowanie pracy urządzeń zewnętrznych podłączonych do Kanału Przesyłania Słów odbywa się tak samo jak w przypadku podłączenia ich do Kanału Przesyłania Znaków.

4.2.2. ZAINICJOWANIE PRACY KANAŁU I URZĄDZEŃ.

Praca Kanału Przesyłania Słów i Urządzenia zostaje zapoczątkowana rozkazem WYK N, analogicznie jak w przypadku Kanału Przesyłania Znaków, z tą różnicą, że słowo przesyłane może zawierać jeden, trzy lub cztery znaki.

Zatem część O adresu N tego rozkazu /bity 10-11/ określa numer operacji w Kanale, zaś część R tego adresu /bity 12-14/ jest dodatkową informacją dla elektroniki urządzenia i bliżej określa sposób wprowadzania i wyprowadzania informacji.

W Kanale Przesyłania Słów /analogicznie do KPZ/ zrealizowano następujące operacje:

Wartość ósemkowa w części N ₁₀₋₁₁	Skrót literowy operacji	Treść operacji przesyłania
0	CJS	Czytaj Jedno Słowo z Urządzenia U podłączonego do Kanału K, do komórki pamięci operacyjnej wskazanej przez rejestr AK /niezależnie od stanu rejestru DE/.
1	PJS	Pisz Jedno Słowo do Urządzenia.

- 2 CBS Czytaj Blok Słów z Urządzenia do Pamięci Operacyjnej. Długość bloku określa stan rejestru DŁ.
- 3 PBS Pisz Blok Słów do Urządzenia z Pamięci Operacyjnej.

U w a g a 1: W czasie realizacji dwóch pierwszych rozkazów /CJS, PJS/ stan rejestru AK nie ulega zmianie, zaś rejestru DŁ jest obojętny. Natomiast w czasie realizacji dwóch ostatnich rozkazów /CBS, PBS/ stan rejestru AK zostaje powiększony o 1 po każdym przeczytaniu lub pisaniu jednego słowa, a stan rejestru DŁ zostaje pomniejszony o jeden.

U w a g a 2: Słowa wprowadzane i wyprowadzane są interpretowane zgodnie ze stanem rejestru IZ. Jeśli np. stan rejestru IZ równa się 1, to Kanał czyta /przy wprowadzaniu/ trzy kolejne znaki 8-bitowe z Urządzenia, kompletuje je w słowo 24-bitowe; po czym zapisuje to słowo w Pamięci Operacyjnej.

U w a g a 3: Fakty zakończenia operacji przesyłania informacji /słów/ są sygnalizowane przez wpisanie 1 do któregoś z rejestru zgłoszeń Kanałów: ZK0, ZK1, ZK2. Natomiast wpisanie 1 do rejestru ZK3 świadczy o tym, że nastąpiło zgłoszenie z inicjatywy jakiegoś modułu, podłączonego do tego Kanału.

Znaczenie poszczególnych zgłoszeń Kanału Przesyłania Słów jest identyczne z funkcją odpowiednich zgłoszeń Kanału Przesyłania Znaków.

Zatem zgłoszenia ZK0, ZK1, ZK2, ZK3 powodują przerwanie programowe /związane odpowiednio z komórkami: $64+4K+0$, $64+4K+1$, $64+4K+2$, $64+4K+3$ / i mają następującą interpretację:

- ZK0=1 oznacza koniec transmisji spowodowany wyzerowaniem się licznika DŁ,
- ZK1=1 oznacza koniec transmisji spowodowany pojawieniem się ZNAKU KOŃCA,
- ZK2=1 oznacza koniec transmisji, przy czym w czasie przesyłania powstał błąd parzystości na drodze pomiędzy Kanałem a modułem Urządzenia,
- ZK3=1 oznacza, że powstał sygnał zgłoszenia w którymś z modułów podłączonych do Kanału. Zgłoszenie to może pojawić się w dowolnym momencie pracy Kanału i zostaje natychmiast załatwiane.

