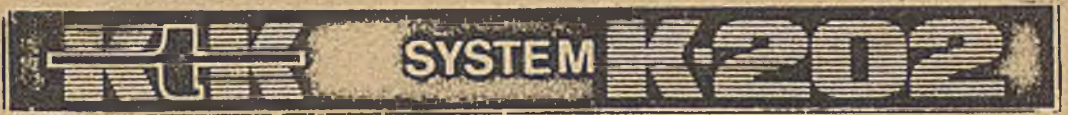


J. Hanczyk



ZAKŁAD DOŚWIADCZALNY MINIKOMPUTERÓW

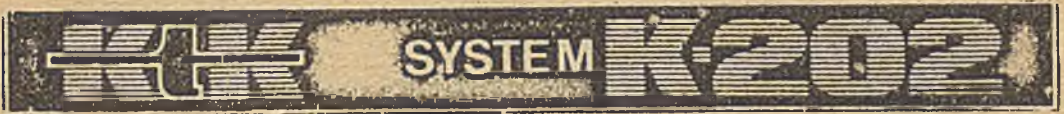
(Opis wyłącznie do celów szkoleniowych)

OPIS UŻYTKOWY EKSTRAKODÓW
DLA MASZYN CYFROWEJ K-202

OŚRODEK DOSKONALENIA KADR TECHNICZNYCH OW/NOT

Warszawa

1973



ZAKŁAD DOŚWIADCZALNY MINIKOMPUTERÓW

(Opis wyłącznie do celów szkoleniowych)

OPIS UŻYTKOWY EKSTRAKODÓW
DLA MASZINY CYFROWEJ K-202

OŚRODEK DOSKONALENIA KADR TECHNICZNYCH OW/NOT

Warszawa

1973

SPIS TREŚCI

	Str.
Rozdz. 1 -	
FORMA EKSTRAKODU, DZIAŁANIE EKSTRAKODU, SYGNALIZACJA BŁĘDÓW I POSTAĆ LICZB	1-1
Rozdz. 2 -	
EKSTRAKODY WEJŚCIA I WYJŚCIA	2-1
Rozdz. 3 -	
DZIAŁANIA NA LICZBACH CAŁKOWITYCH KRÓTKICH	3-1
Rozdz. 4 -	
DZIAŁANIA NA LICZBACH CAŁKOWITYCH DŁUGICH	4-1
Rozdz. 5 -	
DZIAŁANIA NA LICZBACH ZMIENNOPRZECINKOWYCH	5-1
Rozdz. 6 -	
FUNKCJE	6-1
Rozdz. 7 -	
ZMIANA LICZB	7-1
Rozdz. 8 -	
KOMUNIKACJA Z SYSTEMEM	8-1
Rozdz. 9 -	
PRZYKŁADY EKSTRAKODÓW	9-1

W S T Ę P

Oprócz podstawowej listy rozkazów istnieje grupa ekstrakodów, to jest makrorozkazów, które realizują: przesłania między urządzeniami zewnętrznymi, pamięciami zewnętrznymi, działania na liczbach, obliczenie podstawowych funkcji elementarnych itp.

Ekstrakody są makrorozkazami systemowymi, mieszczą się w pamięci niedostępnej dla programisty /blok zerowy/.

Rozdział 1

FORMA EKSTRAKODU, DZIAŁANIE EKSTRAKODU, SYGNALIZACJA BŁĘDÓW I POSTAĆ LICZB

Wybrane określenia

- | | |
|--|--|
| 1/ R \emptyset , R1,.....R7 | - numer rejestru |
| 2/ S/...../ | - zawartość pamięci |
| 3/ ZNAK WCZYTYWANY | - kombinacja bitów otrzymana z urzę
dzenia /7 lub 8 bitów/. W wypadku
wczytywania z bufora jest to kom-
binacja 8 bitów |
| 4/ ZNAK PISANY | - kombinacja 8 bitów |
| 5/ TEKST | - ciąg znaków alfanumerycznych |
| 6/ k | - maksymalny symboliczny numer urzę
dzenia dla danego systemu opera -
cyjnego. |
| 7/ <A> | - forma zapisu argumentu drugiego |
| 8/ A | - efektywny argument drugi |
| 9/ NAZWA | - nazwa ekstrakodu |
| 10/ Przyjęto następujące oznaczenia kodów: | |
| CR | - karetką |
| LF | - nowa linia |
| SP | - spacja |
| RUB OUT | - kasowanie |
| BLANK | - puste |

FORMA EKSTRAKODU

Ekstrakod ma formę rozkazu jednoargumentowego. Rozkaz ten może być uzupełniony parametrami ekstrakodu występującymi w kolejnych słowach za nazwą ekstrakodu.

Ilość, kolejność i znaczenie parametrów będzie podana dla każdego ekstrakodu.

SPOSÓB ZAPISU ROZKAZU "EKSTRAKOD"

EX <A> NAZWA. PARAMETR I.....PARAMETR N.

DZIAŁANIE EKSTRAKODU

Rozkaz EX umieszcza efektywny argument drugi /A/ w rejestrze 4 i wywołuje podprogram systemowy określony przez NAZWĘ. Powrót z ekstrakodu następuje do pierwszego rozkazu za ostatnim parametrem /wyjątki są zaznaczone w opisie/, przy czym zachowane są wszystkie rejestry z wyjątkiem 4 i wynikowych /jeżeli rejestr 4 nie jest wynikowym, wtedy jest w nim A/.

SYGNALIZACJA BŁĘDÓW

Sygnalizacja błędów następuje w wyniku przekroczenia zakresu liczby podczas: działań arytmetycznych, obliczaniu funkcji, zamiany liczb i dzielenia przez zero.

Są cztery sposoby reakcji systemu:

- sygnalizacja bitem V w rejestrze \emptyset i kontynuacja programu,
- zatrzymanie programu i wydruk informacyjny dla użytkownika
- różne adresy powrotu do programu w przypadku wykrycia błędu i poprawnego wykonania działania,
- wykonanie programu obsługi błędu wskazanego ekstrakodem
OBSŁUGA NADMIARÓW.

Szczegółowe informacje dotyczące sygnalizacji są podane w opisie.

POSTAĆ LICZB

Ekstrakody działają na liczbach 3 typów.

