

M E R A - 1 0 0

DOKUMENTACJA EKSPLOATACYJNA SYSTEMU OPERACYJNEGO "KB"

OPIS JĘZYKA SYMBOLICZNEGO

Opracowali: Andrzej Brzozowski  
Zbigniew Daleczko  
Tadeusz Nabożny

Wałbrzych 26.02.1981 r.

# S P I S   T R E Ś C I

I.	OPIS SYSTEMU "KB" .....	3
	A/ Wstęp .....	3
	B/ Struktura pamięci MERA-100 w systemie "KB" .....	4
	C/ Wezytywanie systemu .....	5
	D/ Możliwe działania systemu .....	5
	E/ Lampki .....	6
	F/ Etykiety .....	6
	G/ Rejestry robocze .....	6
II.	OPIS INSTRUKCJI JĘZYKA SYMBOLICZNEGO .....	8
	A/ Dyrektywy .....	8
	B/ Instrukcje sterujące .....	9
	C/ Instrukcje arytmetyczne .....	11
	D/ Przesłania .....	16
	E/ Instrukcje tekstowe .....	18
	F/ Instrukcje wejścia/wyjścia .....	19
	G/ Instrukcje kasetowe .....	24
	H/ Instrukcje specjalne .....	25
III.	WPROWADZANIE KOREKT I PROGRAMÓW .....	29
IV.	PRZYKŁADY PROGRAMÓW .....	30
V.	DODATEK A: KODY ZNAKÓW I ROZKAZÓW W SYSTEMIE MERA-100 .....	33
VI.	DODATEK B: LISTA INSTRUKCJI SYSTEMU "KB" NA MERA-100 .....	34

## A/ WSTĘP

System operacyjny KB został zaprojektowany na minikomputer MERA-100 wyposażony w 16k słów 8-bitowych z jedną PK-1 i bez "floppy" dysków. Umożliwia korzystanie z 253 rejestrów roboczych o długości 16B, i wprowadzenie programu źródłowego o długości ponad 4 KB. Przy takich ograniczeniach obszar dostępny dla systemu operacyjnego nie pozwala na stworzenie języka programowania wyższego rzędu. Dlatego ograniczono się do stworzenia języka adresów symbolicznych bardzo podobnego do systemu KBD na minikomputery MERA-300. Znajomość tego języka przez programistów ułatwia znacznie przeszkolenie kadry programistów. System nie wymaga żadnych programów standardowych i firmowych napisanych w języku maszynowym. Umożliwia się programistom umieszczanie wstawek w języku maszynowym w programach źródłowych, a także bezpośrednią ingerencję w system operacyjny. System wymaga standardowego PROM-u, gdzie nie wprowadzono żadnych zmian. Podprogram skinner w assemblerze nie ma pełnej diagnostyki z uwagi na ograniczenia pamięciowe.



## B/ STRUKTURA PAMIĘCI MERA-100 W SYSTEMIE "KB"

		Numer strony	
2KB	Read	000	Rutyny standardowe
	Only	...	
	Memory	007	
14KB	Random	008	Standardowe wykorzystanie w rutynach PROM-u
	Access	009	Rejestr operacyjny
	Memory		
		010	System operacyjny
		...	
		039	
		040	Lista etykiet programu binarnego
		041	Program skompilowany
		...	
		...	
		044	Program w języku źródłowym
		...	
		...	
		048	Rejestry robocze /000-222/
	...	Bufor wejścia lub rejestry robocze /223-239/	
	062		
	063		Bufor wyjścia lub rejestry robocze /240-255/

Poszczególne fragmenty pamięci mogą mieć wspólne obszary, przykładowo podczas kompilacji programu program źródłowy może zajmować obszar rejestrów roboczych, a program skompilowany może wchodzić na obszar programu źródłowego niszcząc go podczas kompilacji, co nie przeszkadza, aby kompilacja odbyła się prawidłowo. Nigdy jednak program skompilowany nie wejdzie na obszar rejestrów roboczych, które w przypadku dłuższego programu binarnego niż 2 KB zostaną przesunięte o odpowiednią ilość stron. Niedostępne wtedy będą rejestry o większych numerach.

## C/ WCZYTYWANIE SYSTEMU "KB"

- a/ założyć kasetę systemową
- b/ nacisnąć klawisze SEL, L, /K/, /B/
- c/ wyjąć kasetę systemową
- d/ nacisnąć klawisz A

## D/ MOŻLIWE DZIAŁANIA SYSTEMU OPERACYJNEGO

- /P/ listowanie strony PAO  
po wypisie S? należy wprowadzić trzycyfrowy nr strony i nacisnąć A
- /K/ korekta w PAO  
po wypisie S? wprowadzić trzycyfrowy nr strony i nacisnąć A  
po wypisie K? wprowadzić trzycyfrowy nr komórki i nacisnąć A  
po wypisie W? wprowadzić trzycyfrowy kod i nacisnąć A
- /W/ wprowadzanie programów źródłowych do PAO  
po wypisie P= wprowadzić dwuliterową nazwę programu i dalej wprowadzać instrukcje kończąc je średnikiem, a po ostatniej instrukcji nacisnąć A
- /L/ listowanie programów źródłowych z PAO
- /I/ interpretacja programów źródłowych /assembler z listingiem/
- /E/ egzekutor programów zinterpretowanych /w języku maszynowym/
- /#/ wczytywanie programów źródłowych z kasety bibliotecznej  
wprowadzić dwuliterową nazwę programu  
wypis = OK oznacza prawidłowe wczytywanie programu  
wypis = K? oznacza niewłaściwą kasetę /bez biblioteki źródłowej/  
wypis = P? oznacza brak programu w bibliotece
- /\*/ wczytywanie programów zinterpretowanych z kasety bibliotecznej, obsługa jak w przypadku programów źródłowych
- /Z/ zapis programu źródłowego na kasecie bibliotecznej  
st - zapis programu jako pierwszego na kasecie /założenie biblioteki/  
s - zapis kolejnego programu
- /B/ zapis programu binarnego z PAO na kasecie bibliotecznej  
obsługa jak w przypadku programów źródłowych
- /X/ korekta programów źródłowych umieszczonych w PAO

Uwaga: Po pracy w trybie /\*/, /#/ , /Z/, /B/ należy ponownie nacisnąć klawisz A

W sterowaniu ogólnym system kontroluje czas przestoju, podając po 10 minutach komunikat na drukarkę.

## E/ LAMPKI

System MERA-100 znajdujący się w sterowaniu ogólnym jest sygnalizowany na lampkach 5, 6, 7, 8, 9. Przy błędach dotyczących obsługi systemu zapala się lampka FB. Przy błędach dotyczących PK-1 zapala się lampka FD. Lampkę FB gasimy naciśnięciem klawisza Ck /cancel/. Lampkę FD podobnie. Podczas współpracy z PK-1 zapala się lampka L. Podczas oczekiwania na polecenie z klawiatury przy wykonywaniu programów gaśnie lampka 9. Na lampkach 1 - 4 i 5 - 8 palą się wówczas kody dwóch ostatnich cyfr numeru etykiety, w jakiej znajduje się program. Wykonanie instrukcji deklaracji etykiety powoduje zawsze zapalenie na lampkach 1 - 8 w/w kodów.

## F/ ETYKIETY

Etykieta jest symbolem, którym oznacza się miejsce w programie. Może przybierać wartości od 001 do 123. Konieczne jest zacyklowanie podprogramu, a numery etykiet podprogramów i etykiet naczelnie deklarowanych dyrektywą dl nie mogą się powtarzać.

## G/ REJESTRY ROBOCZE

W systemie przewidziano możliwość korzystania z 253 rejestrów roboczych o numerach 000 - 255 z tym, że nie wolno korzystać z rejestrów o numerach 248, 253 i 196. Rejestrom przydzielono obszar pamięci od strony 048 do 063 z tym, że rejestry 223 - 239 są wykorzystywane przez bufor wejścia, a rejestry 240 - 255 mogą być wykorzystane jako bufor wyjścia na PK-1.