U w a g a 4: Struktura Kanału Przesyłania Znaków /4.1./ jest uproszczoną wersją struktury Kanału Przesyłania Słów. Kanał Przesyłania Znaków posiada następujące uproszczenia:

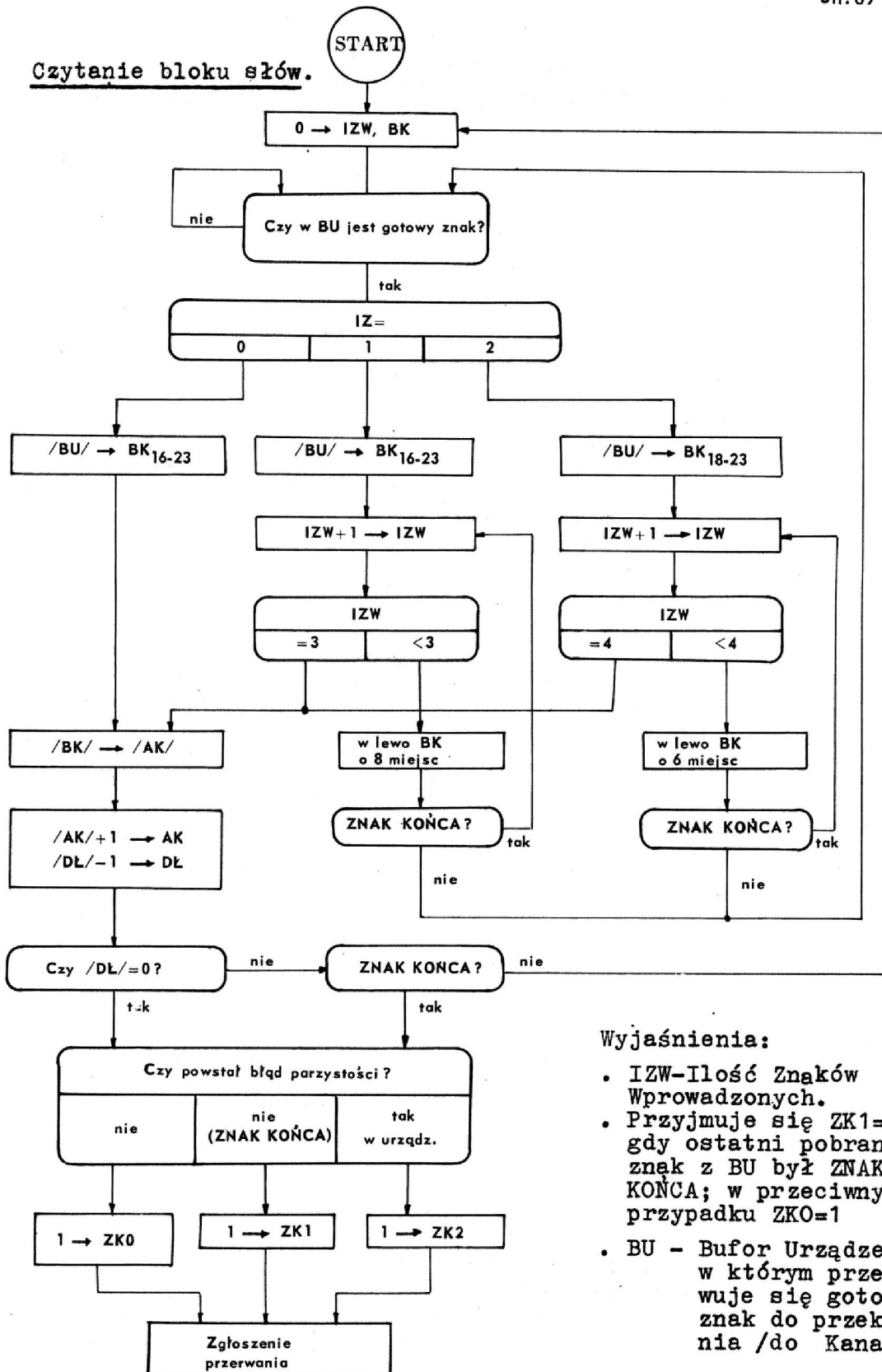
- nie posiada Buforu kompletującego /i dekompletującego/ słowa ze znaków,
- posiada 14-bitowy licznik długości bloków /zamiast 16-bitowego/,

