

MASZYNA CYFROWA
K - 202
Organizacja logiczna

Opracował Zespół
pod kierunkiem J. Karpińskiego

Treść

1. Postać informacji w maszynie
2. Schemat blokowy maszyny
3. Rejestry i wskaźniki maszyny
4. Ogólne zasady wykonywania rozkazów
 - 4.1. Pobranie rozkazu
 - 4.2. Określenie efektywności rozkazu
 - 4.3. Wyznaczanie efektywnego argumentu
 - 4.4. Wykonanie operacji określonej rozkazem
5. Lista rozkazów
6. Przerwania
7. Stany maszyny
8. Pulpit techniczny maszyny
9. Wstępne wprowadzanie programu
10. Zestawienie kodów operacji

1. POSTAĆ INFORMACJI W MASZYNIE

Maszyna K-202 pracuje w systemie binarnym. Dla zapisu liczb i w arytmetyce stosowana jest notacja uzupełnieniowa do dwóch. Słowo maszyny zawiera 16 bitów numerowanych od 0 do 15 i służy do przedstawiania następujących informacji:

- L i c z b a s t a ł o p r z e c i n k o w a k r ó t k a zajmuje jedno słowo i traktowana jest jako liczba całkowita. (Rys. 1a),
- I n f o r m a c j a l o g i c z n a zajmuje jedno słowo i traktowana jest jako ciąg 16 niezależnych bitów (Rys. 1b),
- Z n a k i a l f a n u m e r y c z n e. Ich długość, rozmieszczenie w słowie oraz kody określone są programem. W szczególności stosowane są znaki 7 lub 8-bitowe umieszczone po dwa w słowie. (Rys. 1c), według kodu zewnętrznego i wewnętrznego ISO - 7,
- L i c z b a s t a ł o p r z e c i n k o w a d ł u g a określona jest programem. W szczególności stosowana jest postać, w której liczba długa zajmuje dwa słowa i traktowana jest jako liczba całkowita. (Rys. 1d),
- L i c z b a z m i e n n o p r z e c i n k o w a zajmuje trzy słowa (Rys. 1e). Cecha zajmuje pierwsze słowo i jest liczbą stałoprzecinkową. Mantysa zajmuje dwa następne słowa i traktowana jest jako liczba zaprzecinkowa.
- R o z k a z zajmuje od jednego do czterech słów i może posiadać kilka postaci (Rys. 2).

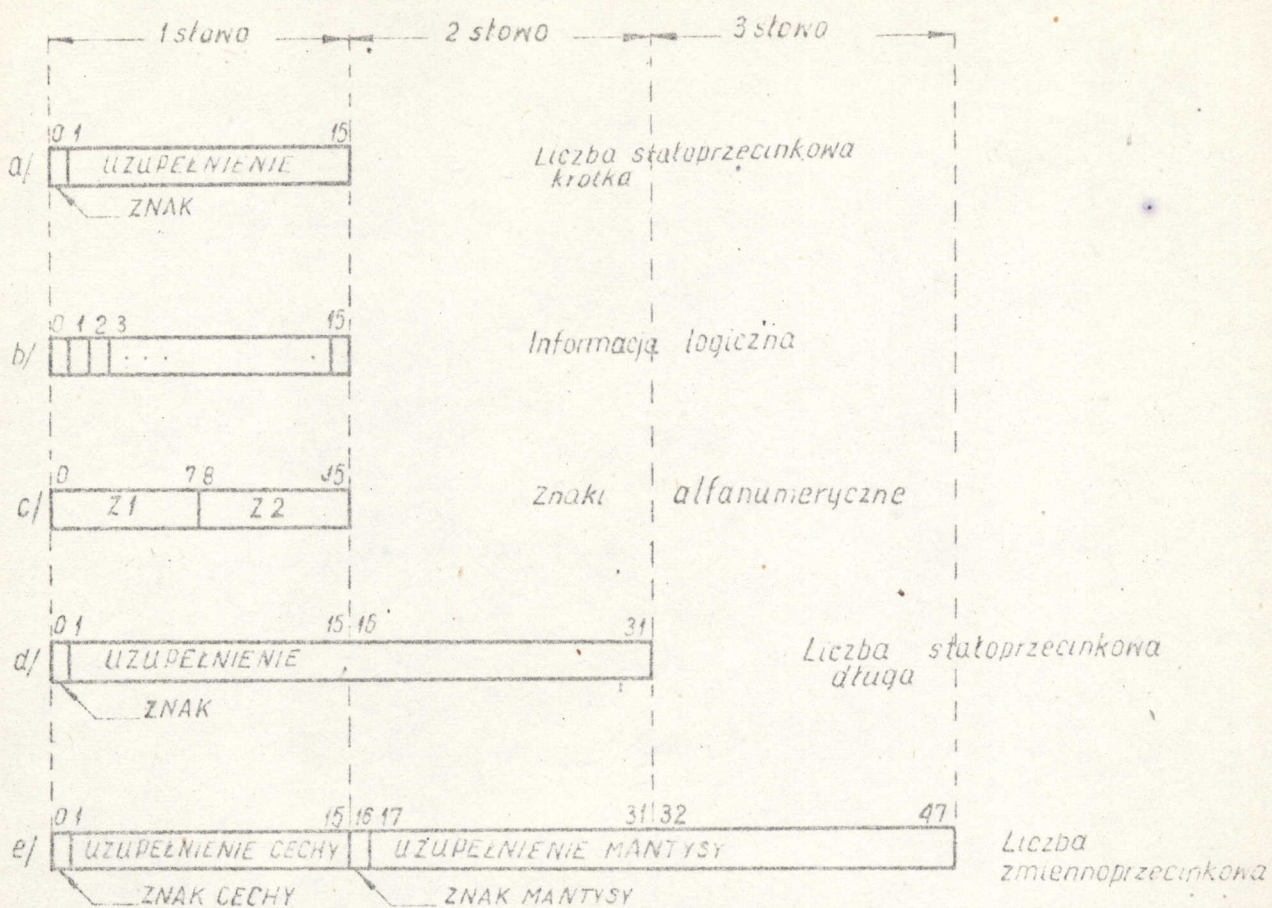
a) Rozkaz w podstawowej postaci zajmuje jedno słowo (Rys. 2a) i zawiera następujące informacje:

- KOD OPERACJI (poz. 0-4) - pole 5-bitowe pozwalające na rozróżnienie 32 rozkazów podstawowych i grup rozkazowych. W wielu przypadkach część pozostałych pól rozkazu stanowi przedłużenie kodu operacji, co pozwala na faktyczne zwiększenie pełnej ilości rozkazów do ponad 90,
- D (poz. 5) - zawiera bit pośredniego argumentu lub znak argumentu operacji lub stanowi przedłużenie kodu operacji,