Każdy rejestr roboczy składa się z 16 bajtów, które w celach identyfikacyjnych ponumerowane zostały od strony lewej do prawej od numeru 0 do 15. Rejestr może zawierać jedną liczbę dziesiętną lub tekst /fragment tekstu/.

Jeżeli rejestr zawiera liczbę, to w bajcie o numerze 0 znajduje się znak algebraiczny liczby /" + " - kod 207, " - " - kod 206/, w ostatnim ogranicznik /kod 255/, a pozostałe bajty przeznaczone są dla cyfr /kody 207 - 0, 206 - 1, 205 - 2, 204 - 3, 203 - 4, 202 - 5, 201 - 6, 200 - 7, 199 - 8, i 199 - 9/. Rozmieszczenie poszczególnych elementów liczby przedstawia poniższy rysunek:

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

gdzie: + - znak liczby  
 C - cyfra  
 0 - ogranicznik /255/

Liczbą w rejestrze traktowana jest jako stałoprzecinkowa całkowita /bez przecinka/. W rejestrze mogą być zapamiętywane liczby ułamkowe bez przecinka, z tym że liczby tego samego rodzaju muszą zawierać jednakową liczbę miejsc ułamkowych. Jeżeli na przykład rejestr ma rejestrować liczby z dwoma znakami po przecinku, to liczby 0,25, 1,0, 2 muszą być wczytywane do rejestru w postaci: 25, 100, 200. Należy o tym pamiętać podczas wykonywania operacji arytmetycznych i wyprowadzania wyników.

Jeżeli w rejestrze znajduje się tekst, to bajty o numeracji od 0 do 14 zawierają kolejne znaki /Z/ tekstu, zaś bajt o numerze 15 - ogranicznik, jak to przedstawia poniższy rysunek:

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

Tekst w rejestrze zapisywany jest w kodzie ASCII dla MERY-100.

W jednym rejestrze może być zapisana maksymalnie 14 cyfrowa liczba dziesiętna ze znakiem lub 15 znakowy tekst /fragment tekstu/. Podczas wykonywania operacji dodawania, odejmowania i mnożenia prowadzona jest kontrola przepełnienia zawartości rejestru. Dientu-  
 alne przepełnienie jest sygnalizowane zapisaniem warunku W3 w rejestrze C.

## II. OPIS INSTRUKCJI JĘZYKA SYMBOLICZNEGO

### A. DYREKTYWY

#### 1. Deklaracja etykiety En

Format: dl En;

Instrukcja nadaje symboliczny numer miejscu, w którym występuje w programie. Numer ten nazywany jest etykietą i może przybierać wartości od 001 do 123. Odpowiednio podczas wykonywania programu zapala lampki 1 - 8.

#### 2. Podprogram En

Format: sk En;

Instrukcja ta wyznacza punkt wejścia do podprogramu. Do tego miejsca można przejść tylko za pomocą instrukcji "Wywołaj podprogram". Może przybierać wartości od 001 do 123, zapala lampki 1 - 8. Możliwe jestwołanie jednego podprogramu w podprogramie.

#### 3. Fizyczny koniec programu

Format: dz;

Instrukcja ta oznacza koniec programu dla assemblera. W czasie wykonywania programu spełnia funkcję instrukcji sh.



## B. INSTRUKCJE STERUJĄCE

## 1. Skocz do etykiety En

Format: st En;

Instrukcja powoduje przekazanie sterowania do punktu w programie oznaczonego etykieta En. Jeżeli etykieta była niezadeklarowana w programie podczas wykonania na drukarce ukazuje się komunikat "xh" i praca zostaje przerwana. Dotyczy to wszystkich instrukcji sterujących.

## 2. Gdy C=WØ skocz do etykiety En

Format: sr En;

Instrukcja powoduje - w przypadku, gdy C=WØ - przekazanie sterowania do punktu oznaczonego etykieta En. W przeciwnym wypadku sterowanie przechodzi do następnej instrukcji w programie.

## 3. Gdy C=W 2 skocz do etykiety En

Format: ms En;

Działanie instrukcji jak wyżej dla C=W2

## 4. Gdy C=W1 skocz do etykiety En

Format: wi En;

Działanie instrukcji jak wyżej dla C=W1

## 5. Gdy C=W3 skocz do etykiety En

Format: wn En;

Działanie instrukcji jak wyżej dla C=W3

## 6. Wywołaj podprogram En

Format: wo En;

Instrukcja powoduje przekazanie sterowania do podprogramu oznaczonego etykietą En. Etykieta musi być argumentem instrukcji sk. Możliwe jest dwukrotne wykorzystanie tej instrukcji bez użycia instrukcji wa.

## 7. Wróć z podprogramu En

Format: wa En;

Instrukcja powoduje powrót z podprogramu En do instrukcji następującej po tej, która wywołała podprogram En.

## 8. Wróć do sterowania ogólnego

Format: sh;

Instrukcja powoduje zakończenie realizacji programu, przewinięcie kasety /musi być założona! / i przejście do sterowania ogólnego.

## 9. Skocz do etykiety o numerze w Ru

Format: sq Ru;

Instrukcja powoduje przekazanie sterowania do punktu oznaczonego etykietą, której numer znajduje się w rejestrze Ru. Brane są tu pod uwagę trzy ostatnie bajty /12, 13, 14/ modulo 128.

## C. INSTRUKCJE ARYTMETYCZNE

1. Do  $R_m$  dodaj  $R_n$ Format: rd  $R_m$   $R_n$ ;Funkcja:  $R_m = R_m + R_n$ 

Instrukcja dodaje liczbę znajdującą się w rejestrze  $R_n$  do liczby znajdującej się w rejestrze  $R_m$ . Wynik zostaje umieszczony w rejestrze  $R_m$ , zaś liczba w rejestrze  $R_n$  zostaje niezmienną. Po wykonaniu instrukcji w rejestrze warunku C ustawiany jest:

- W0 - gdy wynik jest równy 0,
- W1 - gdy wynik jest większy od 0,
- W2 - gdy wynik jest mniejszy od 0,
- W3 - gdy wystąpił nadmiar.

Jeżeli podczas wykonywania instrukcji wystąpił nadmiar, to wynik nie jest przenoszony do rejestru  $R_m$ . Oznacza to, że liczby w rejestrze  $R_m$  i  $R_n$  zostają niezmienione. Efektem instrukcji dodawania przy wystąpieniu przepełnienia jest tylko załadowanie warunku W3 do rejestru C.

Przykład: W rejestrach o podanych niżej numerach znajdują się następujące liczby:

- w rejestrze 000 - liczba a,
- w rejestrze 001 - liczba b,
- w rejestrze 002 - liczba c,
- w rejestrze 003 - liczba 0.

Należy obliczyć wartość wyrażenia  $x = a + b + c$  i wynik zapamiętać w rejestrze 003 zachowując zawartość rejestrów od 000 do 002. Przepełnienie nie wystąpi.

rd 003 000;

rd 003 001;

rd 003 002;



2. Od  $R_n$  odejmij  $R_m$ 

Format: ro:  $R_n \quad R_m$ ;

Funkcja:  $R_n = R_n - R_m$

Liczba znajdująca się w rejestrze  $R_n$  jest odejmowana od liczby w rejestrze  $R_m$ . Wynik jest umieszczony w rejestrze  $R_n$ , jeżeli nie wystąpiło przepełnienie. Liczba w rejestrze  $R_m$  zostaje niezmienną. W rejestrze C ustawiane są warunki zgodnie z regułą podaną w opisie instrukcji dodawania.