Elektroniki modułów urządzeń zewnętrznych podłącza się do Kanału Przesyłania Słów w sposób identyczny jak do Kanału Przesyłania Znaków. Oznacza to, że wprowadzanie /i wyprowadzanie/ informacji, z modułu urządzenia do KPS lub KPZ, odbywa się za pomocą tych samych standardowych złącz.

4.2.3. BLOKOWE PRZEDSTAWIENIE FUNKCJI KPS.

Rysunki pokazane na następnych dwóch stronach przedstawiają operacje Kanału Przesyłania Słów, które wykonuje się przy:

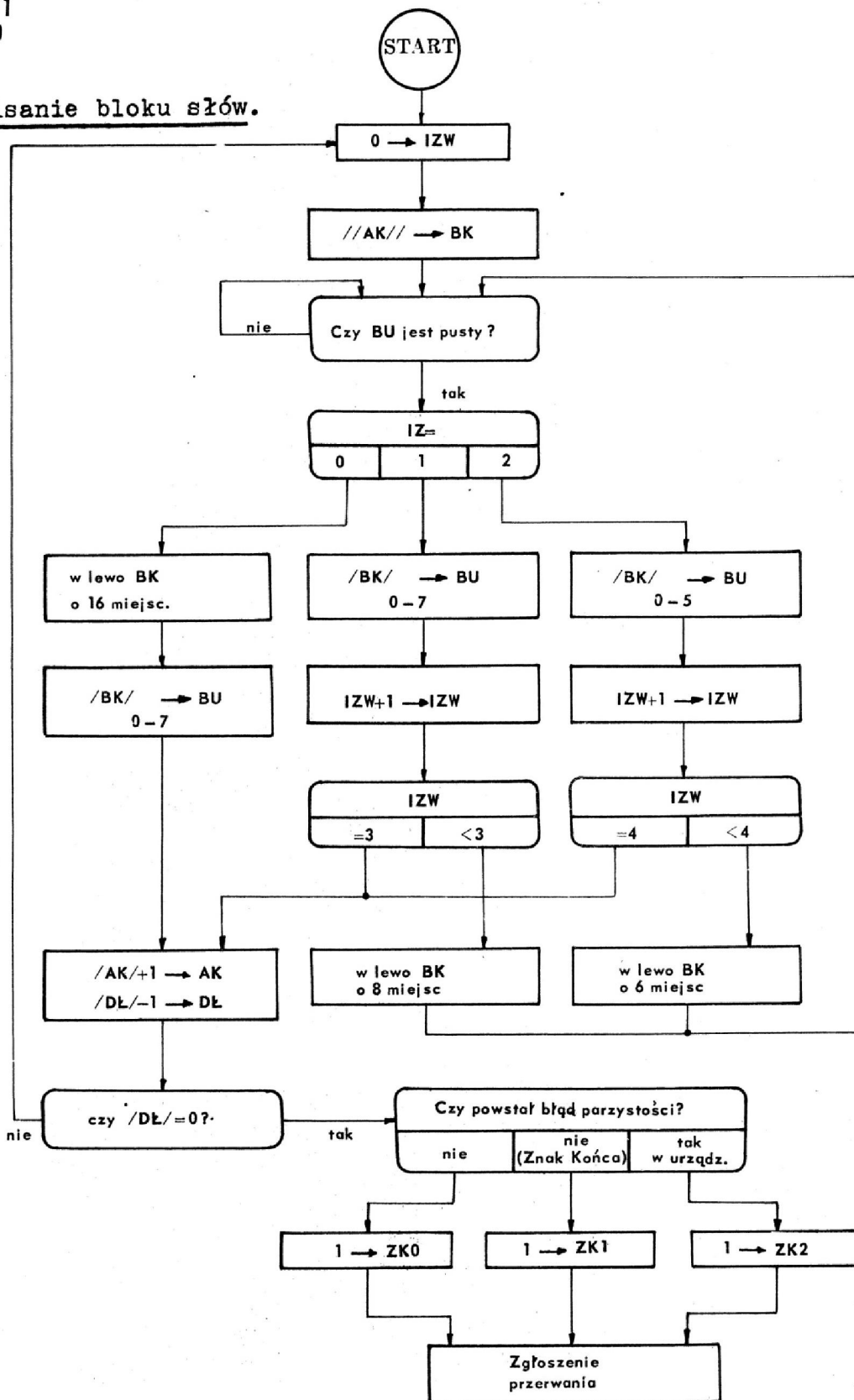
- czytaniu bloku słów z Urządzenia do Pamięci Operacyjnej,
- pisaniu bloku słów do Urządzenia z Pamięci Operacyjnej.