1. Liczba całkowita KRÓTKA.

Liczba ma długość 16 bitów, zajmuje 1 rejestr lub 1 słowo pamięci.

Bit \emptyset - określa znak liczby,

Bity /1-15/ - bity cyfrowe.

Zakres liczby: - 32768, + 32767

2. Liczba całkowita DŁUGA.

Liczba ma długość 32 bity, zajmuje rejestry 1 i 2 lub 2 słowa pamięci.

Bit \emptyset w pierwszym słowie - określa znak liczby.

Następne 31 bitów - bity cyfrowe.

Zakres liczby: - 2147483648, + 2147483647.

3. Liczba ZMIENNOPRZECINKOWA.

Liczba ma długość 48 bitów, zajmuje rejestry 1, 2 i 3 lub 3 słowa pamięci.

W pierwszym słowie jest zapisana cecha.

Bit \emptyset określa znak cechy.

Pozostałe 15 bitów są bitami cyfrowymi cechy.

W drugim i trzecim słowie jest zapisana mantysa.

Bit \emptyset drugiego słowa określa znak mantysy.

Pozostałe 31 bitów są bitami cyfrowymi mantysy.

Mantysa liczby zmiennoprzecinkowej posiada ok. 1 \emptyset

cyfr dziesiętnych, cecha ma zakres - $2^{\uparrow}32768$,

+ $2^{\uparrow}32767$.

Zero zmiennoprzecinkowe jest reprezentowane przez cechę i mantysę zerową.

ARGUMENT <A>. SPOSOBY ZAPISU I INTERPRETACJA.

Argument <A> rozkazu EX jest podawany i interpretowany w taki sam sposób, jak dla rozkazów posiadających tylko drugi argument /patrz opis ASSK-3/. Wyjątek stanowią ekstrakody działające na liczbach długich oraz liczbach zmiennoprzecinkowych. Wówczas argument drugi efektywny jest interpretowany jako adres wielkości, na której wykonane zostanie działanie. Może on posiadać następującą formę:

I. <A>, nr rejestru.

1/ Argumentem efektywnym działania ekstrakodu jest zawartość podanego rejestru.

Przykład:

<A>=, 5. R5 = 500

Ekstrakod działa na liczbie 500.

2/ W wypadku działania na liczbach długich i liczbach zmiennoprzecinkowych argumentem efektywnym jest wielkość zawierająca się w dwu lub trzech kolejnych komórkach pamięci, począwszy od miejsca wskazanego przez zawartość rejestru.

Przykład:

<A>=, 5. R5 = 500

Ekstrakod działa na wielkości z komórek:

- 500, 501 dla liczb długich

- 500, 501, 502 dla liczb zmiennoprzecinkowych

II. <A>=, nr rejestru.

1/ Argumentem efektywnym działania ekstrakodu jest zawartość komórki pamięci, której adres jest w rejestrze.

Przykład:

<A>=, 5. R5 = 400 S/400/ = 700

Ekstrakod działa na liczbie 700.

2/ Dla działań na liczbach długich i liczbach zmiennoprzecinkowych argument efektywny zostanie pobrany z pamięci począwszy od miejsca, na które wskazuje zawartość komórki pamięci o adresie podanym w rejestrze.

Przykład:

$$\langle A \rangle =, 5'. \quad R5 = 400 \quad S/400/ = 700$$

Ekstrakod działa na wielkości z komórek:

700, 701 - dla liczb długich

700, 701, 702 - dla liczb zmiennoprzecinkowych.

III. $\langle A \rangle =$, nr rejestru & nr rejestru modyfikującego.

1/ Argumentem efektywnym działania ekstrakodu jest suma za - wartości obydwu rejestrów.

Przykłady:

$$a/ \langle A \rangle =, 5 \& 5. \quad R5' = 200$$

Ekstrakod działa na liczbie 400.

$$b/ \langle A \rangle =, 5 \& 7. \quad R5 = 100 \quad R7 = 3500 /$$

Ekstrakod działa na liczbie 3600.

2/ Dla działań na liczbach długich i liczbach zmiennoprzecinkowych efektywny argument drugi zostanie pobrany z pamięci począwszy od miejsca, na które wskazuje suma zawartości obydwu rejestrów.

Przykłady:

$$a/ \langle A \rangle =, 5 \& 5. \quad R5 = 200$$

Ekstrakod działa na wielkości z komórek:

400, 401 - dla liczb długich

400, 401, 402 - dla liczb zmiennoprzecinkowych

$$b/ \langle A \rangle =, 5 \& 7. \quad R5 = 100 \quad R7 = 3500$$

Ekstrakod działa na wielkości z komórek:

3600, 3601 - dla liczb długich

3600, 3601, 3602 - dla liczb zmiennoprzecinkowych

IV. $\langle A \rangle =$, nr rejestru & nr rejestru modyfikującego

1/ Argumentem efektywnym działania ekstrakodu jest zawartość komórki pamięci, której adres wskazuje suma zawartości obydwu wyszczególnionych rejestrów.

Przykład:

$$a/ \langle A \rangle =, 5 \& 5'. \quad R5 = 100 \quad S/200/ = 3$$

Ekstrakod działa na liczbie 3.

$$b/ \langle A \rangle =, 1 \& 7'. \quad R1 = 3 \quad R7 = 378 \quad S/381/ = 55$$

Ekstrakod działa na liczbie 55.

- .2/ Dla działań na liczbach długich i liczbach zmiennoprzecinkowych efektywny argument drugi pobrany zostanie z pamięci począwszy od miejsca na które wskazuje zawartość komórki pamięci będący sumą zawartości obydwu rejestrów.

Przykłady:

a/ $\langle A \rangle = 5 \& 5'$. $R5 = 100$ $S/200/ = 300$

Ekstrakod działa na wielkości z komórek:

300, 301 - dla liczb długich

300, 301, 302 - dla liczb zmiennoprzecinkowych

b/ $\langle A \rangle = 1 \& 7'$. $R1 = 3$ $R7 = 378$ $S/381/ = 550$

Ekstrakod działa na wielkości z komórek:

550, 551 - dla liczb długich

550, 551, 552 - dla liczb zmiennoprzecinkowych.