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

rd 003 000;

ro 003 001;

ro 003 002;

## 3. Ładuj parametry

Format: rx a b;

gdzie: a - liczba obcinanych miejsc w wyniku,

b - symbol służący do zaokrąglania wyniku /0 lub 5/.

Funkcja: zapamiętanie parametrów służących do sterowania obcinaniem i zaokrąglaniem iloczynu i ilorazu /przy mnożeniu i dzieleniu/

Instrukcja jest stosowana przy mnożeniu i dzieleniu. Parametr a może przyjmować wartość od 0 do 9. Jeżeli parametr a = 0, to obcinanie nie następuje i wówczas nie ma znaczenia symbol służący do zaokrąglania. Jeżeli natomiast  $a \neq 0$ , to następuje obcięcie a najmniej znaczących cyfr wyniku, a do ostatniej obcinanej cyfry dodana zostanie cyfra b. Jeżeli b = 0, to obcinanie odbywa się bez zaokrąglania, zaś gdy b = 5 - zaokrąglanie następuje. Instrukcja musi być użyta bezpośrednio przed instrukcją  $r_n$  i  $r_i$ .

Jeżeli na przykład w wyniku mają być obcięte 2 ostatnie znaki z zaokrągleniem, to instrukcja rx ma postać:

rx 2 5;

## 4. Rn mnoż przez Rm

Format: rm Rm Rn;

Funkcja: Rn: = Rm x Rm

Instrukcja mnoży liczbę znajdującą się w rejestrze Rm przez liczbę znajdującą się w rejestrze Rm i wynik umieszcza w rejestrze Rn, jeżeli nie nastąpiło przepełnienie. Liczba w Rm zostaje niezmieniona. Po wykonaniu instrukcji w rejestrze C ustawiają się warunki zgodnie z regułą podaną w opisie instrukcji dodawania. W przypadku przepełnienia oba rejestry zostaną niezmienione. Efektem instrukcji w tym przypadku jest tylko wpisanie warunku W3 do rejestru C.

Przed wpisaniem wyniku mnożenia do rejestru Rm iloczyn jest obciążony /lub nie/ z zaokrągleniem /lub bez zaokrąglenia/ w zależności od zapamiętanych przez instrukcję rx parametrów.

Przykład: W rejestrze 001 znajduje się liczba a, którą należy pomnożyć przez liczbę b znajdującą się w rejestrze 002. Liczba a jest z dwoma znakami po przecinku, a b - z jednym znakiem. Wynik ma być z dwoma znakami po przecinku. Ponieważ wynik powstanie z trzema miejscami po przecinku, to aby go otrzymać z dwoma znakami po przecinku, należy obciążyć jeden znak, np. z zaokrągleniem.

Instrukcje wykonujące te operacje są następujące:

rx 1 5;

rm 1 2;

## 5. Rn dziel przez Rm

Format: rl Rm Rn;

Funkcja: Rn: = Rm : Rm

Instrukcja dzieli liczbę znajdującą się w rejestrze Rm przez liczbę znajdującą się w rejestrze Rm i wynik umieszcza się w rejestrze Rn. Liczba z rejestru Rm nie ulega zmianie. Po wykonaniu instrukcji w rejestrze C ustawiają się warunki zgodnie z regułą podaną w opisie instrukcji dodawania. Jeżeli usiłowano dzielić przez 0, dzielenie nie odbywa się. W tym przypadku zawartość rejestru Rm nie zmienia się, a w rejestrze C ustawia się warunek W3.

W wyniku wykonania instrukcji powstaje iloraz bez miejsc po przecinku, gdyż dzielna i dzielnik traktowane są jako liczby całkowite. Dokładność wyniku może być osiągnięta przez pomnożenie dzielnej instrukcją rl przed wykonaniem dzielenia. Jeżeli przed dzieleniem użyto instrukcji rl 001,2; to znajdującą się w reje-

strze 001 dzielna zostanie pomnożona przez  $10^2 = 100$  i wynik dzielenia będzie z 2 znakami po przecinku /bez przecinka/.

Przykład: W rejestrze 001 znajduje się liczba a określająca wykonanie planu, w rejestrze 002 liczbą oznaczającą plan. Należy obliczyć % wykonania planu z dwoma znakami po przecinku i zaokrąglić do 1 znaku po przecinku. Obliczanie z dwoma znakami po przecinku odbywa się według wzoru:

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

Instrukcje realizujące zadanie:

rl 1 4;      mnożenie przez  $10^4$   
 rx 1 5;  
 ri 1 2;      dzielenie

#### 6. Rm porównaj z Rn

Format: pw Rn Rn;

Funkcja: C: = W0 ∨ W1 ∨ W2

Instrukcja porównuje liczbę zapisaną w rejestrze Rn z liczbą zapisaną w rejestrze Rn. W wyniku porównania w rejestrze warunku C ustawiane są warunki:

- W0 - gdy liczba w Rn równa się liczbie w Rn,
- W1 - gdy liczba w Rn jest większa od liczby w Rn,
- W2 - gdy liczba w Rn jest mniejsza od liczby w Rn.

Zawartości rejestrów pozostają bez zmiany.

#### 7. Rm porównaj z zerem

Format: on Rm;

Funkcja: C: = W0 ∨ W1 ∨ W2

Instrukcja porównuje liczbę zapisaną w rejestrze Rm z zerem.

W wyniku porównania w rejestrze C ustawiane są warunki:

- W0 - gdy liczba w Rm jest równa zero,
- W1 - gdy liczba w Rm jest większa od zera,
- W2 - gdy liczba w Rm jest mniejsza od zera.

Zawartość rejestru Rm pozostaje niezmienną.



## 8. Przesuń liczbę w prawo

Format: rp Rm d;

gdzie: d - ilość pozycji cyfrowych rejestru /od 1 do 14/

Funkcja Rm: =  $Rm \times 10^{-d}$ 

Instrukcja powoduje arytmetyczne przesunięcie liczby znajdującej się w rejestrze Rm o d pozycji cyfrowych w prawo. Cyfry z prawej strony nie mieszczące się w rejestrze są gubione, zwalniane pozycje z lewej strony zapełniane są zerami. Znak liczby nie bierze udziału w przesunięciu. Działanie instrukcji równoważne jest dzieleniem zawartości rejestru przez  $10^d$  /z obcinaniem miejsc po przecinku/.

Przykład: Jeżeli w rejestrze R1 znajduje się liczba:

100000000012345

to po wykonaniu instrukcji: rp 1 2; otrzymany:

100000000000123

## 9. Przesuń liczbę w lewo

Format: rl Rm d;

gdzie: d - ilość pozycji cyfrowych rejestru /od 1 do 14/

Funkcja: Rm: =  $Rm \times 10^d$ 

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

Przykład: Jeżeli w rejestrze R1 znajduje się liczba:

101234560001234

to po wykonaniu instrukcji rl 1 5; otrzymany:

156000123400000

## D. PRZESŁANIA

## 1. Wymień zawartość rejestrów:

Format: rz Rn Rn;

Funkcja: Rn: = Rn

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

Przykład: Jeżeli zawartość poniższych rejestrów jest następująca:

R5: 10000000012345

R101: SYSTEM KB - 100

to po wykonaniu instrukcji rz5 101; otrzymamy:

R5: SYSTEM KB - 100

R101: SYSTEM KB - 100

## 2. Prześlij rejestr do rejestru wskazanego przez inny rejestr

Format: ai Rm Rn;

Funkcja: R (Rm) : = Rn

Instrukcja powoduje przesłanie zawartości rejestru Rn do rejestru, którego numer zapisany jest jako liczba w rejestrze Rm.