Wyjaśnienia:

- IZW-Ilość Znaków Wprowadzonych.
- Przyjmuje się ZK1=1, gdy ostatni pobrany znak z BU był ZNAKIEM KOŃCA; w przeciwnym przypadku ZK0=1
- BU - Bufor Urządzenia, w którym przechowywany jest gotowy znak do przekazania /do Kanału/

Pisanie bloku słów.



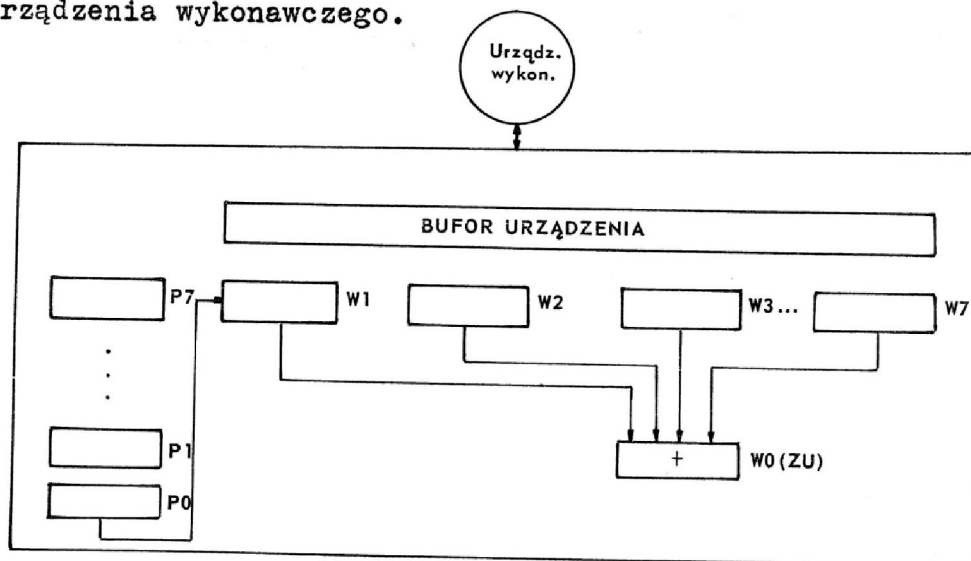
BU - Bufor Urządzenia,
w którym przechowuje się
wyprowadzany znak /z Kanału/

IZW - Ilość Znaków
Wyprowadzonych.

W rysunkach tych pokazano sposób przygotowania rejestrów zgłoszeń Kanału Przesyłania Słów ZK0, ZK1 i ZK2, gdyż zgłoszenia te są ściśle związane z transmisją w Kanale. Nie pokazano natomiast, ładowania rejestru ZK3, gdyż jest to zgłoszenie dodatkowe i nie zawsze związane z modułem, będącym aktualnie w transmisji.