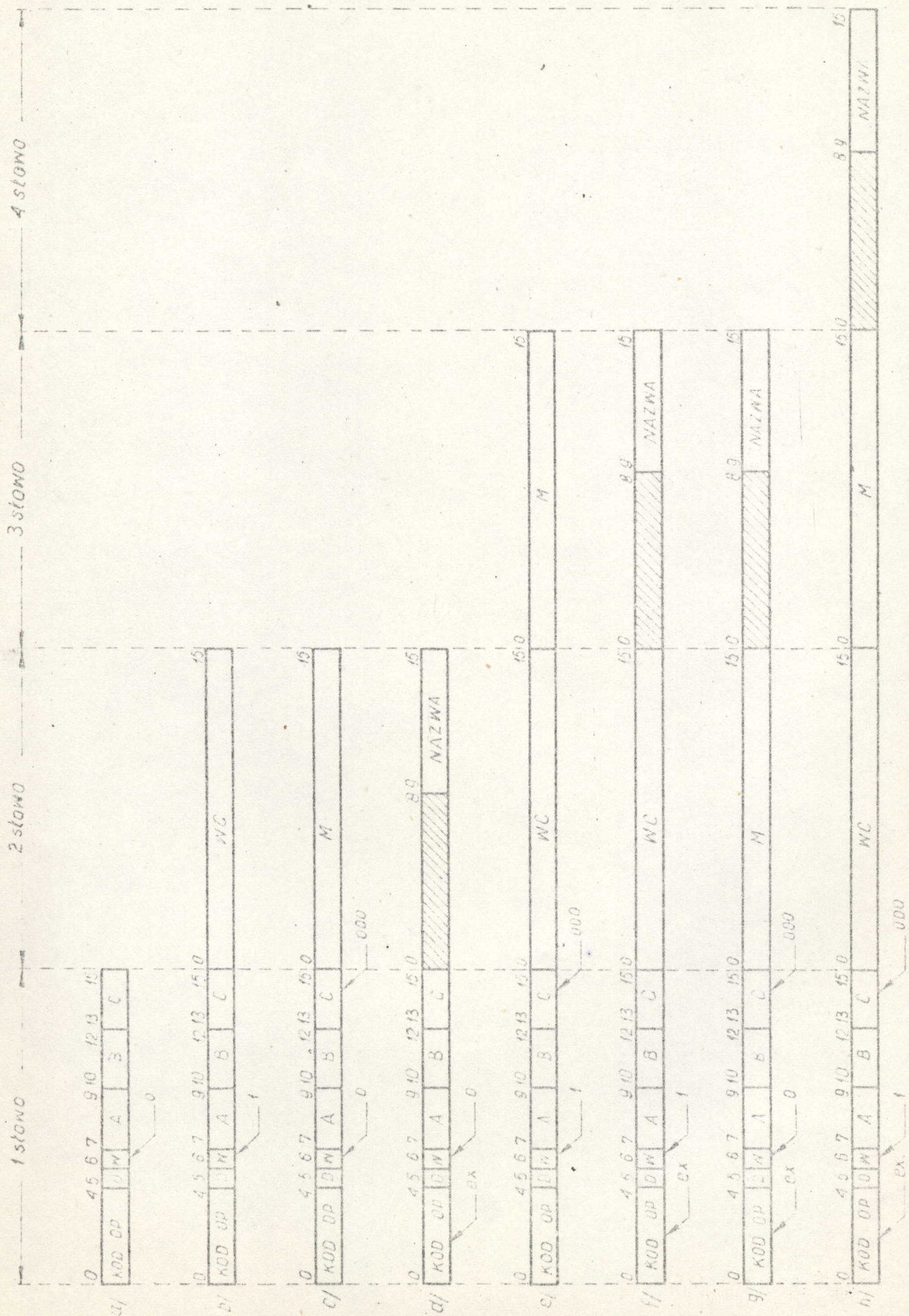
- W (poz. 6) - bit wskazujący, że rozkaz jest warunkowy,
- A (poz. 7-9) zawiera numer rejestru uniwersalnego lub stanowi przedłużenie kodu operacji,
- B (poz. 10-12) - zawiera numer rejestru indeksowego służącego do B-modyfikacji argumentu lub stanowi przedłużenie kodu operacji,
- C (poz. 13-15) - służy do wyznaczenia argumentu rozkazu lub stanowi przedłużenie kodu operacji.

W niektórych przypadkach pola D, B i C łącznie zawierają bezpośrednio 7-bitowy argument rozkazu.

- b) Rozkaz warunkowy - ($W=1$) - (Rys. 2b,e,f,h) posiada pole warunków WC, które jest informacją logiczną zajmującą następne słowo za słowem podstawowym rozkazu.
- c) Rozkaz z polem M - ($C=0$) - (Rys. 2c,e,g,h). Pole M jest liczbą stałoprzecinkową i zajmuje następne słowo za polem WC, jeśli rozkaz posiada je, lub za słowem podstawowym rozkazu w przeciwnym przypadku. Pole M zawiera pierwotny argument rozkazu.
- d) Rozkaz ex - wywołanie ekstrakodu (Rys. 2d,f,g,h) posiada pole NAZWA w następnym słowie za pozostałymi polami rozkazu. NAZWA zajmuje pozycje 9-15 tego słowa i jest całkowitą, nieujemną liczbą binarną. Pozostałe pozycje są ignorowane.



Rys 1 Postać informacji w K-202



2. SCHEMAT BLOKOWY MASZINY

Ogólny schemat blokowy maszyny przedstawiony jest na Rys. 3.

K-202 jest maszyną równoległą o słowie 16-bitowym, o sterowaniu asynchronicznym.