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

Przykład: Jeżeli zawartość poniższych rejestrów jest następująca:

R10: WARSZAWA

R50: 10000012345010

R100: 00000000012345

to po wykonaniu instrukcji ai 50 100; otrzymamy

R10: 00000000012345

R50: 10000012345010

R100: 00000000012345

## 3. Załaduj liczbę do rejestru

Format: lr Rm/± 99.....9/;

Funkcja: Rm: = ± 99.....9

Instrukcja powoduje załadowanie do rejestru Rm liczby podanej jako argument instrukcji. Liczba może się składać maksymalnie z 14 cyfr, bez kropki ani przecinka dziesiętnego. Znak algebra-

liczny /plus lub minus/ musi wystąpić zawsze na początku liczby. Jeżeli ładujemy liczbę mniejszą niż 14-cyfrową, wolne miejsca z lewej strony rejestru wypełnione są zerami.

Przykład: Jeżeli zawartość rejestru R105 jest następująca:

R105: TABLICA ROBOCZA

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

R105: 100000000012345

#### 4. Załaduj tekst do rejestru

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

gdzie /tt.....t/ nie zawiera ; i /o długości max 195 B.

Funkcja: Rm: = tt.....t

Instrukcja powoduje załadowanie do rejestru Rm tekstu podanego jako argument instrukcji. Tekst może liczyć maksymalnie 15 znaków. Wprowadzony do rejestru tekst jest dosuwany do lewego końca rejestru. Jeśli tekst liczy mniej niż 15 znaków, wolne miejsca z prawej strony rejestru wypełniane są znakiem spacji.

Przykład: Jeżeli zawartość rej. R120 jest następująca:

R120: 100000000012345

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

R120: MERA-100,????????

#### 5. Wypełnij rejestr znakiem

Format: az Rm z;

Funkcja: Rm: = zzzzzzzzzzzzzzz

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

#### 6. Prześlij rejestr wskazany przez inny rejestr do rejestru

Format am Rm Rn;

Funkcja Rm: = R (Rn)

Instrukcja powoduje przesłanie do zawartości Rm zawartości z rejestru, którego numer podany jest w Rn.



## E. INSTRUKCJE TEKSTOWE

## 1. Porównaj teksty

Format: qq Rm Rn;

Funkcja c: = W0  $\vee$  W1

Instrukcja porównuje ze sobą tekstowe zawartości rejestrów Rm i Rn. Jeżeli teksty są identyczne, umieszcza warunek W0 w rejestrze warunku C, jeżeli teksty są różne - warunek W1.

## 2. Porównaj znaki

Format: a1 Rm Z;

Funkcja c: = W0  $\vee$  W1

Instrukcja porównuje znak zapisany na pozycji nr 0 rejestru Rm ze znakiem, którego wartość w kodzie ISO podana jest jako argument instrukcji. Jeżeli znaki są identyczne, umieszcza warunek W0 w rejestrze C, jeżeli różne - warunek W1.

## F. INSTRUKCJE WEJŚCIA/WYJŚCIA

## 1. Aktywne jest wejście S

Format: we S;

Z chwilą wykonania instrukcji czytanie danych odbywać się będzie z wejścia zadeklarowanego parametrem S, gdzie S może przybierać wartości:

4 - klawiatura

5 - kaseta

Znaki czytane pobierane są zawsze z bufora wejścia znajdującego się na 62 stronie. Deklaracje wejścia zawsze kasuje zawartość bufora. Instrukcja we 5; powoduje ponadto zarezerwowanie i spozycjonowanie kasety do odczytu. Podczas realizacji czytania z klawiatury naciśnięcie klawisza Ck powoduje wyzerowanie bufora wejściowego, naciśnięcie klawisza funkcyjnego B powoduje skasowanie ostatniego znaku, naciśnięcie klawisza funkcyjnego A lub C powoduje zaakceptowanie wprowadzonej uprzednio informacji.

Instrukcje czytania cj, qt, qy, w wypadku czytania z kasety nie wymagają korzystania z instrukcji cb;

## 2. Aktywne jest wyjście S.

Format: wy S;

Pisanie danych z chwilą wykonania instrukcji odbywać się będzie na wyjściu wskazanym parametrem S. Parametr ten może przybierać następujące wartości:

5 - kaseta

8 - drukarka

9 - kaseta + drukarka

Deklaracja wy 5; lub wy 9; powoduje przewinięcie założonej kasety do znacznika logicznego końca zapisu lub do metryki taśmy, jeżeli taśma jest niezapisana. Założona kaseta musi posiadać znacznik końca zapisu lub metrykę /gdy jest niezapisana/. Informacja zapisywana kasuje pierwszym zapisem w/w bloki. Zapis na kasecie realizowany jest poprzez bufor wyjścia znajdujący się na 63 stronie. Bufor jest zerowany instrukcją pb;. Instrukcja mt; nie wymaga deklaracji wyjścia.

Pozycjonowanie kasety lub przewinięcie jej do znacznika logicznego zapisu odbywa się jedynie podczas pierwszej deklaracji wejścia/ /wyjścia w programie.

3. Czytaj liczbę do  $R_m$  a ogranicznik do  $R_n$ Format: qy  $R_m$   $R_n$ ;

Z aktywnego wejścia do rejestru  $R_m$  wczytana zostaje liczba, zaś jej ogranicznik do  $R_n$  na skrajną prawą pozycję  $R_n$ . W skład liczby wchodzi cyfry, a może wchodzić przecinek lub kropka. Znaki +, -, spacja, nowa linia powoduje zakończenie realizacji czytania liczby i ustawienie warunku  $W_0$ . Warunek  $W_2$  ustawiany jest dla pozostałych znaków kończących liczbę. Warunek  $W_1$  ustawiany jest po przeczytaniu znaku "#". Warunek  $W_3$  ustawiany może być podczas realizacji czytania z kasety, gdy przeczytana zostanie metryka taśmy lub znacznik logicznego końca zapisu. Odczytanie znaku #, lub -<sub>v</sub> powoduje przeczytanie liczby 0 i ustawia warunek  $W_0$ .

## 4. Zapamiętaj parametry drukowania

Format: ru N m;

gdzie: N - Nr cyfry, począwszy od której rozpoczyna się drukowanie liczby

m - liczba miejsc po przecinku.

Instrukcja zapamiętuje parametry określające sposób drukowania liczby z rejestru. Musi ona występować przed instrukcją pu.

N może przybierać wartości od 1 do 14, bowiem poszczególne cyfry liczby są numerowane następująco:

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

Parametr m może przybierać wartości od 0 do 12. Przy  $m = 0$  liczba jest drukowana bez przecinka. Aby liczba była drukowana prawidłowo musi być spełniona zależność:

$$14 - N > m$$



## 5. Drukuj liczbę

Format: pu Rm Z;

gdzie: Rm - numer rejestru,

Z - znak /0, x, -/, który ma być drukowany, gdy liczba  
równa się zeru.

Instrukcja drukuje zawartość rejestru Rm od zastanego położenia głowicy drukującej począwszy od cyfry o numerze N z m miejscami po przecinku, jeżeli zadeklarowano rodzaj wyjścia 008 lub 009. Za ostatnią /najmniej znaczącą cyfrą/ drukowany jest znak liczby. Jeżeli liczba jest dodatnia, drukowana jest spacja, w przypadku gdy liczba jest ujemna, drukowany jest znak "-".

Liczba drukowana bez przecinka równa zeru drukowana jest w postaci  $1^4$  - N spacji, znaku Z oraz znaku liczby. W liczbie nie równej zeru zera nieznaczące zamieniane są na spacje, następnie drukowane są cyfry znaczące i znak liczby.