4.3. OGÓLNA ORGANIZACJA MODUŁÓW URZĄDZEŃ ZEWNĘTRZNYCH.

W tej części opisu omówimy pewne wspólne elementy programowe struktury modułów. W praktyce struktura poszczególnych modułów /np. czytnika lub perforatora taśmy papierowej/ jest szczególnym przypadkiem niżej podanej ogólnej struktury modułu, a także jest technicznie związana ze specyfiką danego Urządzenia wykonawczego.



W strukturze modułu wyróżnia się:

- W s k a ź n i k i oznaczone przez W0, W1, ..., W7;
- P r z e r z u t n i k i oznaczone przez P0, ..., P7;
- B u f o r U r z ą d z e n i a /BU/, w którym przechowuje się gotową informację modułu do przekazania /do Kanału/ lub do wyprowadzenia na u r z ą d z e n i e w y k o n a w c z e.

Wskaźniki modułu opisują stan, w którym moduł ten znajduje się w czasie pracy. Wskaźniki mogą w szczególności podawać,

informację o tym, że Bufor modułu jest pełny lub wolny, że nastawiono jakąś informację na pulpicie modułu czy też opisują typy błędów lub innych zdarzeń, jakie zaszły w module.

Rola wskaźników.

Stan poszczególnych wskaźników może być zbadany przy pomocy rozkazu TES N /numer wskaźnika jest określony przez część R adresu N/.

Przyjmuje się, że normalnie wskaźnik jest w stanie zerowym /0/, zaś przechodzi w stan jeden /1/ wówczas, gdy zajdzie zdarzenie, z którym wskaźnik ten jest związany. Przyjmuje się także, że testowany wskaźnik zostaje automatycznie zerowany po przeczytaniu jego stanu za pomocą rozkazu TES N.

Rola wskaźników w module jest dokładnie sprecyzowana przez jego strukturę i podana jest w opisie tego modułu. Niemniej pewne wskaźniki mają ustaloną funkcję, która obowiązuje we wszystkich modułach. Dokładnie:

- wskaźnik W1 określa g o t o w o ś ć l o g i c z n ą modułu do przekazania /lub odbioru/ kolejnej informacji /względnie grupy informacji/. Gotowość logiczna jest określona przez stan 1 we wskaźniku W1. Oczywiście wskaźnik ten jest testowany i zerowany przy pomocy rozkazu TES N /zerowanie następuje zawsze, czyli zanika gotowość logiczna niezależnie od gotowości fizycznej modułu, która na ogół przez jakiś czas jeszcze istnieje/,
- wskaźnik W2 określa fakt naciśnięcia klawisza /niestabilnego/ ZGŁOSZENIE OPERATORA, który znajduje się na pulpicie modułu;
- wskaźnik W0 jest s u m ą l o g i c z n ą wszystkich wskaźników modułu od W1 do W7. Oznacza to, że wskaźnik W0 jest w stanie 1, jeśli którykolwiek ze wskaźników od W1 do W7 jest w stanie 1; w stanie 0 znajduje się wtedy, jeśli wszystkie wskaźniki od W1 do W7 są w stanie 0. Oczywiście wskaźnik W0 jest także badany za pomocą rozkazu TES N, lecz nie zerowany po zbadaniu jego stanu. Wskaźnik W0 jest oznaczony również przez ZU /Zgłoszenie

Urządzenia/. Sygnał ZU /jedynka we wskaźniku WO/ jest podawany do Kanału, który powoduje natychmiast przerwanie programowe związane z drogą ZK3 danego Kanału.

Przyjmuje się oczywiście, że przerwanie ZK3 danego Kanału następuje wówczas, jeśli ogólne zezwolenie układu przerywań OZ jest w stanie 1, oraz 1 jest w odpowiedniej pozycji rejestru maski ZW /patrz punkt 2.5./.

Jeśli natomiast OZ jest w stanie 0 /lub 0 w pozycji maski/, to przerwanie to nie następuje a sygnał ZK3 jest podtrzymany.

Rola przerzutników.