V. $\langle A \rangle =$ (argument 2)

- 1/ Argumentem efektywnym działania ekstrakodu jest wielkość podana w nawiasie.

Przykłady:

a/ $\langle A \rangle = (500)$

Ekstrakod działa na liczbie 500.

b/ $\langle A \rangle = (ety + 100)$ $ety = 3000$

Ekstrakod działa na liczbie 3100

- 2/ Dla działań na liczbach długich i liczbach zmiennoprzecinkowych efektywny argument drugi zostanie pobrany z pamięci od adresu, na który wskazuje wielkość zawarta w nawiasie.

Przykłady:

a/ $\langle A \rangle = (500)$

Ekstrakod działa na wielkości z komórek:

500, 501 - dla liczb długich

500, 501, 502 - dla liczb zmiennoprzecinkowych

b/ $\langle A \rangle = (ety)$ $ety = 3000$

Ekstrakod działa na wielkości z komórek:

3000, 3001 - dla liczb długich

3000, 3001, 3002 - dla liczb zmiennoprzecinkowych.

VI. $\langle A \rangle =$ (argument 2')

- 1/ Argumentem efektywnym działania ekstrakodu jest zawartość komórki pamięci, której adres wskazuje argument 2.

Przykład:

$\langle A \rangle = (500')$ $8/500/ = 1000$

Ekstrakod działa na liczbie 1000.

- 2/ Dla działań na liczbach długich i zmiennoprzecinkowych argument efektywny pobrany zostanie z pamięci począwszy od miejsca, na które wskazuje zawartość komórki pamięci o adresie podanym w nawiasach okrągłych.

Przykład:

$$\langle A \rangle = (5000) \quad S/5000/ = 1000$$

Ekstrakod działa na wielkości z komórek:

1000, 1001 - dla liczb długich

1000, 1001, 1002 - dla liczb zmiennoprzecinkowych

VII. $\langle A \rangle = (\text{argument 2} \ \& \ \text{nr rejestru modyfikującego})$

- 1/ Argumentem efektywnym działania ekstrakodu jest wielkość będąca sumą argumentu drugiego i zawartości rejestru modyfikującego.

Przykłady:

a/ $\langle A \rangle = (355 \ \& \ 5) \quad R5 = 700$

Ekstrakod działa na liczbie 1055

b/ $\langle A \rangle = (\text{etyk} \ \& \ 5) \quad R5 = 300 \quad \text{etyk} = 10000$

Ekstrakod działa na liczbie 10300.

- 2/ Dla działań na liczbach długich i liczbach zmiennoprzecinkowych efektywny argument drugi pobrany zostanie z pamięci począwszy od miejsca, na które wskazuje suma zawartości rejestru i podanej w nawiasie wielkości.

Przykłady:

a/ $\langle A \rangle = (555 \ \& \ 5) \quad R5 = 500$

Ekstrakod działa na wielkości z komórek:

1055, 1056 - dla liczb długich

1055, 1056, 1057 - dla liczb zmiennoprzecinkowych

b/ $\langle A \rangle = (\text{etyk} \ \& \ 5) \quad R5 = 300 \quad \text{etyk} = 10000$

Ekstrakod działa na liczbie z komórek:

10300, 10301 - dla liczb długich

10300, 10301, 10302 - dla liczb zmiennoprzecinkowych.

VIII. $\langle A \rangle = (\text{argument 2} \ \& \ \text{nr rejestru modyfikującego})$

- 1/ Argumentem efektywnym działania ekstrakodu jest zawartość komórki pamięci, której adresem jest suma argumentu drugiego i zawartości rejestru modyfikującego.

Przykłady:

a/ <A>=(35 & 1') R1 = 13 S/48/ = 55

Ekstrakod działa na liczbie 55.

b/ <A>=(agata & 7') agata = 370 R7 = 35 S/405/ = 38

Ekstrakod działa na liczbie 38.

- 2/ Dla działań na liczbach zmiennoprzecinkowych i liczbach długich efektywny argument drugi pobrany zostanie z pamięci począwszy od miejsca, na które wskazuje zawartość komórki pamięci obliczonej jak w VIII.1/.

Przykłady:

a/ <A>=(35 & 1') R1 = 13 S/48/ = 550

Ekstrakod działa na wielkości z komórek:

550, 551 - dla liczb długich

550, 551, 552 - dla liczb zmiennoprzecinkowych

b/ <A>=(agata & 7') agata = 370 R7 = 35 S/405/ = 380

Ekstrakod działa na wielkości z komórek:

380, 381 - dla liczb długich

380, 381, 382 - dla liczb zmiennoprzecinkowych.

- IX. Efektywny argument drugi może być modyfikowany przy pomocy rozkazu mod wg zasad określonych w opisie ASSK-3.

Uwaga 1.

WARUNKI MOŻNA UMIESZCZAĆ TYLKO W EKSTRAKODACH NIE POSIADAJĄCYCH PARAMETRÓW

Przykłady:

a/ ex (we' ? G) redf.

b/ ex, 4 ? 1. csgf.

c/ ex, 4 ? 2. absi.

Uwaga 2.

PRZED EKSTRAKODAMI Z PARAMETRAMI NIE MOŻNA UMIESZCZAĆ ROZKAZÓW Z PRZESKOKIEM tzn. rozkazów

A D O S, A D O T, C L B O, C L M O.

Rozdział 2

EKSTRAKODY WEJŚCIA I WYJŚCIA

		str.
REC	- czytaj znak	2-2
PRC	- pisz znak	2-3
RET	- czytaj tekst	2-4
PRT	- pisz tekst	2-5
RL	- czytaj linię	2-6
WL	- pisz linię	2-7
PREC.	- czytaj record	2-8
DROW	- rysuj ploter	2-9
REOW	- czytaj liczbę oktalną krótką	2-10
PROW	- pisz liczbę krótką oktalnie	2-11
REDW	- czytaj liczbę dziesiętną krótką	2-12
PRDW	- pisz liczbę krótką dziesiętnie	2-13
REDI	- czytaj liczbę długą dziesiętną	2-14
PRDI	- pisz liczbę długą dziesiętnie	2-15
REDF	- czytaj liczbę zmiennoprzecinkową	2-16
PRDF	- pisz liczbę zmiennoprzecinkową	2-17

CZYTAJ ZNAK

FORMA ROZKAZU:

EX <A> REC.PARAMETR.