W liczbie  $\neq 0$  drukowanej z przecinkiem wszystkie zera nieznaczące z wyjątkiem bezpośrednio występującego przed przecinkiem z lewej strony /które pozostaje/ zamieniane są na spacje. Pozostałe znaki drukowane są jak podano wyżej. W liczbie równej zeru drukowanej z przecinkiem /przy  $Z = 0$ / wszystkie zera nieznaczące występujące przed przecinkiem /z wyjątkiem bezpośrednio występującego z lewej strony przecinka/ zamieniane są na spacje. Zera za przecinkiem pozostają. Tak więc liczba jest drukowana w postaci spacji, zera, przecinka, zer po przecinku i znaku liczby. Liczba równa zeru drukowana z przecinkiem przy  $Z \neq 0$  drukowana jest w postaci  $1^5$  - N spacji, znaku Z oraz znaku liczby. Wszystkie bowiem zera z wyjątkiem ostatniego z prawej /w miejsce którego wpisywany jest znak Z/ oraz przecinek zamieniane są na spacje.

W przypadku, gdy zadeklarowano rodzaj wyjścia 005, liczba nie jest drukowana a tylko zapamiętywana w buforze.

Instrukcje nie bada prawidłowości parametrów drukowania. Jeżeli np. drukowana liczba jest większa niż przewiduje parametr N, to znaczy cyfry o numerach N-1, N-2 itd nie są zerami nieznaczącymi, to są one obcinane.

Ilość znaków drukowanej liczby z przecinkiem jest większa od ilości znaków liczba bez przecinka o 1 /o przecinek/. Należy to mieć na uwadze przy drukowaniu liczb bez przecinka i z przecinkiem w jednej rubryce. Aby wyrównać wydruk w stosunku do prawego marginesu, jeżeli dla liczby bez przecinka przyjęto parametr N,

to dla liczb z przecinkiem należy przyjąć parametr N+1.

Przykład: Znajdującą się w rejestrze 001 liczbę należy wydrukować począwszy od cyfry 2 z dwoma znakami po przecinku. W przypadku gdy liczba jest równa zeru, liczba ma być drukowana w postaci zera, przecinka, dwóch zer po przecinku i znaku liczby.

ru 002 002

pu 001 207

#### 6. Czytaj tekst do Rm

Format: qt Rm;

Czytany tekst jest bezpośrednio wprowadzony z buforu wejścia, a czytanie jest realizowane do zapelnienia rejestru piętnastoma znakami lub napotkania znaku nowej linii.

Gdy podczas czytania napotkany zostanie znak nowej linii ustawiany jest warunek W0, a pozostałe znaki w rejestrze są uzupełniane spacjami, gdy znak nowej linii nie zostanie napotkany ustawiany jest warunek W1, natomiast w wypadku czytania z kasy i napotkania znacznika końca logicznego zapisu ustawiany jest warunek W3.

#### 7. Pisz z Rm d początkowych znaków tekstu

Format: ph l<sub>m</sub> d;

Parametr d mówi o ilości znaków wypisywanych z rejestru l<sub>m</sub> /ewentualnie z rejestrów następných/. Parametr d nie może przekraczać 195. Tekst wypisywany na kasetę jest faktycznie przesyłany na bufor wyjścia, a zapis jest realizowany dopiero instrukcją pb;.

#### 8. Pisz znak Z

Format: wl Z;

Znak Z, którego wartość kodu ISO jest podana w instrukcji, wypisywany jest na aktywne wyjście /bufor wyjścia/.

## 9. Pisz tekst tt...t

Format: wt/tt...t/;

Żądany tekst jest wypisywany na aktywne wyjście. Długość tekstu nie może przekraczać 200 znaków i nie może zawierać średnika, kodów klawiszy funkcyjnych i znaku /-/slash/.

Tekst w wypadku wyjścia na kasetę jest odsyłany dopiero instrukcją pb, która powoduje fizyczne odsyłanie zawartości bufora wyjściowego na kasetę.

## 10. Czytaj znak do Rm

Format: cj Rm;

Znak z bufora wejściowego wprowadzany jest jako pierwszy znak w rejestrze Rm. W wypadku realizacji czytania z kasy i napotkania znacznika końca zapisu instrukcja ustawia warunek WJ.

## 11. Pisz znak z Rm

Format: zz Rm;

Pierwszy znak z rejestru Rm jest wypisywany na aktywne wyjście. Zawartość rejestru przy instrukcjach wyjścia nie ulega zmianie. W wypadku wyjścia na kasetę wypis jest nie-efektywny, dopiero instrukcja pb powoduje fizyczny zapis na kasecie.



## G. INSTRUKCJE KASETOWE

## 1. Metrykuj taśmę

Format: mt;

Instrukcja powoduje logiczne oczyszczenie kasety i zapisanie jednego bloku znacznika końca logicznego zapisu. Instrukcja powyższa nie wymaga uprzednio zadeklarowania wyjścia. Kaseeta po zametrykowaniu jest zwalniana.

## 2. Cofnij o blok

Format: cf;

Używać instrukcji można jedynie po uprzedniej deklaracji wy 5; lub deklaracji wy 5; lub wy 9; Skutkiem działania instrukcji jest cofnięcie taśmy o jeden blok lub do znaku BOT w przypadku pierwszego bloku.

## 3. Zapisz logiczny znacznik końca zapisu

Format: kz;

Instrukcja powoduje zapisanie jako kolejnego bloku znacznika końca zapisu i zwolnienie kasety po uprzednim przewinięciu. Instrukcja wymaga uprzednio zadeklarowania wy 5; lub wy 9;

## 4. Pisz blok na kasecie

Format: pb;

Instrukcja powoduje zapisanie bloku przygotowanego na stronie 63 na taśmie kasetowej. W wypadku błędnego zapisu naciśnięcie klawisza funkcyjnego Ck powoduje powtórny próbę zapisu bloku na kasecie. Instrukcję można realizować jedynie po uprzedniej deklaracji wyjścia na kasetę.

## 5. Czytaj blok z kasety

Format: cb;

Instrukcja powoduje przeczytanie jednego bloku z kasety. W wypadku wystąpienia błędu podczas odczytu naciśnięcie klawisza funkcyjnego A spowoduje ponowny odczyt, natomiast naciśnięcie klawisza B spowoduje zrealizowanie błędnego odczytu.

## H. INSTRUKCJE SPECJALNE

## 1. Dwubajtowa wstawka w języku wewnętrznym

Format: en 999 999;

Program w trakcie wykonywania potraktuje wstawkę jako rozkazy które zostaną wykonane. Wstawka nie może zawierać rozkazów 253, 248 i 196.

## 2. Zapamiętaj parametry drukowania tablicy

Format: pp Rm Rn;

gdzie: Rm - numer rejestru, w którym znajduje się pierwsza liczba podlegająca wydrukowi,

Rn - numer rejestru, w którym znajduje się ostatnia liczba podlegająca wydrukowi

Instrukcja zapamiętuje parametry niezbędne dla prawidłowego działania instrukcji pt i powinna występować bezpośrednio przed nią. Parametry te wspólnie z parametrami instrukcji pt określają wielkość tablicy i sposób wydruku.

## 3. Drukuj tablicę

Format: pt n m;

gdzie: n - liczba rubryk w drukowanej tablicy /ilość liczb w drukowanym wierszu

m - pozycja drukowania pierwszej liczby /od lewego marginesu /licząc od zera/

Działanie instrukcji zależy od parametru Rm podanego w instrukcji pt oraz parametru n instrukcji pp.

Przy  $Rm < n$  drukowany jest ciąg liczb /zawartość rejestrów od Rm.do Rn/ w jednej linii począwszy od pozycji m /licząc od zera/. Drukowany jest więc wiersz lub jego część, co jest szczególnym przypadkiem tablicy /tablica składająca się z n rubryk i 1 wiersza/.

W przypadku, gdy  $Rm \geq n$  od pozycji m drukowana jest tablica składająca się z n rubryk. Liczba wierszy tablicy określana jest wzorem:

$$\frac{Rn - Rm + 1}{n}$$

gdz wyrażenie dzieli się bez reszty. Jeżeli wystąpi reszta liczba wierszy tablicy jest większa o 1. Ostatni drukowany wiersz

będzie pełny, gdy podane wyrażenie dzieli się bez reszty lub niepełny w przypadku wystąpienia reszty.