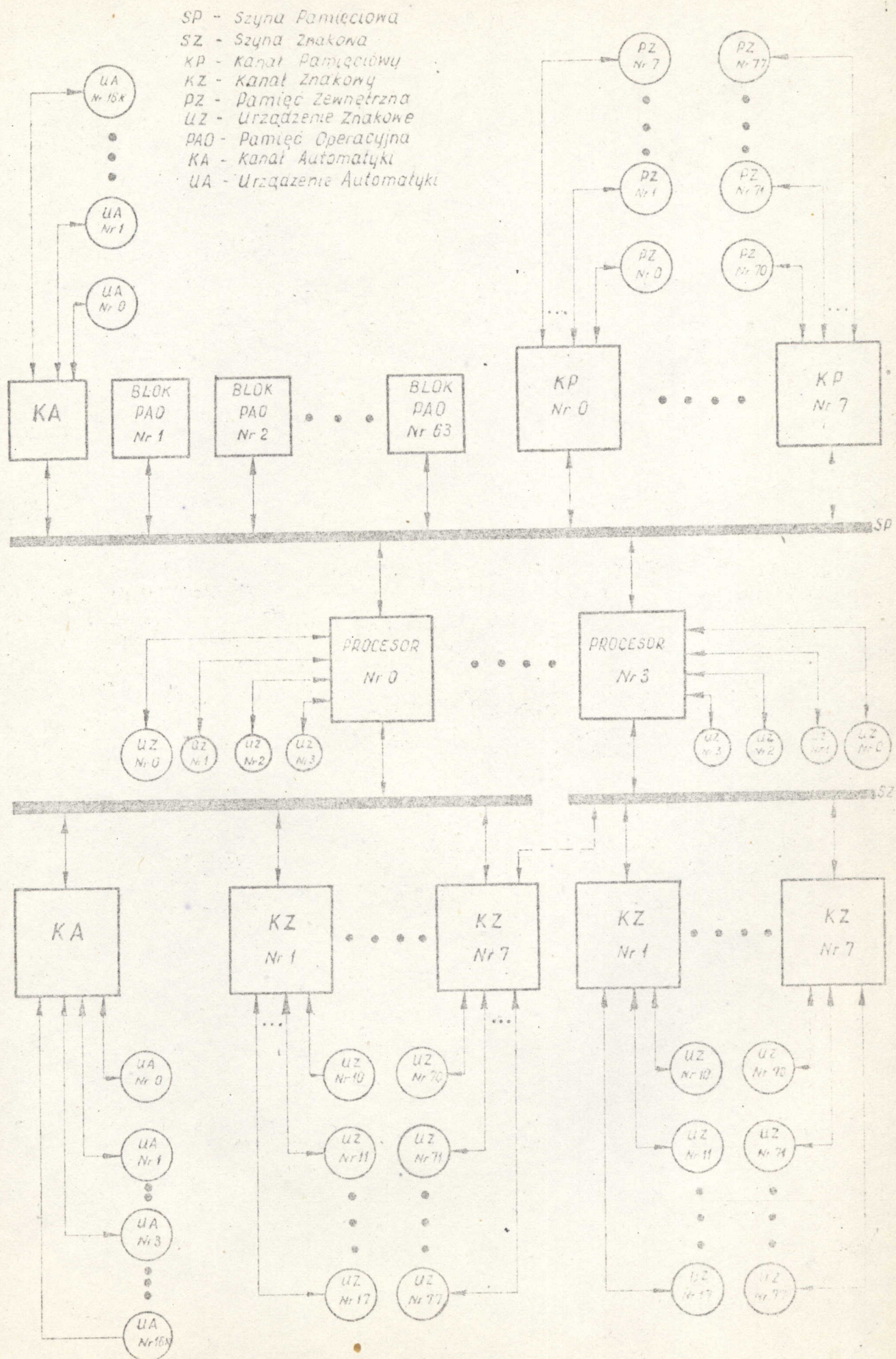
Maszyna K-202 składa się z następujących modułów:

- Procesor (jednostka centralna), podstawowa jednostka przetwarzania informacji. W skład maszyny może wchodzić do 4 procesorów.
- Bloki Pamięci Operacyjnej (PAO) w ilości do 64, numerowane od 0 do 63. Blok zerowy może mieć pojemność 4k, 8k lub 16k słów 16-bitowych i mieści się w jednostce centralnej. Dalsze bloki mogą mieć pojemność 16k, 32k lub 64k słów każdy,
- Kanały Pamięciowe w ilości do 8, służące do połączenia modułów Pamięci Zewnętrznych z Pamięcią Operacyjną i Procesorem.
- Pamięci Zewnętrzne. Do każdego Kanału można dołączyć do 8 modułów Pamięci Zewnętrznej, jak dyski, bębny, taśmy magnetyczne itp.
- Kanały Znakowe w ilości do 8, służące do połączenia Urządzeń Znakowych z Procesorem. Kanał Nr. 0 mieści się w jednostce centralnej.
- Urządzenia Znakowe - urządzenia WE-WY. Do kanału Nr. 0 można dołączyć do 4 urządzeń, do innych kanałów znakowych po 8 urządzeń We-Wy, w sumie 60 urządzeń.
Ilość urządzeń w kanale zerowym można powiększyć do 8, (druga część modułu - 4 urządzenia na zewnątrz) co razem daje 64 urządzenia We-Wy.

Pomiędzy modułami istnieją następujące standardowe połączenia:

- Interface Pamięciowy jest łączem standardowym pomiędzy Kanałami Pamięciowymi, blokami PAO i Procesorami. Do Interface'u Pamięciowego można dołączyć do 4 Procesorów, co umożliwia pracę wieloprocessorową, 64 bloki PAO, 8 kanałów Pamięciowych, oraz kanały automatyki i pracy na bieżąco - do 16k sygnałów (kanałów automatyki).
- Interface Znakowy jest łączem standardowym pomiędzy Procesorem a Kanałami Znakowymi urządzeń wejścia-wyjścia. Do Interface'u Znakowego można na zewnątrz dołączyć do 7 Kanałów Znakowych, oraz kanały automatyki i pracy na bieżąco (w czasie rzeczywistym) - do 16k sygnałów.

Każdy Interface posiada ustaloną ilość przewodów, służących do przesyłania informacji, adresów i sygnałów sterujących.



Rys. 3. Schemat blokowy maszyny K-202

3. REJESTRY I WSKAŹNIKI MASZINY

W maszynie K-202 dostępne są dla programisty lub wpływają na przebieg pracy następujące rejestry i wskaźniki:

- Rejestry i wskaźniki Procesora,
- Rejestry i wskaźniki Urządzeń Wejścia - Wyjścia
- Pamięć Operacyjna maszyny.

a) Rejestry i wskaźniki Procesora (Rys. 4a)

<u>Symbol</u>	<u>Nazwa i funkcja</u>
RO	Rejestr stanu maszyny 16-bitowy. Stanowi zbiór wskaźników decydujących o sposobie wykonywania programu. Pozycje rejestru RO oznaczają: Q (poz. 0) - wskaźnik systemu. Stan $Q=1$ odpowiada pracy programu normalnego, stan $Q=0$ - pracy Systemu Operacyjnego. Stan Q wpływa na sposób adresowania PAO, sposób wykonania rozkazów nielegalnych, oraz sposób zapisu do rejestru RO. L, E, G (poz. 1-3) - wskaźniki ustawiane w wyniku operacji porównań arytmetycznych. V (poz. 4) - wskaźnik nadmiaru, do którego wpisywane jest "1" w przypadku przekroczenia zakresu liczb przy operacjach arytmetycznych i w niektórych innych przypadkach, C (poz. 5) - wskaźnik przeniesienia, ustawiany zależnie od przeniesienia z zerowej pozycji sumatora przy operacjach arytmetycznych, Y (poz. 6) - wskaźnik przechowujący bit wychodzący poza rejestr przy rozkazach przesuwania. I (poz. 7) - maska przerw zewnętrznych. Gdy $I=0$ zgłoszenia przerw zewnętrznych są przyjmowane, ale przerwania nie są realizowane. X (poz. 8) - wskaźnik ustawiany programowo, - poz. 9-15 - wskaźniki przeznaczone do użytku programisty.