- A = \emptysetk - jest symbolicznym numerem urządzenia wejścia,
- A \geq 256 - jest adresem bufora w pamięci operacyjnej; bufor zawiera znak na młodszej połowie słowa,
- PARAMETR = 1...7 - jest numerem rejestru, do którego ma zostać wczytany znak; po wykonaniu ekstrakodu młodsza połowa rejestru o numerze równym parametrowi, będzie zawierała wczytany znak, a w starszej będzie zero.

Jeśli PARAMETR \neq 4 i A \geq 256 wtedy w rejestrze jest adres następnego znaku.

PISZ ZNAK

FORMA ROZKAZU:

EX <A> PRC.PARAMETR.

- A = \emptyset k - jest symbolicznym numerem urządzenia wyjścia,
- A \geq 256 - jest adresem bufora w pamięci operacyjnej, w którym zostanie umieszczony znak na młodszej części słowa.

PARAMETR = 1 3 1

- 5.....7 - jest numerem rejestru, z którego będzie wyprowadzony znak do urządzenia o numerze =<A> /pod adres = <A> /.

Jeśli A \geq 256 wtedy REJESTR 4 po wykonaniu rozkazu zawiera adres następnego słowa.

CZYTAJ TEKST

FORMA ROZKAZU:

EX < A > RET.PARAMETR 1. PARAMETR 2.

- A = 0.....k - jest symbolicznym numerem urządzenia wejścia, z którego będą wprowadzane znaki,
- A ≥ 256 - jest adresem bufora w pamięci operacyjnej, w którym młodsze połowy kolejnych słów będą traktowane jak znaki,
- PARAMETR 1 - jest adresem początkowym bufora, w którym będzie umieszczony tekst po dwa znaki w słowie /na starszej połowie pierwszy, na młodszej drugi/, uzupełniony jednym lub dwoma znakami o kodzie 128 do parzystej ilości znaków w tekście /ilość znaków w tekście nie jest ograniczona/; znak BLANK i RUB OUT są pomijane.
- PARAMETR 2 - jest znakiem o kodzie z przedziału /1-127/ zapisanym w postaci np. !!z. lub innej, który ma być traktowany jako znak końca tekstu; znak ten nie będzie wczytany do bufora
- REJESTR 4 - po wykonaniu rozkazu będzie zawierał adres końca tekstu + 1.

Należy pamiętać, że tekst musi być zakończony znakiem końca tekstu zdefiniowanym przez PARAMETR 2.

PISZ TEKST

FORMA ROZKAZU:

EX < A > PRT.PARAMETR.

A = 0...k

- jest symbolicznym numerem urządzenia wyjścia, na którym będzie pisany tekst,

A ≥ 256

- jest adresem bufora, do którego będzie wpisywany tekst po jednym znaku w słowo,

PARAMETR

- jest adresem początkowym tekstu w pamięci operacyjnej; tekst umieszczony jest po dwa znaki w słowie, a znakiem kończącym tekst jest znak o kodzie 128.

REJESTR 4 /gdy A ≥ 256/

- zawiera adres pierwszego wolnego słowa w buforze.

TA:

!ab. !ed. !ef. 100200

ad, 6, 6 & 6.
 ed, 6 (TA)
 st, 6 (TAB)
 ex(3) prt-TAB

CZYTAJ LINIE

FORMA ROZKAZU:

EX <A> RL.PARAMETR.

- A = \emptysetk - jest symbolicznym numerem urzędze -
 nia, z którego będą czytane znaki,
 PARAMETR - wskazuje początek bufora, do które-
 go będą wpisywane znaki po jednym w
 słowo /do młodszej połowy słowa,
 starsza połowá = \emptyset /, ostatnim zna -
 kiem wpisanym jest "LF" /kod 1 \emptyset /.
- ZNAKI WPROWADZONE - z przedziału 32....95 oraz "CR"
 jeśli nie poprzedzają i nie nastę -
 pują po "LF",
- ZNAKI IGNOROWANE - pozostałe,
- MAX ILOŚĆ ZNAKÓW = 12 \emptyset - znak 121 jest generowany i ma kod
 "LF" /1 \emptyset /.
- REJESTR 4 - zawiera adres pierwszego wolnego
 słowa w buforze.

PISZ LINIE

FORMA ROZKAZU:

EX < A > WL.PARAMETR.

A = 0....k

- jest symbolicznym numerem urzą-
dzenia wyjścia,

PARAMETR

- wskazuje początek bufora, w któ-
rym są wpisane znaki po jednym w
słowo /w młodszej połowie słowa
znak, starsza połowa dowolna/,
ostatnim znakiem wyprowadzonym
jest "LF" /kod 10/; przed nim
jest generowany "CR" /kod 13/.

REJESTR 4

- zawiera adres pierwszego nie wy-
prowadzonego znaku.

CZYTAJ REKORD

FORMA ROZKAZU:

EX < A > RREC. PARAMETR 1. PARAMETR 2.

A = Ø.....k

- jest symbolicznym numerem urządzenia, z którego będzie czytany rekord.

PARAMETR 1

- jest adresem początkowym bufora w którym będzie umieszczany rekord po jednym znaku w słowie /na młodszej połowie słowa, starsza połowa = Ø/.

PARAMETR 2

- określa ilość znaków wczytywanych z urządzenia.

REJESTR 4

- zawiera adres końca wczytanego rekordu + 1.

RYSUJ NA PLOTER

FORMA ROZKAZU:

EX <A> DROW.

<A> = 0.....k

- jest symbolicznym numerem urządzenia wyjścia, do którego jest podłączony ploter.

Parametry odcinka rysowanego na ploter muszą być podane w R1, R2 i R3.

gdzie:

R1 - przyrost dx - liczba całkowita dodatnia lub ujemna

R2 - przyrost dy - liczba całkowita dodatnia lub ujemna

R3 - skala

przy czym skala < 31

jeśli skala jest liczbą dodatnią - odcinek jest rysowany /pióro opuszczone/

jeśli skala jest liczbą ujemną - pusty przebieg /pióro podniesione/.

U w a g a: w przypadku gdy dx, dy > 63

ekstrakod zaokrągla do +,-63,

jeśli skala > 31 ekstrakod zaokrągla do +, -31.