W przypadku, gdy  $n = 1$  wszystkie liczby /zawartości rejestrów od Ru do Ru/ drukowane są od pozycji m w jednej rubryce.

Drukowanie poszczególnych liczb odbywa się zgodnie z parametrem N podanym w instrukcji ru. Liczby równe zero drukowane są w postaci odpowiedniej ilości spacji, "-" i znaku liczby. Sposób drukowania jest jednokowy dla wszystkich liczb tablicy.

W celu uzyskania odpowiedniej szerokości rubryk należy dobrać odpowiednio parametr N w instrukcji ru.

Przykład: Ciąg podanych niżej instrukcji przy podanych zawartościach rejestrów wydrukuje tablicę w postaci:

ru 009 000

pp 000 007

pt 003 000

000 - 00000000123456

001 - 10000000999999

002 - 00000000000001

003 - 00000000000010

004 - 00000000000000

005 - 00000000001234

006 - 00000000012345

007 - 00000000067890

23456 99999 - 1

10 - 1234

12345 67890

Zgodnie z parametrami drukowana jest zawartość rejestrów od 000 do 007 począwszy od cyfry Nr 9 bez miejsc po przecinku od pozycji 0 w trzech rubrykach. Liczba wierszy jest równa:

$$\frac{7 - 0 + 1}{3} = 2$$

a ponieważ 8 nie dzieli się przez 3 bez reszty, liczba wierszy równa się 3 z tym, że wiersz 3 jest niepełny.

Jeżeli zadeklarowano rodzaj wyjścia 008 tablica jest drukowana w sposób podany wyżej. Jeżeli natomiast zadeklarowano rodzaj wyjścia 005, poszczególne liczby tablicy ładowane są tylko do bufora. Przy zadeklarowaniu rodzaju wyjścia 009 liczby są drukowane i ładowane do bufora.



## 4. Ładuj parametry dodawania /odejmowania/

Format: pd Rm Rn;

gdzie: Rm - numer rejestru, w którym znajduje się pierwsza liczba podlegająca dodawaniu /odejmowaniu/

Rn - numer rejestru, w którym znajduje się ostatnia liczba podlegająca dodawaniu /odejmowaniu/

Instrukcja zapamiętuje parametry niezbędne do prawidłowego działania instrukcji ra i rb. Instrukcja pd powinna występować bezpośrednio przed tymi instrukcjami. Parametry instrukcji pd określają adresy oraz ilość dodawanych /odejmowanych/ liczb. Parametr Rn decyduje o przerwaniu dodawania /odejmowania/.

## 5. Dodaj zawartość rejestrów od Rm do Rn do zawartości rejestrów od Rv

Format: ra n Rv;

gdzie: Rv - numer rejestru początkujący obszar, do którego należy dodać zawartość rejestrów od Rm do Rn

n - liczba określająca, ile razy należy zwiększać o 1 adres rejestru Rv /przy n = 1 nie zwiększać/

Działanie instrukcji zależy od parametru n w instrukcji ra oraz parametru Rn w instrukcji pd. We wszystkich przypadkach dodawania zawartość rejestrów od Rm do Rn nie ulega zmianie.

Przy n=1 zawartość rejestrów od Rm do Rn dodawana jest do zawartości rejestru Rv, gdzie umieszczony jest wynik.

Przykład: Należy dodać zawartość rejestrów od 001 do 015 do rejestru 16 i wynik zapamiętać w rejestrze 016.

pd 001 015;

ra 001 016;

Przy  $n \neq 1$  działanie instrukcji zależy od Rn i n. Przy  $Rn \leq n$  zawartość rejestrów od Rm do Rn dodawana jest do zawartości rejestrów od Rv do  $Rv+Rn-Rm+1$  //Rv/ := /Rv+Rn/, /Rv+1/ := /Rv+1/+ /Rn-1/ itd

lub tablicy jednowymiarowej rozpoczynającej się od rejestru Rm do innego wektora lub tablicy jednowymiarowej zaczynającej się od rejestru Rv.

Przykład: Zawartość rejestrów od 001 do 025 należy dodać do zawartości rejestrów od 051 do 075.

pd 001 025;

ra 025 051;

Parametr n w instrukcji ra w tym przypadku może przybierać wartości od 025 do 255.

Jeżeli  $Rn > n$  zawartość rejestrów od Rm do  $Rm+n-1$  dodawana jest do zawartości rejestrów od Rv do  $Rv+n-1$ , następnie zawartość rejestrów od  $Rm+n$  do  $Rm+2n-1$  do zawartości rejestrów od Rv do  $Rv+n-1$  itd. Po każdym dodawaniu adres rejestru dodawanego /Rv/

jest zwiększany o 1, zaś adres rejestru R<sub>w</sub> jest zwiększany o 1 n razy, po czym przywracany jest ponownie adres R<sub>w</sub>. Takie dodawanie można określić jako dodawanie n ciągów do jednego ciągu zaczynającego się od rejestru R<sub>w</sub>. Dodawanie jest przerywane po osiągnięciu rejestru R<sub>n</sub>.

Przykład: W rejestrach od 001 do 020 zapamiętany jest ciąg liczb stanowiący tablicę składającą się z 5 rubryk po 4 wiersze. Zawartość rejestrów od 021 do 040 jest wyzerowana. Należy obliczyć wiersz "ogółem" i zapamiętać począwszy od rejestru 021.

pd 001 020;

ra 005 021;

Uwaga: Przy każdym dodawaniu badany jest wynik. Jeżeli nastąpiło przepełnienie, wynik dodawania, przy którym to przepełnienie wystąpiło nie jest zapamiętywany, a dodawanie jest przerywane. Pod c ładowany jest wj.

#### 6. Odejmij zawartość rejestrów od R<sub>n</sub> do R<sub>m</sub> od zawartości rejestrów od R<sub>w</sub>

Format: rb n R<sub>w</sub>;

gdzie: R<sub>w</sub> - numer rejestru początkujący obszar, od którego należy odjąć zawartość rejestrów od R<sub>m</sub> do R<sub>n</sub>,

n - liczba określająca, ile razy należy zwiększać o 1 adres rejestru R<sub>w</sub>

Działanie instrukcji jest analogiczne jak instrukcji ra.

Przykład: Zawartość rejestrów od 001 do 015 należy odjąć od rejestru 016 i tam zapamiętać wynik

pd 001 015;

rb 001 016;

Przykład: Zawartość rejestrów od 001 do 025 należy odjąć od zawartości rejestrów od 051 do 075

pd 001 025;

rb 025 051;

#### 7. Zeruj rejestry od R<sub>m</sub> do R<sub>n</sub>

Format: zr R<sub>m</sub> R<sub>n</sub>;

Instrukcja powoduje wyzerowanie rejestrów o numerach R<sub>m</sub> do R<sub>n</sub> włącznie.



### III. WPROWADZANIE KOREKT I PROGRAMÓW

Programy źródłowe wprowadzamy z klawiatury. Wprowadzamy instrukcję po instrukcji do średnika, po którym automatycznie wykonuje się nowa linia. Po ostatniej instrukcji nacisnąć należy klawisz funkcyjny A, wówczas program wróci do sterowania ogólnego. W wypadku instrukcji zeroargumentowych /dz, sh, mt, cf, kz, pb, cb/ po wprowadzeniu instrukcji wprowadzamy średnik.