Przerzutniki są także rejestrami jednobitowymi, lecz stan tych przerzutników ustawiany jest programowo za pomocą rozkazu PGT N. W ten sposób można ustawiać moduł na określony typ pracy lub na wykonanie pewnych czynności p r z y g o t o w a w c z y c h. Niektóre z tych przerzutników mogą być ustawione programowo w stan 0 lub 1, niektóre tylko w stan 1, a w stan 0 przechodzą automatycznie po wykonaniu zleconej czynności.

Numer przygotowanego przerzutnika /za pomocą rozkazu PGT N/ jest określony przez część R adresu N tego rozkazu.

Przerzutniki, które są programowo ustawiane w obu stanach mają formalnie dwa kolejne numery /np. 0 i 1 lub 6 i 7/.

Takie przerzutniki są ustawiane w stan 0, gdy podaje się ich numer parzysty /przez część R adresu N/; zaś w stan 1, gdy podaje się ich numer nieparzysty.

Funkcja poszczególnych przerzutników zależy od struktury modułu i jest określona w opisach danego modułu.

Tutaj natomiast ustalimy jedną funkcję, która ma zastosowanie w większości modułów klasycznych urządzeń. Mianowicie przyjmujemy, że pierwszy przerzutnik jest programowo ustawiony w stan 0 lub 1, czyli ma "podwójny" numer /0 i 1/.

Oznacza to, że pojawienie się rozkazu PGT N w tym module z zerem w części R /bity 12-14 adresu N/ powoduje wyzerowanie tego przerzutnika; zaś z 1 w części R powoduje załadowanie 1 w tym przerzutniku.

Rola przerzutnika $P_{0,1}$ jest następująca:

- jeśli w przerzutniku $P_{0,1}$ jest jeden, to wskaźnik W1 /gotowość logiczna/ może przyjmować stan 1, gdy moduł przechodzi w stan gotowości /ze stanu zajętości/;
- jeśli zaś w przerzutniku $P_{0,1}$ jest 0, to wskaźnik W1 jest stale w stanie 0 niezależnie od fizycznej gotowości urządzenia do przyjęcia lub oddania informacji.

Innymi słowy przerzutnik $P_{0,1}$ warunkuje pojawienie się przerwania programowego spowodowanego przez wskaźnik W1 /za pośrednictwem sygnału ZU i drogi ZK3/.

Wykorzystanie wskaźników i przerzutników.

Jak wynika ze struktury modułu może on posiadać 7 różnych zgłoszeń, które powodują dodatkowe przerwanie programowe drogą ZK3 Kanału /do którego są podłączone/.

Ta konstrukcja daje programom organizacyjnym możliwość przyjęcia sygnałów przerwania, nadanych przez elektroniki modułów. Istnieje również prosta i stosunkowo szybka metoda bliższej lokalizacji zgłoszenia danego modułu /jeśli zaistnieje taka potrzeba/.

Przedstawiona struktura Modułu oraz Kanału pozwala na zorganizowanie transmisji s z y b k i e j i w o l n e j pomiędzy Pamięcią Operacyjną a Modułem.

Transmisja szybka, która jest związana z szybkimi Urządzeniami Zewnętrznymi /np. pisanie do pamięci bębnowej lub taśmowej, ładowanie bufora drukarki wierszowej/ jest przesłanie blokowe sterowane przez Kanał zgodnie ze stanem jego rejestrów: AK, DŁ, IZ. Natomiast transmisja wolna, którą można stosować do wolnych Urządzeń Zewnętrznych /np. monitor, perforator taśmy, czytnik taśmy/ jest to przesłanie po jednym znaku pomiędzy Pamięcią Operacyjną a gotowym buforem /do przyjęcia lub oddania znaku/ modułu Urządzenia Zewnętrznego.

Transmisja wolna odbywa się także przez Kanał, lecz zaangażowany /i przygotowany każdorazowo programem/ jest tylko jego rejestr adresowy AK.