CZYTAJ LICZBĘ OKTALNĄ KRÓTKĄ

FORMA ROZKAZU:

EX <A> REOW. PARAMETR.

A = \emptysetk

- jest symbolicznym numerem urządzenia wejścia,

A \geq 256

- jest adresem bufora w pamięci operacyjnej, w którym liczba jest zapisana po jednym znaku alfanumerycznym w słowie /w młodszej połowie słowa/,

PARAMETR = 1...7

- jest numerem rejestru, do którego będzie wprowadzona liczba /znak kończący liczbę jest w komórce CHAR/,

REJESTR 4

/gdy A \geq 256/

- zawiera adres znaku kończącego liczbę +1,

Znaki ignorowane

- BLANK, RUB OUT,

- wszystkie znaki przed pierwszą cyfrą,

- SP w liczbie, jeśli są pojedyncze.

Znaki końca liczby:

- każdy znak kończy liczbę z wyjątkiem cyfry i pojedynczej SP.

Przekroczenie zakresu liczby: sygnalizowane bitem V w rejestrze \emptyset .

PISZ LICZBĘ KRÓTKĄ OKTALNIE

FORMA ROZKAZU:

EX <A> PROW. PARAMETR.

A = \emptysetk

- jest symbolicznym numerem urządzenia wyjścia,

A \geq 256

- jest adresem początkowym bufora pamięci, do którego ma być wpisana liczba po jednym znaku alfanumerycznym w słowo,

PARAMETR = 1...3 i 5...7

- jest numerem rejestru, z którego jest pobierana liczba do wydrukowania.

REJESTR 4 /gdy A \geq 256/

- zawiera adres pierwszego wolnego słowa w buforze.

Postać wydruku liczby:

liczba jest wyprowadzana w postaci oktalnej, cyfry są z zakresu $[\emptyset, 7]$, przy czym cyfrą pierwszą jest zawsze \emptyset . Obliczając szerokość pola dla liczb wyprowadzanych należy uwzględnić:

- spację początkową,
- siedem miejsc na liczbę.

CZYTAJ LICZBĘ DZIESIETNĄ KRÓTKĄ

FORMA ROZKAZU:

EX <A> REDW. PARAMETR.

A = 0...k

- jest symbolicznym numerem urządzenia wejścia,

A \geq 256

- jest adresem początkowym bufora w pamięci operacyjnej, w którym liczba jest zapisana po jednym znaku alfanumerycznym w słowie /w młodszej połowie słowa/,

PARAMETR = 1...7

- jest numerem rejestru, do którego będzie wprowadzona liczba /znak kończący liczbę jest w komórce CHAR/,

REJESTR 4 /gdy A \geq 256/

- zawiera adres znaku kończącego liczbę + 1.

Znaki ignorowane:

- BLANK, RUB OUT,

- wszystkie znaki przed znakami "+", "-", lub pierwszą cyfrą.

- SP w liczbie, jeśli są pojedyncze.

Znak końca liczby:

- każdy znak kończy liczbę z wyjątkiem cyfry i pojedynczej SP.

Przekroczenie zakresu liczby: sygnalizowane bitem V w rejestrze 0.

PISZ LICZBE KRÓTKA DZIESIĘTNIIE

FORMA ROZKAZU:

EX <A> PRDW.PARAMETR 1. PARAMETR 2.

A = 0.....k

- jest symbolicznym numerem urzędze -
nia wyjścia,

A ≥ 256

- jest adresem początkowym bufora w
pamięci, do którego ma być wpisana
liczba po jednym znaku alfanumerycz
nym w słowo,PARAMETR 1 = 1...3 i 5...7 - jest numerem rejestru, z którego
jest pobrana liczba do wydrukowania,

PARAMETR 2 - określa postać wydruku liczby.

REJESTR 4 /gdy A ≥ 256/ - zawiera adres pierwszego wolnego
słowa w buforze.Postać wydruku liczby: liczba jest wyprowadzana w postaci
liczby dziesiętnej całkowitej, przy
czym PARAMETR 2 oznacza:

bit 0 = 1

nie drukuj SP przed liczbą,

bit 1 = 1

nie rezerwuj miejsca na znak,

bit 1 = 0 i bit 2 = 1

drukowanie "+" zamiast SP,

bit 3 = 1

drukowanie zer nieznaczących,

bit 4

ignorowany,

bit 5 = 1

wydruk przecinka za liczbą

bity 6-10

ignorowane,

bity 11 - 15

ilość znaków przeznaczona na zapisa
nie liczby; jeśli wartość liczby bę
dzie większa od maksymalnej warto -
ści oczekiwanej wtedy ilość wydruko
wanych znaków będzie odpowiednio
większa

CZYTAJ LICZBE DŁUGA DZIESIĘTNA

FORMA ROZKAZU:

EX < A > REDI.

A = \emptyset ...k

- jest symbolicznym numerem urządzenia wejścia,

A \geq 256

- jest adresem początkowym bufora w pamięci operacyjnej, w którym liczba jest zapisana po jednym znaku alfanumerycznym w słowie /w młodszej połowie słowa/,

Liczba jest wprowadzona do rejestrów R1 i R2.

REJESTR 4 /gdy A \geq 256/

- zawiera adres znaku kończącego liczbę + 1.

Znaki ignorowane:

- BLANK, RUB OUT,
- wszystkie znaki przed "+", "-" lub pierwszą cyfrą,
- SP w liczbie jeśli są pojedyncze.

Znak końca liczby:

- każdy znak kończy liczbę z wyjątkiem cyfry i pojedynczej SP.

Przekroczenie zakresu:

- sygnalizowane bitem V w rejestrze \emptyset .

PISZ LICZBĘ DŁUGA DZIESIĘTNI

FORMA ROZKAZU:

EX < A > PRDI.PARAMETR.

- A = \emptysetk - jest symbolicznym numerem urządzenia wyjścia,
- A \geq 256 - jest adresem początkowym bufora w pamięci, od którego ma być wpisana liczba po jednym znaku alfanumerycznym w słowo,
- PARAMETR - określa postać wydruku liczby.
- REJESTR 4 /gdy A \geq 256/ - zawiera adres pierwszego wolnego słowa w buforze.