Rejestr RO może być używany w programach jako rejestr wynikowy operacji (akumulator), ale wówczas zapis do pozycji 0-7 rejestru RO przy stanie Q=1 nie jest wykonywany.

R1 - R7	Rejestry Uniwersalne - 16-bitowe, podstawowe rejestry używane do przechowywania informacji, wykonywania operacji, oraz używane jako rejestry indeksowe służące do B-modyfikacji argumentów.
IC	Licznik Rozkazów 16-bitowy, zawierający adres miejsca Pamięci Operacyjnej, z którego pobrany zostaje rozkaz.
BAR	Rejestr Numeru Bloku 8-bitowy. Pozycje 8-9 rejestru BAR zawierają numer Procesora wykorzystywany przy pracy wieloprocesorowej, pozycje 10-15 wskazują numer bloku PAO, z którym odbywa się współpraca.
BIR	Rejestr Buforowy Przerwań 8-bitowy, służący do przechowywania Znak Stanu Kanału zgłaszającego przerwanie.
P	Wskaźnik Przeskoku 1 bit, ustawiony w stan "1" powoduje, że najbliższy rozkaz nie zostaje wykonany.
IRQ	Rejestr Zgłoszeń Przerwań 32-bitowy, stanowiący zbiór 32 Wskaźników Przerwań. Zapalenie Wskaźnika powoduje przerwanie programu. Część Wskaźników odpowiada zgłoszeniom zewnętrznym i te zgłoszenia maskowane są wskaźnikiem I.

b) Rejestry i wskaźniki Urządzeń Wejścia - Wyjścia (Rys. 4b)

Każde zewnętrzne Urządzenie Znakowe posiada 8-bitowy rejestr informacji RD, z którego można pobrać lub do którego można zapisać informację rozkazem z Procesora.

Ponadto każde Urządzenie Znakowe może mieć pewną ilość wskaźników T ustawianych programowo rozkazem z Procesora.

c) Pamięć Operacyjna - PAO (Rys. 4c).

Pamięć Operacyjna składa się z bloków w ilości od 1 do 64, numerowanych od 0 do 63.

Pojemność zerowego bloku może wynosić 4K, 8K, lub 16K, a pojemność każdego z dalszych bloków 16K, 32K, lub 64K słów 16-bitowych. Czas cyklu różnych bloków PAO może być różny, w zakresie od 0,3 μ s do 2 μ s.

Określenie miejsca Pamięci dokonuje się przez podanie obu współrzędnych, to znaczy numeru bloku oraz adresu słowa wewnątrz bloku, zwanego w dalszym ciągu krótko adresem.

Adres jest 16-bitową liczbą binarną, co pozwala na zaadresowanie bloku o pojemności 64K słów. Jeżeli blok posiada mniejszą pojemność, wówczas odwołanie do miejsca o adresie większym niż pojemność bloku rozpoznawane jest jako błąd i powoduje przerwanie.

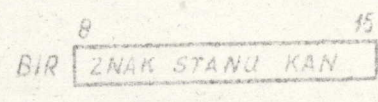
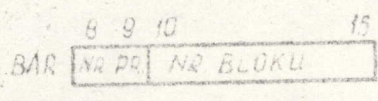
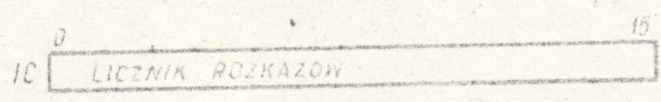
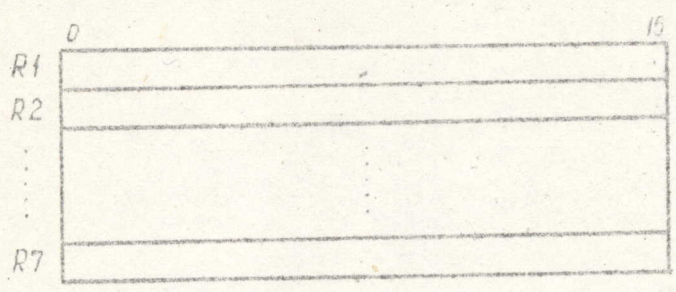
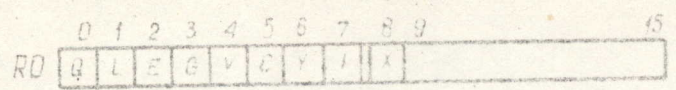
Numer bloku jest 6-bitową liczbą binarną, co pozwala rozróżnić do 64 bloków. W większości przypadków numer bloku wskazany jest zawartością pozycji 10-15 rejestru BAR.

d) Pamięci Zewnętrzne (Rys. 4d)

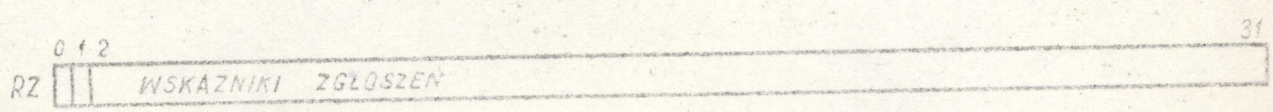
Każdy moduł Pamięci Zewnętrznej posiada 16-bitowy Rejestr Adresu (ADR) oraz 16-bitowy Rejestr Informacji DTR,

ładowane programowo rozkazem z Procesora. Ponadto każdy moduł może posiadać pewną ilość wskaźników T ustawianych programowo rozkazem z Procesora.