Wprowadzając instrukcje jednoargumentowe po napisaniu instrukcji należy nacisnąć spację i wprowadzić argument z ogranicznikiem /;/. Dotyczy to instrukcji: dl, sk, st, sr, ms, wi, wn, wo, wa, sq, sn, we, wy qt, wl, oj, zz. W wypadku pisania instrukcji dwuargumentowych /rd, ro, rx, rm, ri, pw, rp, rl, rz, ai, am, az, qq, al, qy, ru, pu, ph, en, ra, rb, pp, pt/ po wprowadzeniu kodu instrukcji należy nacisnąć spację, wprowadzić pierwszy argument z ogranicznikiem spacją, a następnie drugi argument ze średnikiem jako ogranicznikiem drugiego argumentu. Pisząc instrukcję wt bezpośrednio po podaniu kodu instrukcji naciskamy klawisz /-/slash/ i wprowadzamy tekst kończąc go także znakiem /po którym koniecznie należy wprowadzić średnik. Instrukcje lr i ts wprowadzamy pisząc po podaniu kodu instrukcji spację, argument z ogranicznikiem /slash/ wprowadzając liczbę /koniecznie musi wystąpić znak na początku liczby/ lub tekst kończąc wprowadzanie znakiem slash i średnik.

Korekty programu źródłowego należy dokonać bezpośrednio po wprowadzeniu z klawiatury lub wczytaniu z kasety. Po wypisie instrukcji naciśnięcie klawisza funkcyjnego A powoduje akceptację instrukcji, naciśnięcie klawisza funkcyjnego C spowoduje skasowanie instrukcji, naciśnięcie klawisza funkcyjnego B umożliwia wprowadzenie wstawki w programie, którą kończymy naciśnięciem klawisza funkcyjnego /♦/ Naciśnięcie po wprowadzeniu instrukcji ponownie klawisza B powoduje wprowadzenie dalszego ciągu wstawki. Po zaakceptowaniu ostatniej instrukcji należy nacisnąć klawisz funkcyjny /\*/ i wprowadzić nazwę programu. Nowa nazwa wymagana jest ze względu na to, iż na kasecie bibliotecznej mogą być co prawda dwa programy o tej samej nazwie, lecz czytając program zawsze czytany jest pierwszy program posiadający zadeklarowaną nazwę.



## IV. Przykłady Programów.

## Program liczenia pierwiastka kwadratowego

z dokładnością czterech miejsc po przecinku.

```

SS
wo 004 ;
wy 008 ;
dl 001 ;
lr 102 /+000000000000002/;
lr 105 /+000000000000000/;
lr 106 /+000000000000040/;
qy 100 101 ;
wi 088 ;
rz 101 100 ;
rz 104 100 ;
rl 104 007 ;
dl 077 ;
wo 011 ;
rd 105 102 ;
pw 105 106 ;
sr 012 ;
st 077 ;
sk 011 ;
rz 103 100 ;
rl 103 007 ;
rx 000 000 ;
ri 103 101 ;
rd 104 103 ;
rx 000 000 ;
ri 104 102 ;
rz 101 104 ;
rp 101 007 ;
wa 011 ;
dl 012 ;
wl 010 ;
wt /Pierwiastek z liczby/;
ru 001 006 ;
pu 100 045 ;
wt /wynosi/;
ru 001 010 ;
pu 104 045 ;
st 001 ;
dl 088 ;
sh ;
dz ;
042 138
045 064

```

```

1,000000 2,000000 3,000000
4,000000 5,000000 4,000000
7,000000 8,000000 9,000000

```

Pierwiastek z liczby	1,000000	wynosi	1,0000000103
Pierwiastek z liczby	2,000000	wynosi	1,4144261058
Pierwiastek z liczby	3,000000	wynosi	1,7321035080
Pierwiastek z liczby	4,000000	wynosi	2,0000039669
Pierwiastek z liczby	5,000000	wynosi	2,2361384926
Pierwiastek z liczby	4,000000	wynosi	2,0000039669
Pierwiastek z liczby	7,000000	wynosi	2,6461697850
Pierwiastek z liczby	8,000000	wynosi	2,8288546793
Pierwiastek z liczby	9,000000	wynosi	3,0000079305

MT=OK

MT Program motrykowania kasety

mt ;  
 dz ;  
 041 011  
 044 013

ZD=OK

ZD Program zapisu danych na kasocie

ZD  
 wo 004 ;  
 wy 005 ;  
 dl 001 ;  
 cj 000 ;  
 al 000 010 ;  
 sr 002 ;  
 zz 000 ;  
 al 000 035 ;  
 sr 003 ;  
 st 001 ;  
 dl 002 ;  
 zz 000 ;  
 pb ;  
 st 001 ;  
 dl 003 ;  
 pb ;  
 kz ;  
 sh ;  
 dz ;  
 041 139  
 044 115

## BIBLIOGRAFIA:

- 1/ Yachan CHu hu "Organizacja i mikroprogramowanie maszyn cyfrowych" WNT Warszawa 1979 r.  
297
- 2/ Boris Beizer "Organizacja systemów komputerowych" PWN " Warszawa 1979 r.
- 3/ System MERA-100 Podręcznik Programowania "MERA-BLONIE" Warszawa 1979 r.

Program listowania danych z kasety

LD=OK

LD  
 we 005 ;  
 wy 008 ;  
 dl 001 ;  
 cj 000 ;  
 al 000 035 ;  
 sr 002 ;  
 zz 000 ;  
 st 001 ;  
 dl 002 ;  
 sh ;  
 dz ;  
 041 082  
 044 067

## BIBLIOGRAFIA:

- 1/ Yachan CHu "Organizacja i mikroprogramowanie maszyn cyfrowych" WNT Warszawa 1979 r.
- 2/ Boris Beizer "Organizacja systemów komputerowych" PWN Warszawa 1979 r.
- 3/ System MERA-100 Podręcznik programowania "MERA-BLONIE" Warszawa 1979 r.

WY  
 we 004 ;  
 wy 008 ;  
 dl 001 ;  
 zr 000 200 ; Przykładowy program wykorzystania instrukcji  
 dl 002 ; specjalnych zr, pt, ra.  
 lr 201 /+000000000000001/;  
 lr 202 /+000000000000201/;  
 dl 003 ;  
 ro 202 201 ;  
 cn 202 ;  
 sr 004 ;  
 ai 202 202 ;  
 st 003 ;  
 dl 004 ;  
 ru 002 002 ;  
 pp 000 119 ;  
 pt 005 010 ;  
 vl 010 ;  
 wt /=====/  
 wt /

OGDLEM / ;  
 zr 201 206 ;  
 pd 000 119 ;  
 ra 005 201 ;  
 ru 002 002 ;  
 pp 201 205 ;  
 pt 005 010 ;  
 sh ;  
 dz ;  
 042 090  
 045 046

-	-,01	-,02	-,03	-,04
-,05	-,06	-,07	-,08	-,09
-,10	-,11	-,12	-,13	-,14
-,15	-,16	-,17	-,18	-,19
-,20	-,21	-,22	-,23	-,24
-,25	-,26	-,27	-,28	-,29
-,30	-,31	-,32	-,33	-,34
-,35	-,36	-,37	-,38	-,39
-,40	-,41	-,42	-,43	-,44
-,45	-,46	-,47	-,48	-,49
-,50	-,51	-,52	-,53	-,54
-,55	-,56	-,57	-,58	-,59
-,60	-,61	-,62	-,63	-,64
-,65	-,66	-,67	-,68	-,69
-,70	-,71	-,72	-,73	-,74
-,75	-,76	-,77	-,78	-,79
-,80	-,81	-,82	-,83	-,84
-,85	-,86	-,87	-,88	-,89
-,90	-,91	-,92	-,93	-,94
-,95	-,96	-,97	-,98	-,99
1,00	1,01	1,02	1,03	1,04
1,05	1,06	1,07	1,08	1,09
1,10	1,11	1,12	1,13	1,14
1,15	1,16	1,17	1,18	1,19