Liczba do wydrukowania /wpisania w postaci alfanumerycznej do pamięci/ jest pobierana z rejestrów R1 i R2.

Postać wydruku liczby: liczba jest wyprowadzona w postaci liczby dziesiętnej całkowitej, przy czym PARAMETR oznacza:

- bit \emptyset = 1 nie drukuj SP przed liczbą,
- bit 1 = 1 nie rezerwuj miejsca na znak,
- bit 1 = \emptyset i bit 2 = 1 drukowanie "+" zamiast SP,
- bit 3 = 1 drukowanie zer nieznaczących,
- bit 4 ignorowany,
- bit 5 = 1 wydruk przecinka za liczbą,
- bity 6-1 \emptyset ignorowane,
- bity 11-15 ilość znaków przeznaczona na zapisanie liczby jeśli wartość liczby będzie większa od maksymalnej wartości oczekiwanej, wtedy ilość wydrukowanych znaków będzie odpowiednio większa.

CZYTAJ LICZBE ZMIENNOPRZECINKOWA

FORMA ROZKAZU:

EX <A> REDF.

A = 0...k

- jest symbolicznym numerem urządzenia wejścia,

A \geq 256

- jest adresem początkowym bufora w pamięci operacyjnej, w którym liczba jest zapisana po jednym znaku alfanumerycznym w słowie /w młodszej połowie słowa/,

Liczba jest wprowadzana do rejestrów R1, R2 i R3 w postaci zmiennoprzecinkowej, znormalizowanej.

REJESTR 4 /gdy A \geq 256/

- zawiera adres znaku kończącego liczbę + 1.

Znaki ignorowane:

- blank, rub out,

- wszystkie znaki przed "+", "-", ".", "E" lub pierwszą cyfrą,

- SP w liczbie jeśli są pojedyncze.

Znaki końca liczby:

- "." i "E" jeśli nie są częścią wczytywanej liczby oraz pozostałe znaki z wyjątkiem: cyfry, pojedynczej SP w liczbie.

Przekroczenie zakresu liczby:

1/ adres programu obsługi określa ekstrakod OBSŁUGA NADMIARÓW OV,

2/ jeśli ekstrakod OBSŁUGA NADMIARÓW OV nie był wcześniej wykonany, zadanie jest przerwane, a wydruk na monitorze określa przyczynę i adres ekstrakodu, w którym powstał nadmiar.

PISZ LICZBĘ ZMIENNOPRZECINKOWA

FORMA ROZKAZU:

EX < A > PRDF.PARAMETR.

A = 0.....k

- jest symbolicznym numerem urządzenia wyjścia,

A \geq 256

- jest adresem początkowym bufora w pamięci, do którego ma być wpisana liczba po jednym znaku alfanumerycznym w słowo,

PARAMETR

- określa postać wydruku liczby.

REJESTR 4 /gdy A \geq 256/

- zawiera adres pierwszego wolnego słowa w buforze.

Liczba do wydrukowania /wpisana w postaci alfanumerycznej do pamięci/ jest pobierana z rejestrów 1, 2 i 3.

Postać wydruku liczby: liczba jest wyprowadzana w postaci liczby

dziesiętnej z kropką pozycyjną lub liczby z wydzieloną potęgą 10 o wykładniku całkowitym dwucyfrowym ze znakiem /zamiast podstawy drukowane "E"/ przy czym PARAMETR oznacza:

bit 0 = 1

nie drukuj SP przed liczbą,

bit 1=1

nie rezerwuj miejsca na znak,

bit 1 = 0 i bit 2 = 1

drukowanie "+" zamiast SP,

bit 3 = 1 i bit 4 = 1

drukowanie zer nieznaczących,

bit 4 = 1

drukuj liczbę bez wykładnika,

bit 5 = 1

wydruk przecinka za liczbą,

bity 6-10

ilość cyfr przed kropką,

bity 11-15

ilość cyfr po kropce.

Rozdział 3

DZIAŁANIA NA LICZBACH CAŁKOWITYCH KRÓTKICH

	Str.
MPLW - mnożenie liczb całkowitych krótkich	3-2
DIVW - dzielenie całkowite liczb krótkich	3-3

MNOŻENIE LICZB CAŁKOWITYCH KRÓTKICH

FORMA ROZKAZU:

EX < A > MPLW.PARAMETR.

- MNOŻNA - jest umieszczona w rejestrze, którego numer określa PARAMETR = 1..3 lub 5...7,
- MNOŻNIK - jest A rozkazu,
- ILOCZYN - jest umieszczony jako liczba krótka w rejestrze, w którym była mnożna,
- NADMIAR - jest sygnalizowany w rejestrze przez ustawienie bitu V.

DZIELENIE CAŁKOWITE LICZB KRÓTKICH

FORMA ROZKAZU:

EX < A > DIVW.PARAMETR.

- DZIELNA - jest liczbą krótką umieszczoną w rejestrze, którego numer określa PARAMETR = 1...3 lub 5...7,
- DZIELNIK - jest A rozkazu,
- ILORAZ - zostanie umieszczony w rejestrze w którym była dzielna,
- RESZTA - zostanie umieszczona w rejestrze 4, $\text{ILORAZ} = \text{DZIELNA} / \text{DZIELNIK}$
 $\text{RESZTA} = \text{DZIELNA} - \text{ILORAZ} \times \text{DZIELNIK}$
 przy czym:
 $|\text{RESZTA}| < |\text{DZIELNIK}|$
 i $\text{SGN}(\text{RESZTA}) \times \text{SGN}(\text{DZIELNA}) \neq -1$
- NADMIAR - ustawia bit V w rejestrze 0;
 działanie nie jest wykonywane,
- DZIELNIK = 0 - patrz: OBSŁUGA NADMIARÓW OV na stronie 5-7,

Rozdział 4

DZIAŁANIA NA LICZBACH CAŁKOWITYCH DŁUGICH

	Str.
ADDI - dodawanie liczb całkowitych długich	4-2
SUBI - odejmowanie liczb całkowitych długich	4-3
MPLI - mnożenie liczb całkowitych długich	4-4
DIVI - dzielenie całkowite liczb długich	4-5
ABSI - wartość bezwzględna liczby całkowitej długiej	4-6
COMI - porównanie liczb całkowitych długich	4-7

DODAWANIE LICZB CAŁKOWITYCH DŁUGICH

FORMA ROZKAZU:

EX < A > ADDI.