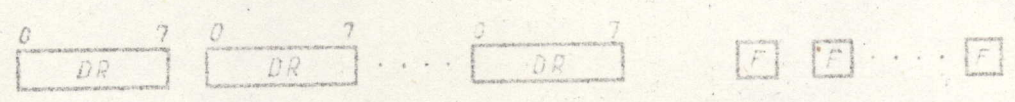
a/ Procesor



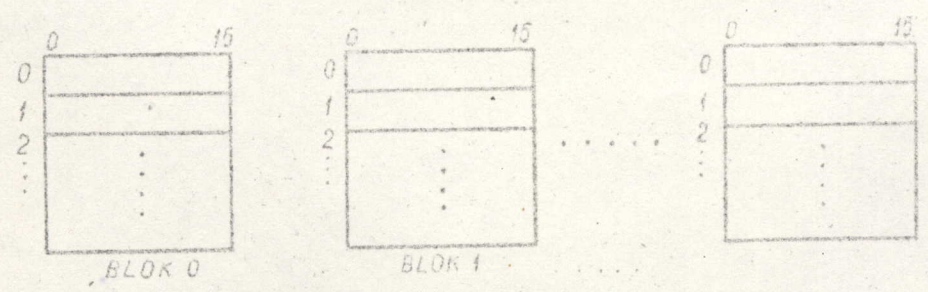
P



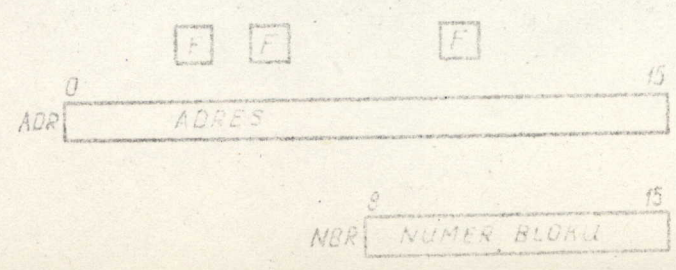
b/ Urządzenia Wejścia - Wyjścia



c/ Pamięć Operacyjna



d/ Pamięci Zewnętrzne



Rys 4. Rejestry i wskaźniki maszyny K-202

4. OGÓLNE ZASADY WYKONYWANIA ROZKAZÓW.

Pełny cykl czynności maszyny związanych z wykonaniem każdego rozkazu jest następujący:

- Pobranie rozkazu z miejsca Pamięci Operacyjnej wskazanego przez aktualną zawartość Licznika Rozkazów IC,
- Określenie efektywności rozkazu (warunek W, P i inne),
- Wyznaczenie efektywnego argumentu
- Wykonanie operacji określonej kodem operacji rozkazu.

W skład czynności związanych z wykonaniem cyklu rozkazowego wchodzi ustalenie nowej zawartości Licznika Rozkazów. W przypadku gdy nie jest to rozkaz typu skokowego i nie zachodzą przypadki szczególne, polega ono na zwiększeniu zawartości IC o długość rozkazu (liczoną w słowach).

Po zakończeniu każdego cyklu rozkazowego w zależności od warunków zaistniałych w maszynie następuje przejście do jednej z następujących czynności:

- Wykonanie następnego cyklu rozkazowego,
- Wykonanie przerwania,
- Zatrzymanie maszyny, która przechodzi wówczas w stan STOP, lub w stan CZEKAJ.

4.1 Pobranie rozkazu.

Rozkaz pobierany jest z bloku zerowego Pamięci Operacyjnej gdy wskaźnik $Q=0$, lub z bloku o numerze wskazanym zawartością pozycji 10-15 rejestru BAR gdy $Q=1$.

Adres pierwszego słowa rozkazu wskazany jest zawartością Licznika Rozkazów. Ilość pobranych słów zależy od długości rozkazu.

4.2 Określenie efektywności rozkazu.

Rozkaz jest nieefektywny, to znaczy nie zostaje wykonany w następujących przypadkach:

- a) Rozkaz nieprawidłowy. Rozpoznanie nieprawidłowego rozkazu może być spowodowane jedną z trzech przyczyn:
 - Błędny kod operacji,
 - Czwarty raz z rzędu użyty rozkaz mod (modyfikuj),
 - Rozkaz nielegalny użyty przy wskaźniku $Q=1$.

Wykrycie nieprawidłowego rozkazu powoduje wykonanie w maszynie następujących czynności:

- wpisanie "0" do wskaźnika przeskoku P,
- wyzerowanie modyfikatora (MOD),
- wpisanie "1" do odpowiedniego Wskaźnika Zgłoszenia, powodującego przerwanie wewnętrzne programu,
- natychmiastowe zakończenie rozkazu.

Po wykryciu nieprawidłowego rozkazu żadne inne warunki (pole WC, wskaźnik P itp.) nie są badane. Zawartość rejestrów programowych i pamięci maszyny nie ulega zmianie. Licznik Rozkazów w przypadku błędnego kodu operacji rozkazu zawiera adres pierwszego słowa tego rozkazu zwiększony o 1. W obu pozostałych przypadkach Licznik Rozkazów zawiera adres pierwszego słowa następnego rozkazu.

- b) Wskaźnik przeskoku P zawiera "1".
- c) Rozkaz warunkowy (bit W=1) przy zawartości rejestru stanu maszyny RO niezgodnej z zawartością pola WC. Wykrycie niezgodności następuje, jeśli istnieje pozycja taka, że RO zawiera "0" a WC zawiera "1" na tej pozycji.

W obu przypadkach b) i c) następuje wykonanie w maszynie następujących czynności:

- wpisanie "0" do wskaźnika P,
- wyzerowanie modyfikatora,
- natychmiastowe zakończenie rozkazu.

Zawartość rejestrów programowych i pamięci maszyny nie ulega zmianie. Licznik Rozkazów zawiera adres pierwszego słowa następnego rozkazu.

4.3 Wyznaczenie efektywnego argumentu.

Argument efektywny rozkazu jest to argument uzyskany z argumentu pierwotnego rozkazu przez wykonanie wszystkich wskazanych modyfikacji i jest zawsze pełną liczbą 16-bitową. Jeśli żadna z modyfikacji nie jest wykonywana, argumentem efektywnym jest argument pierwotny rozkazu wydłużony w razie potrzeby do pełnej 16-bitowej długości. Argument efektywny wyznaczany jest na kilka sposobów zależnie od postaci argumentu pierwotnego rozkazu.

Argument pierwotny rozkazu może należeć do jednego z następujących rodzajów:

Argument normalny jest liczbą 16-bitową znajdującą się w miejscu wskazanym przez pole C rozkazu:

- Jeśli $C=0$, argument stanowi zawartość pola M rozkazu,
- Jeśli $C \neq 0$, argument stanowi zawartość rejestru uniwersalnego o numerze wskazanym polem C.

Argument krótki jest liczbą 7-bitową zapisaną w notacji

znak - wartość bezwzględna i umieszczoną bezpośrednio w rozkazie:

- bit D zawiera znak argumentu,
- pola B i C zawierają łącznie 6-bitową wartość bezwzględną argumentu.

Przed użyciem krótkiego argumentu jest on przekształcany do postaci pełnej, 16-bitowej liczby stałoprzecinkowej.

MODYFIKACJE ARGUMENTU.