Np. czytanie gotowego znaku z modułu odbywa się za pomocą

dwóch rozkazów: UKN i CJZ /Umieść w Kanale adres N i Czytaj
Jeden Znak/.

KOD ROZKAZU Ósemkowy Literowy	FUNKCJA ROZKAZU	CZAS WYKONANIA ROZKAZU (w μ s)
002	PLR $n := l + 2$	30
003	NIC -	15
004	SKW Wykonaj rozkaz n	15
005	PGW $0 := n$	15
006	SKS $F := 0, (16) := l + 1, l := (16 + N)$	50
007	SKP $n := l + 1, l := N + 1$	32
010	SKN $l := N$ gdy $Nd = 1$	13
011	SKD $l := N$ gdy $U = Z = 0$	13
012	SKZ $l := N$ gdy $Z = 1$	13
013	SNU $l := N$ gdy $U = 0$	13
014	SKU $l := N$ gdy $U = 1$	13
015	SNZ $l := N$ gdy $Z = 0$	13
016	SND $l := N$ gdy $U = 1$ lub $Z = 1$	13
017	SKB $l := N$	10
020	USAK $a := n$	15
021	SSAK $a := a + n$	15
022	OSAK $a := a - n$	15
023	RSAK $a := n - a$	15
024	MLAK $a := a @ n$	15
025	ZMAK $a := n$ i $n := a$	26
026	UAZK $a := n$ i $n := 0$	26
027	PSAK 1. jeśli $a > n$, to $U = Z = 0$ 2. jeśli $a = n$, to $U = 0, Z = 1$ 3. jeśli $a < n$, to $U = 1, Z = 0$	21
030	PSKA $n := a$	21
031	SSKA $n := a + n$	26
032	OSKA $n := a - n$	26
033	RSKA $n := n - a$	26
034	MLKA $n := a @ n$	25
035	DOKJ $n := n + 1$	26
036	PKZA $n := a$ i $a := 0$	25
037	ODKJ $n := n - 1$	26
040	USAN $a := N$	10
041	SSAN $a := a + N$	12
042	OSAN $a := a - N$	14
043	RSAN $a := N - a$	14
044	MLAN $a := a @ N$	10
045	ZERK $n := 0$	21
046	PAKW $n := w$	21
047	USWK $w := n$	15
050	UZAK $aw := (N - 1, N)$	30
051	SZAK $aw := aw + \frac{1}{z} (N - 1, N)$	94
052	OZAK $aw := aw - \frac{1}{z} (N - 1, N)$	93

* gdy omawiany skok jest efektywny, to jego wykonanie trwa 13 μ sek; gdy natomiast jest nieefektywny, to wykonanie trwa 17 μ sek.