- | | |
|------------|---|
| SKŁADNIK 1 | - jest umieszczony w rejestrach 1, 2, |
| SKŁADNIK 2 | - jest umieszczony od adresu = A w pamięci w dwóch kolejnych słowach, |
| SUMA | - zostanie umieszczona w rejestrach 1, 2, |
| NADMIAR | - będzie sygnalizowany w rejestrze \emptyset przez ustawienie bitu V. |

ODEJMOWANIE LICZB CAŁKOWITYCH DŁUGICH

FORMA ROZKAZU:

EX < A > SUBI.

- ODJEMNA - jest umieszczona w rejestrach
1, 2,
- ODJEMNIK - jest umieszczony od adresu = A
w dwóch kolejnych słowach,
- RÓŻNICA - zostanie umieszczona w rejest-
rach 1, 2,
- NADMIAR - będzie sygnalizowany w rejest -
rze \emptyset przez ustawienie bitu V.

MNOŻENIE LICZB CAŁKOWITYCH DŁUGICH

FORMA ROZKAZU:

EX <A> MPLI.

- MNOŻNA - jest umieszczona w rejestrach 1, 2,
- MNOŻNIK - jest umieszczony od adresu = A w dwóch kolejnych słowach pamięci
- ILOCZYN - jako liczba długa zostanie umieszczony w rejestrach 1, 2,
- NADMIAR - będzie sygnalizowany w rejestrze \emptyset przez ustawienie bitu V.

DZIELENIE CAŁKOWITE LICZB DŁUGICH

FORMA ROZKAZU:

EX <A> DIVI.

- DZIELNA - jest umieszczona w rejestrach 1, 2,
- DZIELNIK - jest umieszczony od adresu = A w dwóch kolejnych słowach pamięci
- IŁORAZ - jest umieszczony w rejestrach 1, 2.
- RESZTA - nie jest podawana,
- NADMIAR - będzie sygnalizowany w rejestrze \emptyset przez ustawienie bitu V; działanie nie jest wykonywane,
- DZIELNIK = \emptyset - patrz: OBSŁUGA NADMIARÓW OV na stronie 5-7.

WARTOŚĆ BEZWZGLEDNA LICZBY CAŁKOWITEJ DŁUGIEJ

FORMA ROZKAZU:

EX <A> ABSI.

ROZKAZ jest JEDNOARGUMENTOWY - z argumentem umieszczonym w rejestrach 1, 2,

WYNIK jest umieszczony w rejestrach 1, 2.

NADMIAR będzie sygnalizowany w rejestrze \emptyset przez ustawienie bitu V.

Uwaga: Postać części <A> rozkazu nie ma wpływu na wykonanie ekstrakodu. Po wyjściu z ekstrakodu w R4 znajduje się efektywny argument drugi A. W przypadku, gdy $A = 4$, rejestr R4 nie ulega zmianie w trakcie wykonywania ekstrakodu.

PORÓWNANIE LICZB CAŁKOWITYCH DŁUGICH

FORMA ROZKAZU:

EX <A> COMI.

LICZBA 1

- jest umieszczona w rejestrach
1, 2,

LICZBA 2

- jest umieszczona od adresu = A
w dwóch kolejnych słowach pamięci

PORÓWNANIE

- ustawia wskaźniki L, E, G, w rejestrze \emptyset :LICZBA 1 < LICZBA 2 L = 1 i E = G = \emptyset LICZBA 1 = LICZBA 2 E = 1 i L = G = \emptyset LICZBA 1 > LICZBA 2 G = 1 i L = E = \emptyset

Rozdział 5

DZIAŁANIA NA LICZBACH ZMIENNOPRZECINKOWYCH

	Str.
Uwagi	5-2
ADDF - dodawanie liczb zmiennoprzecinkowych	5-3
SUBF - odejmowanie liczb zmiennoprzecinkowych	5-4
MPLF - mnożenie liczb zmiennoprzecinkowych	5-5
DIVF - dzielenie liczb zmiennoprzecinkowych	5-6
OV - obsługa nadmiarów	5-7
COMF - porównaj liczbę zmiennoprzecinkową	5-8
CSGF - zmień znak liczby zmiennoprzecinkowej	5-9

UWAGI:

1. Ekstrakody działające na liczbach zmiennoprzecinkowych wymagają, aby argument pierwszy był umieszczony w rejestrach 1,2,3 w ten sposób, aby:

- w rejestrze 1 była cecha,
- w rejestrach 2,3 była mantysa.

Argument drugi brany jest z trzech kolejnych komórek pamięci wskazanych przez efektywny argument drugi.

Wynik działania jest znormalizowany i umieszczony w rejestrach 1,2,3. Liczby, na których wykonuje się działanie powinny być znormalizowane. W przeciwnym razie można uzyskać błędny wynik bez sygnalizacji.

2. Dla ekstrakodów jednoargumentowych, z argumentem umieszczonym w rejestrach 1,2,3, na przykład funkcje, argument efektywny rozkazu jest po powrocie z ekstrakodu umieszczony w rejestrze 4. Jeśli, więc, dla tych ekstrakodów rozkaz będzie napisany w formie EX,4.NAZWA., wtedy zawartość rejestru 4 nie ulegnie zmianie.

3. Dla działań arytmetycznych na liczbach zmiennoprzecinkowych /ADDF, SUBF, MPLF, DIVF/ są możliwe dwa sposoby odwoływania się do ekstrakodów:

a/ EX <A> NAZWA.

b/ NAZWA <A>

Oba sposoby odwołania się są równoważne.

Przykłady:

EX(228)ADDF.	lub	ADDF(228)
EX,5 & 2'. SUBF.	lub	SUBF, 5 & 2'.
MOD, 3'. EX(29 & 5'?L5)DIVF.	lub	DIVF(29 & 5'?L5)

DODAWANIE LICZB ZMIENNOPRZECINKOWYCH

FORMY ROZKAZU:

a/ EX <A> ADDF.

b/ ADDF <A>

SKŁADNIK 1

- jest umieszczony w rejestrach
1,2,3,

SKŁADNIK 2

- jest umieszczony od adresu = A
w trzech kolejnych słowach pa-
mięci,

SUMA

- jest umieszczona w rejestrach
1,2,3,

NADMIAR

- 1/ adres programu obsługi błę-
du określa ekstrakod OBSŁUGA
NADMIARÓW OV,
2/ jeśli ekstrakod OBSŁUGA NAD-
MIARÓW OV nie był wcześniej
wykonany, zadanie jest przer-
wane, a wydruk na monitorze
określa przyczynę i adres
ekstrakodu, w którym powstał
nadmiar.