W maszynie K-202 możliwe są 3 różne modyfikacje argumentu:

pre-modyfikacja, B-modyfikacja i D-modyfikacja.

Są one wykonywane w podanej niżej kolejności:

- a) pre-modyfikacja możliwa jest w każdym rozkazie i ma miejsce wówczas, gdy zostanie on poprzedzony rozkazem mod (modyfikuj). Argument efektywny rozkazu mod, nazywany modyfikatorem (MOD), przechowywany jest w roboczym rejestrze Procesora. Pre-modyfikacja polega na dodaniu modyfikatora do argumentu pierwotnego rozkazu. Dodawanie jest algebraiczne z zaniechaniem ewentualnego nadmiaru. Rozkaz mod może być również poprzedzony rozkazem mod, ale czwarty kolejny rozkaz mod jest traktowany jako nieprawidłowy i nie jest wykonywany, natomiast powoduje przerwanie programu.
- b) B-modyfikacja jest możliwa w rozkazach, w których pole B oznacza numer rejestru indeksowego. Polega ona na dodaniu zawartości rejestru indeksowego do argumentu rozkazu. Dodawanie jest algebraiczne z zaniechaniem ewentualnego nadmiaru. Jako rejestry indeksowe mogą być użyte rejestry R1 - R7. Zerowa zawartość pola B oznacza brak B-modyfikacji. B-modyfikacja nie jest możliwa w rozkazach z krótkim argumentem.

c) D-modyfikacja jest możliwa w rozkazach, w których D oznacza bit argumentu pośredniego.

Polega ona na tym, że jeśli $D=1$, wówczas uzyskany w wyniku poprzednich modyfikacji argument traktowany jest jako adres argumentu efektywnego.

W celu otrzymania argumentu efektywnego dokonywany jest odczyt z bloku zerowego Pamięci Operacyjnej gdy $Q=0$, lub z bloku o numerze wskazanym zawartością pozycji 10-15 rejestru BAR gdy $Q=1$. Jeśli $D=0$, argumentem efektywnym jest argument wyznaczony w wyniku poprzednich modyfikacji. D-modyfikacja nie jest możliwa w rozkazach z krótkim argumentem, w rozkazach przesyłania grupowego (log i stg), oraz w rozkazach operacji zewnętrznych (peri, pero, meri, mero).

Zestawienie przypadków wyznaczania argumentu efektywnego.

a) Dla rozkazów o normalnym argumentcie (z wyjątkiem log, stg, peri, pero, meri, mero) możliwe są wszystkie modyfikacje.

Zestawienie dla nich przedstawia poniższa tabela:

MODYFIKACJE			C=0	Argument efektywny
MOD	B (B≠0)	D (D=1)		
n	n	n	n	$N = R(C)$
n	n	t	n	$N = S(R(C))$
n	t	n	n	$N = R(C) + R(B)$
n	t	t	n	$N = S(R(C) + R(B))$
t	n	n	n	$N = R(C) + MOD$
t	n	t	n	$N = S(R(C) + MOD)$
t	t	n	n	$N = R(C) + MOD + R(B)$
t	t	t	n	$N = S(R(C) + MOD + R(B))$
n	n	n	t	$N = M$
n	n	t	t	$N = S(M)$
n	t	n	t	$N = M + R(B)$
n	t	t	t	$N = S(M + R(B))$
t	n	n	t	$N = M + MOD$
t	n	t	t	$N = S(M + MOD)$
t	t	n	t	$N = M + MOD + R(B)$
t	t	t	t	$N = S(M + MOD + R(B))$

- b) Dla rozkazów log, stg, peri, pero, meri, mero obowiązują tylko te wiersze tabeli, w których nie występuje D-modyfikacja.
- c) W rozkazach o krótkich argumentach możliwa jest tylko modyfikacja za pomocą modyfikatora.
- d) W rozkazach nie posiadających drugiego argumentu operacja wyznaczania efektywnego argumentu nie jest wykonywana.

4.4. Wykonanie operacji określonej rozkazem.

Po wyznaczeniu argumentu efektywnego następuje przejście do wykonania operacji określonej kodem operacji rozkazu.

Maszyna K-202 posiada rozkazy dwuargumentowe, jednoargumentowe i bezargumentowe. Pierwszym argumentem operacji jest zawsze zawartość rejestru uniwersalnego wskazanego polem A rozkazu lub zawartość komórki pamięci operacyjnej. Drugim argumentem jest argument efektywny rozkazu. Rozkazy jednoargumentowe mogą posiadać tylko pierwszy argument lub tylko drugi argument.

Wykonanie operacji odbywa się zgodnie z opisem zawartym w rozdziale 5. Operacja wykonywana jest zawsze do zakończenia zgodnego z opisem. Jedynym wyjątkiem jest przypadek, gdy operacja zawiera odwołanie do Pamięci Operacyjnej lub Urządzenia Zewnętrznego, a wskazany blok Pamięci lub Urządzenie jest niedołączony do maszyny. Wówczas następuje natychmiastowe zakończenie rozkazu i zgłoszenie przerwania. Zawartość rejestru wynikowego operacji jest w tym przypadku nieokreślona.

5. LISTA ROZKAZÓW

Poniższe zestawienie zawiera wszystkie rozkazy maszyny K-202 zebrane grupami w zależności od znaczenia poszczególnych pól rozkazu.

Przy opisie każdego rozkazu obowiązują następujące zasady:

1. Jeżeli w opisie rozkazu nie jest wymieniony Licznik Rozkazów to przyjmujemy zawsze, że rozkaz ten powoduje zwiększenie zawartości IC o długość rozkazu. Ponadto wynikiem rozkazu są tylko te zmiany, które wynikają jednoznacznie z treści rozkazu.
2. Adresowanie miejsc Pamięci Operacyjnej dokonywane jest następująco:
 $S(x;y)$ oznacza zawartość komórki o adresie x w bloku Pamięci o numerze y
 $S(x)$ oznacza zawartość komórki pamięci o adresie x w bloku zerowym gdy $Q=0$, lub w bloku o numerze wskazanym pozycjami 10-15 rejestru BAR gdy $Q=1$.
3. Symbole $RO, \dots, R7, IC, S$ itp. oznaczają zawartość rejestrów programowych i miejsc Pamięci; na przykład $R(A)$ oznacza zawartość rejestru uniwersalnego o numerze wskazanym polem A rozkazu.
4. Indeksy przy symbolach rejestrów oznaczają numery pozycji; na przykład $R5[0-8, 10-15]$ oznacza "zawartość pozycji: od 0 do 8 oraz od 10 do 15 rejestru o numerze 5".
5. Symbolem argumentu efektywnego jest N i w zależności od rozkazu może oznaczać liczbę stałoprzecinkową, informację logiczną lub parametr rozkazu $WE-WY$.
6. Użyte symbole logiczne mają następujące znaczenie:
 - nad symbolem oznacza negację
 - $\wedge$ oznacza iloczyn logiczny
 - $\vee$ oznacza sumę logiczną
 - $\oplus$ oznacza różnicę symetryczną
 - $\equiv$ oznacza tożsamość
7. Przez funkcję logiczną określoną na dwóch słowach rozumiemy funkcję wykonaną na każdej pozycji niezależnie.
8. Jeśli funkcja logiczna określona na dwóch słowach występuje jako warunek, to uważamy ten warunek za spełniony, jeśli spełniony jest na wszystkich pozycjach równocześnie.