=====

OGDLEM

13,80

14,04

14,28

14,52

14,76



DODATEK A

KODY ZNAKÓW I ROZKAZÓW W SYSTEMIE MERA-100

Legenda:

- 1/ kod w ISO /International Organisation for Standardization/
- 2/ mnemonika polska kodu operacji języka wewnętrznego minisystemu MERA-100
- 3/ znaki specjalne klawiatury
- 4/ reprezentacja znaku
- 5/ kod w ASCII /American Standard Code for Information Interchange/

1	2	3	4	5	1	2	3	4	5
000	R1→H		NUL	255	064	Kz6		V	191
001	TIME	A	SOH	254	065	Pz5		A	190
002	K→R1	B	STX	253	066	Kz5		B	189
003		C	ETX	252	067	Pz5		C	188
004	"+"		EOT	251	068	Kz4		D	187
005			ENQ	250	069	Pz4		E	186
006		L	ACK	249	070	Kz3		F	185
007			BEL	248	071	Pz3		G	184
008	R1→A	LK	BS	247	072	Kz2		H	183
009		HT	HT	246	073	Pz2		I	182
010		LF	LF	245	074	Kz1		J	181
011		S	VT	244	075	Pz1		K	180
012	"_"	ST	FF	243	076			L	179
013			CR	242	077	KAC		M	178
014		SO	SO	241	078	PAC		N	177
015			CI	240	079	KI→R1		O	176
016			DLE	239	080			P	175
017			DC1	238	081			Q	174
018			DC2	237	082			R	173
019	Z→R1		DC3	236	083			S	172
020	Z + 1		DC4	235	084	KZ12		T	171
021	PO→Z		NAK	234	085	PZ12		U	170
022	PO→V		SYN	233	086	KZ11		V	169
023	PO→R1		ETB	232	087	PZ11		W	168
024	ADR + 1	OO	CAN	231	088	KZ10		X	167
025	POWRÓT	CK	EN	230	089	PZ10		T	166
026	PODR.	BS	SUH	229	090	Kz9		Z	165
027	SK . A > B		ESC	228	091	Pz9		2	164
028	SK . A < B		FS	227	092	Kz8		L	163
029	SK . B = V		GS	226	093	Pz8		á	162
030	SK . A = B		RS	225	094	Kz7		š	161
031	SKOK		US	224	095	Pz7		š	160
032	R1→DH		Sp	223	096			š	159
033	R1→P		!	222	097			š	158
034	P→R1		!	221	098			b	157
035			#	220	099			c	156
036			0	219	100			d	155
037			1	218	101			e	154
038			2	217	102			f	153
039			3	216	103	CRS		g	152
040	P		4	215	104			h	151
041	P - 1		5	214	105			i	150
042	P + 1		6	213	106	RS		j	149
043	PADR		+	212	107	AR		k	148
044	B - 1		3	211	108	STV		l	147
045	B + 1		-	210	109	ODE		m	146
046	BADR		/	209	110	NAD		n	145
047	B		.	208	111	ROZ		o	144
048	LLRC		0	207	112	KDP		p	143
049	PRZEJŁCZ		1	206	113	DP→X		r	142
050	P . CZT		2	205	114	PDP		s	141
051	K . PSZ		3	204	115	R1→DP		t	140
052	P . PSZ		4	203	116	INT.		u	139
053	PRZ		5	202	117	X→R1		w	138
054	KRS		6	201	118			x	137
055	PRS		7	200	119			y	136
056	KRW		8	199	120			z	135
057	PRW		9	198	121	ZER . PK			134
058	KRP		!	197	122	K . DAN			133
059	PRP		!	196	123	ST . SZCZ			132
060	KPO		<	195	124	STAT			131
061	PPO		>	194	125	R1→PK			130
062	K . REZ		>	193	126	PK→R1			129
063	P . REZ		?	192	127	LRC			128



## LISTA INSTRUKCJI SYSTEMU "KB" NA MERE-100

<b>A. DYREKTYWY</b> dl Et; Deklaracja etykiety ek Et; Deklaracja podprogramu dz; Fizyczny koniec programu	<b>E. INSTRUKCJE TEKSTOWE</b> qq Rn Rn; Porównaj teksty w lba i lbn zl Rn Z; Porównaj pierwszy znak z lba ze znakiem Z
<b>B. INSTRUKCJE STERUJĄCE</b> st Et; Skocz do Et sr Et; Skocz do Et gdy W0 sa Et; Skocz do Et gr: W1 vl Et; Skocz do Et gdy W2 wa Et; Skocz do Et gdy W3 wo Et; Wołaj podprogram Et wa Et; Wróć z podprogramu Et sq Rn; Skocz do Et o numerse w lba sh; Stop /powrót do S0/	<b>F. INSTRUKCJE WEJŚCIA/WYJŚCIA</b> wa S; Deklaruj rodzaj wejścia 4 - klawiatura 5 - kasetta wy S; Deklaruj rodzaj wyjścia 5 - kasetta 8 - drukarka 9 - kasetta + drukarka qy Rn Rn; Czytaj liczbę do lba a ogra- nicznik do lbn ru m n; Przygotowanie do pisania liczb m - od której pozycji pisać n - ile cyfr po przecinku pu Rn t; Pisz liczbę z lba Zamiast liczby pisz t $t = \begin{cases} x \\ 0 \end{cases}$ jeśli liczba w lba=0 qt Rn; Czytaj tekst do lba ph lba d; Pisz z lba d początkowych zna- ków tekstu vl z; Pisz znak Z wt/tt...tt/; Pisz tekst tt...tt oj lba; Czytaj znak do lba sz Rn; Pisz znak z lba
<b>C. INSTRUKCJE ARYTMETYCZNE</b> rd lba lbn; Dodaj lba do lbn ro lba lbn; Odejmij lba od lbn rx a b; Ładuj parametry do dzielenia i mnożenia a: ile cyfr obciąż w wyniku b: $\begin{cases} 5 - \text{zaokrąglić wynik} \\ 0 - \text{bez zaokrąglenia} \end{cases}$ ra lba lba; Mnoż lba przez lba ri lba lba; Podziel lba przez lba pw lba lba; Porównaj liczby w lba i lba ou lba; Porównaj liczbę w lba z zerem rp lba d; Przesuń liczbę w lba w prawo o d-znaków rl lba d; Przesuń liczbę w lba w lewo o d-znaków	<b>G. INSTRUKCJE KASETOWE</b> mt; Pisz metrykę taśmy /bez deklaracji wyjścia/ of; Cofnij o blok ka; Pisz znacznik końca zapisu ph; Pisz blok ob; Czytaj blok
<b>D. PRZESŁANIA</b> ra lba lba; Prześlij zawartość lba do lba al lba lba; Prześlij zawartość lba do rejestru o numerze podanym w lba au lba lba; Do rejestru lba prześlij za- wartość rejestru o numerze w lba lr lba/ 99...9/; Umieść liczbę w lba ta lba/tt...t/; Umieść tekst w lba az Rn Z; Zapelnij lba znakiem Z	<b>H. INSTRUKCJE SPECJALNE</b> en 999 999; Dwubajtowa wstawka zr Rn Rn; Zeruj rejestry od lba do lbn pd Rn Rn; Ładuj parametry dodawania /-/ ra n Rn; Dodaj rejestry do rejestrów rb n Rn; Odejmij rejestry od rejestrów pp lba lba; Pamiętaj parametry tablicy pt lba m; Pisz tablicę

## OBJAŚNIENIA

lba, lbn - numery rejestrów roboczych 0 - 255 /bez 248, 253 i 196/

Et - numer etykiety 1 - 123

d - ilość znaków 1 - 247

Z - kod ISO znaku /bez 248, 253, 196/

/tt...tt/- tekst nie zawierający znaków /i; o długości do 247 znaków.