053	RZAK	$aw := (N-1, N) \frac{aw}{z}$	93	
054	PZKA	$(N-1, N) := aw$	43	
056	MZAK	$aw := aw \cdot \frac{(N-1, N)}{z}$	358	
057	DZAK	$aw := aw / \frac{(N-1, N)}{z}$	1064	
060	SDAN	$aw := aw + N$	15	
061	SDAK	$aw := aw + (N-1, N)$	30	
062	ODAK	$aw := aw - (N-1, N)$	30	
063	RDAK	$aw := (N-1, N) - aw$	30	
064	DLAK	$a := a \vee n$	15	
065	OLAK	$a := a \div n$	15	
066	MSAK	$aw := a \cdot n$	80	
067	DSAK	$a := aw / n$	188	
071	PAK	$a := a \cdot 2^{-N}$	$17 + S \cdot 1,5$	*
072	LNK	$a := a \cdot 2^N$	$20 + S \cdot 1,5$	
073	PNK	$a := a \cdot 2^{-N}$	$17 + S \cdot 1,5$	
074	PAD	$aw := aw \cdot 2^{-N}$	$17 + S \cdot 1,5$	
075	LND	$a := aw \cdot 2^N$	$20 + S \cdot 1,5$	
076	PCK	$a := a \cdot 2^{-N}$	$17 + S \cdot 1,5$	
077	PLAK	1. jeśli $a > n$, to $U=Z=0$; 2. jeśli $a = n$, to $U=0, Z=1$; 3. jeśli $a < n$, to $U=1, Z=0$. Uwaga: jest to porównanie logiczne argumentów.	21	
101	PKB 1	$n := (B1)$	38	
102	PKB 2	$n := (B2)$	38	
103	PKB 3	$n := (B3)$	38	
105	MOK 1	modyfikuj rozkaz następny	18	
106	MOK 2	odpowiednio przez n lub $2n$	21	
107	DDAK	$aw := aw / n$	326	
111	UB1K	$(B1) := n$	28	
112	UB2K	$(B2) := n$	28	
113	UB3K	$(B3) := n$	28	
115	UB1N	$(B1) := N$	27	
116	UB2N	$(B2) := N$	27	
117	UB3N	$(B3) := N$	27	
12 T	ROPT	$(4) := N, (5) := l + 1, l := (6 + T)$.	69	
130	SZKR	1. jeśli $\forall x = w$, to $l := l + 2$; 2. jeśli $\wedge x \neq w$, to $l := l + 1$.	$74 + N \cdot 27$	
131	SZKM	1. jeśli $\forall x < w$, to $l := l + 2$; 2. jeśli $\wedge x > w$, to $l := l + 1$.	$74 + N \cdot 34$	
132	SZMR	1. jeśli $\forall x @ (B2) = w @ (B2)$, to $l := l + 2$; 2. jeśli $\wedge x @ (B2) \neq w @ (B2)$, to $l := l + 1$.	$83 + N \cdot 27$	
133	SZMM	1. jeśli $\forall x @ (B2) < w @ (B2)$, to $l := l + 2$; 2. jeśli $\wedge x @ (B2) > w @ (B2)$, to $l := l + 1$.	$91 + N \cdot 34$	
134	PSPS	$(B3) := (B1)$ $(B3) + l := (B1) + 1$ $(B3) + N - 1 := (B1) + N - 1$.	$43 + N \cdot 56$	

* gdzie $S = \frac{N}{2}$ dla parzystych N ,
 $\frac{N+1}{2}$ dla nieparzystych N

135	ZMPS	(B3):=(B1) (B3)+1:=(B1)+1 (B3)+N-1:=(B1)+N-1.	43+N-66
136	SZKL	1. jeśli $\forall x < w$, to $l:=l+2$; 2. jeśli $\wedge x \geq w$, to $l:=l+1$.	85+N-34
137	SZML	1. jeśli $\forall x @ (B2) < w @ (B2)$, to $l:=l+2$; 94+N-34 2. jeśli $\wedge x @ (B2) \geq w @ (B2)$, to $l:=l+1$.	
160	STOP	$l:=N$ i stop	21
161	OSA	1. $a:=(N)$, 2. $w:=(N+1)$, 3. $l:=(N+2)$, 4. $OZ:=Q:=1$.	38
162	PWR	$l:=(N)$, $(N):=0$, $F:=1$	30
163	SKTO	$l:=N$ gdy $T=0$	13
164	SKT1	$l:=N$ gdy $T=1$	13
167	DOKC	$n:=n+c$	30
170	WYB		21
171	TES		21
172	PGT		21
173	WYK		21
174	UKN		17
175	UKK		18
177	UKON		20
00100	MINA	$aw:= -aw$	48
00101	ABSA	$aw:= aw $	24
00103	ENTA	$a:=E[aw]$	82
00104	STND	$a:=S[aw]$	36
00105	DECA	$aw:=aw-10$	35
00107	CZKL	$a:=KL$	17
00110	USAW	$a:=w$	17
00111	PSWA	$w:=a$	17
00112	ZMAW	$a:=w$ i $w:=a$	24
00113	ZEAW	$a:=0$ i $w:=0$	21
00114	MOA 1		21
00115	MOA 2		24
00116	MOW 1		21
00117	MOW 2		24