9. Napis "ust. V" oznacza, że jeśli w wyniku wykonania wskazanej operacji występuje nadmiar, do wskaźnika V wpisywana jest "1", w przeciwnym razie V pozostaje bez zmiany.
W przypadku wystąpienia nadmiaru wynik operacji może być nieprawidłowy.
10. Napis "ust. C" oznacza, że do wskaźnika C wpisywana jest wartość przeniesienia z zerowej pozycji sumatora w czasie wykonania wskazanej operacji.
11. MOD oznacza modyfikator.
12. Jeżeli RO jest rejestrem wynikowym wskazanym przez pole A, oraz w rozkazie reex, zapis do RO[0-7] przy wskaźniku Q=1 nie jest wykonywany.
13. RA oznacza przesunięcie o jeden bit w lewo, to znaczy operację $z := R[0]; R[0-14] := R[1-15];$
RV oznacza przesunięcie o jeden bit w prawo, to znaczy operację $z := R[15]; R[1-15] := R[0-14];$ gdzie z oznacza bit wychodzący poza rejestr przy przesunięciu.

LISTA ROZKAZÓW

A. Rozkazy dwuargumentowe.

Pola rozkazu oznaczają:

A - numer rejestru uniwersalnego

B - numer rejestru indeksowego

C - wskazuje argument normalny

D - bit argumentu pośredniego

<u>Skrót</u>	<u>Nazwa</u>	<u>Treść</u>
ad	Dodaj	$R(A) := R(A) + N$; ust. V, C
adc	Dodaj z przeniesie- niem	$R(A) := R(A) + N + C$; ust. V, C
su	Odejmij	$R(A) := R(A) - N$; ust. V, C
co	Porównaj	Jeśli $R(A) < N$ to $L := 1$, $E := 0$, $G := 0$; Jeśli $R(A) = N$ to $L := 0$, $E := 1$, $G := 0$; Jeśli $R(A) > N$ to $L := 0$, $E := 0$, $G := 1$;
and	Pomnóż logicznie	$R(A) := R(A) \wedge N$
or	Dodaj logicznie	$R(A) := R(A) \vee N$
orn	Pomnóż logicznie przez negację	$R(A) := R(A) \wedge \overline{N}$
clmo	Porównaj logicznie przez maskę i prze- skocz	Jeśli $(R(A) \wedge R7) = (N \wedge R7)$ to $P := 1$
clbo	Porównaj logicznie iloczyn i przeskocz	Jeśli $(R(A) \wedge N) = N$ to $P := 1$
lo	Umieść	$R(A) := N$

lom	Umieść przez maskę	$R(A) := (R(A) \wedge \overline{R7}) \vee (N \wedge R7)$
los	Umieść w pamięci	$S(R(A)) := N; R(A) := R(A) + 1$
lob	Umieść według BAR	$R(A) := S(N; BAR)$
st	Pamiętaj	$S(N) := R(A)$
stb	Pamiętaj według BAR	$S(N; BAR) := R(A)$
jpar	Skocz ze śladem w rejestrze	$R(A) := IC; IC := N$

B. Rozkazy bez pierwszego argumentu.

Pola rozkazu oznaczają:

A - przedłużenie kodu operacji

B - numer rejestru indeksowego

C - wskazuje argument normalny

D - bit argumentu pośredniego

<u>Skrót</u>	<u>A</u>	<u>Nazwa</u>	<u>Treść</u>
jp	0	Skocz	IC:=N
jpl	1	Skocz przy L	Jeśli L=1 to IC:=N
jpe	2	Skocz przy E	Jeśli E=1 to IC:=N
jpg	3	Skocz przy G	Jeśli G=1 to IC:=N
jpr	4	Skocz ze śladem	S(N):=IC; IC:=N+1
jprl	5	Skocz ze śladem przy L	Jeśli L=1 to S(N):=IC; IC:=N+1
jpre	6	Skocz ze śladem przy E	Jeśli E=1 to S(N):=IC; IC:=N+1
jprg	7	Skocz ze śladem przy G	Jeśli G=1 to S(N):=IC; IC:=N+1
mess	0	Wyślij polecenie na zewnątrz	NIELEGALNY N - zawiera polecenie wysyłane na zewnątrz; Poszczególne pozycje N oznaczają: N[0-7] - treść polecenia, N[8-9] - do późniejszego wykorzysta- nia

N[10-12] - numer Kanału Znakowego
N[13-15] - numer Urządzenia Znakowego

Jeśli rozkaz zostaje przyjęty przez Urządzenie to P:=1

lbar	1	Umieść w BAR	NIELEGALNY BAR:=S(N)[8-15]
sbar	2	Pamiętaj BAR	S(N)[0-7]:=0; S(N)[8-15]:=BAR
mod	3	Modyfikuj następny rozkaz	MOD:=N
ex	4	Wywołanie ekstrakodu	R4:=N; w:=0; S(0):=IC; S(1):=RO; IC:=S(128+NAZWA; 0); RO:=0 RO[8]:=w; RO[0-7,9-15]:=0 Powoduje przejście do podprogramu w Systemie Operacyjnym
reex	5	Wróć	IC:=S(N;BAR^(QvX)); RO[4]:=S(N+1;BAR^(QvX))[4]vRO[4]; RO[0-3,5-15]:=S(N+1;BAR^(QvX))[0-3,5-15];
ados	6	Dodaj 1 do pamięci i przeskocz przy zerze	S(N):=S(N)+1; ust. V,C; Jeśli S(N)=0 to P:=1
zs	7	Zeruj pamięć	S(N):=0
puf	-	Wykonaj operację zmiennoprzecinkową	(R1,R2,R3):=f[(R1,R2,R3),(S(N),S(N+1),S(N+2))] gdzie f - jest operacją określoną przez pole A

C. Rozkazy bez drugiego argumentu.

Pola rozkazu oznaczają:

A - numer rejestru uniwersalnego

B, C, D - przedłużenie kodu operacji

<u>Skrót</u>	<u>B</u> <u>C</u> <u>D</u>	<u>Nazwa</u>	<u>Treść</u>
stop	- 0 0	Stop	NIELEGALNY; zatrzymanie maszyny
neg	=0 1 0	Neguj	$R(A) := \overline{R(A)}$
nec	≠0 1 0	Neguj z przeniesie- niem	$R(A) := \overline{R(A)} + C$; ust. V, C.
nega	- 1 1	Neguj arytmetycznie	$R(A) := -R(A)$; ust. V, C.
chan	- 2 0	Zamień znaki	$(R(A)[0-7] := R(A)[8-15])$ i $(R(A)[8-15] := R(A)[0-7])$
lobi	- 2 1	Umieść BIR	$R(A)[0-7] := 0$; $R(A)[8-15] := BIR$
rkey	- 3 0	Czytaj klucze	$R(A) := INF$; gdzie INF oznacza informację ustawioną za pomocą kluczy
ric	- 3 1	Umieść IC	$R(A) := IC$
shl	0 4 0	Przesuń w lewo	$R(A) \ll$; $R(A)[15] := 0$; $Y := z$
shv	≠0 4 0	Przesuń w lewo i badaj	$R(A) \ll$; $R(A)[15] := 0$; $Y := z$; ust. V
shly	0 5 0	Przesuń w lewo z Y	$R(A) \ll$; $R(A)[15] := Y$; $Y := z$
shvy	≠0 5 0	Przesuń w lewo z Y i badaj	$R(A) \ll$; $R(A)[15] := Y$; $Y := z$; ust. V
shlx	0 6 0	Przesuń w lewo z X	$R(A) \ll$; $R(A)[15] := X$; $Y := z$;

shvx	+0 6 0	Przesuń w lewo z X	$R(A) \wedge$; $R(A)[15] := X$; $Y := z$; ust. V
shr	- 4 1	Przesuń w prawo	$R(A) \vee$; $R(A)[0] := 0$; $Y := z$
shry	- 5 1	Przesuń w prawo z Y	$R(A) \vee$; $R(A)[0] := Y$; $Y := z$
shrx	- 6 1	Przesuń w prawo z X	$R(A) \vee$; $R(A)[0] := X$; $Y := z$
stxa	- 7 0	Ustaw X według bitu 0	$X := R(A)[0]$
stxz	- 7 1	Ustaw X według bitu 15	$X := R(A)[15]$

Pozostałe kombinacje w polach B, C i D rozkazu są nieużywane.

D. Rozkazy z krótkim argumentem.

Pola rozkazu oznaczają:

A - numer rejestru uniwersalnego

B, C, D - krótki argument

<u>Skrót</u>	<u>Nazwa</u>	<u>Treść</u>
adt	Dodaj krótko	$R(A) := R(A) + N$; ust. V, C
adot	Dodaj krótko i przeskocz	$R(A) := R(A) + N$; ust. V, C; Jeśli $R(A) = 0$ to $P := 1$
adjt	Dodaj jedynekę i przeskocz krótko	$R(A) := R(A) + 1$; ust. V, C; Jeśli $R(A) \neq 0$ to $IC := IC + N$
cot	Porównaj krótko	Jeśli $R(A) < N$ to $L := 1, E := 0, G := 0$ Jeśli $R(A) = N$ to $L := 0, E := 1, G := 0$ Jeśli $R(A) > N$ to $L := 0, E := 0, G := 1$
lot	Umieść krótko	$R(A) := N$
lts	Umieść zawartość pamięci krótko	$R(A) := S(IC + N)$
sts	Pamiętaj krótko	$S(IC + N) := R(A)$

Uwaga: w rozkazach adjt oraz lts IC oznacza zawartość
Licznika Rozkazów po wyznaczeniu adres pierwszego
słowa następnego rozkazu.

E. Rozkazy przeskoków krótkich.

Pola rozkazu oznaczają:

A - przedłużenie kodu operacji

B,C,D - krótki argument

<u>Skrót</u>	<u>A</u>	<u>Nazwa</u>	<u>Treść</u>
jpt	0	Przeskocz krótko	$IC := IC + N$
jptl	1	Przeskocz krótko przy L	Jeśli $L=1$ to $IC := IC + N$
jpte	2	Przeskocz krótko przy E	Jeśli $E=1$ to $IC := IC + N$
jptg	3	Przeskocz krótko przy G	Jeśli $G=1$ to $IC := IC + N$
jptv	4	Przeskocz krótko przy V	Jeśli $V=1$ to $IC := IC + N; V := 0$
jptx	5	Przeskocz krótko przy X	Jeśli $X=1$ to $IC := IC + N$
jpty	6	Przeskocz krótko przy Y	Jeśli $Y=1$ to $IC := IC + N$
jpti	7	Przeskocz krótko przy I	Jeśli $I=1$ to $IC := IC + N$

Uwaga: w tych rozkazach IC oznacza zawartość Licznika Rozkazów po wyznaczeniu adresu pierwszego słowa następnego rozkazu.

F. Rozkazy przesyłania grupowego.

Pola rozkazu oznaczają:

A, D - przedłużenie kodu operacji

B - numer rejestru indeksowego

C - wskazuje argument normalny

Skrót D A Nazwa Treść

log 0 Umieść grupowo:

```

0      R1:=S(N;BAR);R2:=S(N+1;BAR)
1      R1:=S(N;BAR);R2:=S(N+1;BAR);R3:=S(N+2;BAR)
2      (R1,R2,...,R7):=(S(N;BAR),S(N+1;BAR),...,S(N+6;BAR))
3      R5:=S(N;BAR);R6:=S(N+1;BAR);R7:=S(N+2;BAR)
4      R1:=S(N);R2:=S(N+1)
5      R1:=S(N);R2:=S(N+1);R3:=S(N+2)
6      (R1,R2,...,R7):=(S(N),S(N+1),...,S(N+6))
7      R5:=S(N);R6:=S(N+1);R7:=S(N+2)

```

stg 1 Pamiętaj grupowo:

```

0      S(N;BAR):=R1;S(N+1;BAR):=R2
1      S(N;BAR):=R1;S(N+1;BAR):=R2;S(N+2;BAR):=R3
2      (S(N;BAR),S(N+1;BAR),...,S(N+6;BAR)):= (R1,R2,...,R7)
3      S(N;BAR):=R5;S(N+1;BAR):=R6;S(N+2;BAR):=R7
4      S(N):=R1;S(N+1):=R2
5      S(N):=R1;S(N+1):=R2;S(N+2):=R3
6      (S(N),S(N+1),...,S(N+6)):= (R1,R2,...,R7)
7      S(N):=R5;S(N+1):=R6;S(N+2):=R7

```