ODEJMOWANIE LICZB ZMIENNOPRZECINKOWYCH

FORMY ROZKAZU:

a/ EX <A> SUBF.

b/ SUBF <A>

- ODJEMNA - jest umieszczona w rejestrach 1,2,3,
- ODJEMNIK - jest umieszczony od adresu = A 2 w trzech kolejnych słowach pamięci,
- RÓŻNICA - jest umieszczona w rejestrach 1,2,3,
- NADMIAR - 1/ adres programu obsługi błędu określa ekstrakod OBSŁUGA NADMIARÓW OV,
2/ jeśli ekstrakod OBSŁUGA NADMIARÓW OV nie był wcześniej wykonany, zadanie jest przerwane, a wydruk na monitorze określa przyczynę i adres ekstrakodu, w którym powstał nadmiar.

MNOŻENIE LICZB ZMIENNOPRZECINKOWYCH

FORMA ROZKAZU:

a/ EX <A> MPLF.

b/ MPLF <A>

- MNOŻNA - jest umieszczona w rejestrach 1,2,3,
- MNOŻNIK - jest umieszczony od adresu = A 2 w trzech kolejnych słowach pamięci,
- ILOCZYN - jest umieszczony w rejestrach 1,2,3.
- NADMIAR - 1/ adres programu obsługi błędu określa ekstrakod OBSŁUGA NADMIARÓW OV,
2/ jeśli ekstrakod OBSŁUGA NADMIARÓW OV nie był wcześniej wykonany, zadanie jest przerwane, a wydruk na monitorze określa przyczynę i adres ekstrakodu, w którym powstał nadmiar.

DZIELENIE LICZB ZMIENNOPRZECINKOWYCH

FORMY ROZKAZU:

a/ EX <A> DIVF.

b/ DIVF <A>

- DZIELNA - jest umieszczona w rejestrach 1,2,3,
- DZIELNIK - jest umieszczony od adresu = A w trzech kolejnych słowach pamięci,
- ILORAZ - jest umieszczony w rejestrach 1,2,3,
- DZIELNIK = 0 - 1/ adres programu obsługi błędu określa ekstrakod OBSŁUGA NADMIARÓW OV,
2/ jeśli ekstrakod OBSŁUGA NADMIARÓW OV nie był wcześniej wykonany, zadanie jest przerwane, a wydruk na monitorze określa przyczynę i adres ekstrakodu, w którym powstał błąd.
- NADMIAR - 1/ adres programu obsługi błędu określa ekstrakod OBSŁUGA NADMIARÓW OV,
2/ jeśli ekstrakod OBSŁUGA NADMIARÓW OV nie był wcześniej wykonany, zadanie jest przerwane, a wydruk na monitorze określa przyczynę i adres ekstrakodu, w którym powstał nadmiar.

OBSŁUGA NADMIARÓW

FORMY ROZKAZU:

EX <A> OV.

W przypadku wykrycia:

- nadmiaru w działaniach zmiennoprzecinkowych,
- dzielenia liczby zmiennoprzecinkowej przez zero,
- dzielenia liczby całkowitej przez zero,

działanie nie jest wykonywane,

śląd ekstrakodu, w którym wystąpił błąd jest umieszczony pod adresem = A, a program jest kontynuowany od adresu,

- A + 1, gdy wystąpił nadmiar w działaniu zmiennoprzecinkowym,
- A + 2, gdy wykryta została chęć dzielenia liczby zmiennoprzecinkowej przez zero,
- A + 3, gdy wystąpiło dzielenie liczby całkowitej przez zero.

W przypadku, gdy własna obsługa nadmiarów nie została zadeklarowana ekstrakodem OV przed wystąpieniem jednego z błędów określonych wyżej, następuje przerwanie zadania, a wydruk na monitorze wskazuje przyczynę i miejsce wystąpienia błędu.

W przypadku, gdy efektywny argument drugi jest równy \emptyset maszyna powraca do standardowej obsługi nadmiarów.

Patrz przykład 1 \emptyset na stronie.

PORÓWNAJ LICZBY ZMIENNOPRZECINKOWE

FORMA ROZKAZU:

EX <A> COMF.

Ekstrakod porównuje dwie liczby zmiennoprzecinkowe znormalizowane:

- LICZBA 1 - jest umieszczona w rejestrach 1,2,3.,
- LICZBA 2 - jest umieszczona od adresu = A w trzech kolejnych słowach pamięci,
- PORÓWNANIE - ustawienie wskaźników L,E,G, w rejestrze R0:

LICZBA 1 < LICZBA 2 L = 1 i E=G=0

LICZBA 1 = LICZBA 2 E = 1 i L=G=0

LICZBA 1 > LICZBA 2 G = 1 i L=E=0

Sposoby adresowania i ich skutki są analogiczne, jak w przypadku innych ekstrakodów działających na liczbach zmiennoprzecinkowych, patrz przykłady 15, 16, 17 na str. 9-5, 9-6.

ZMIEN ZNAK LICZBY ZMIENNOPRZECINKOWEJ

FORMA ROZKAZU:

EX < A > CSGF.

- ARGUMENT - jest w rejestrach 1,2,3 i musi być znormalizowany,
- WYNIK - jest umieszczony w rejestrach 1,2,3.
- NADMIAR - 1/ adres programu obsługi błędu określa ekstrakod OBSŁUGA NADMIARÓW OV.
2/ jeśli ekstrakod OBSŁUGA NADMIARÓW OV nie był wcześniej wykonany, zadanie jest przerwane, a wydruk na monitorze określa przyczynę i adres ekstrakodu, w którym powstał nadmiar.

Patrz: Uwaga na str. 4,6.

Rozdział 6

FUNKCJE

	str.
ABS - podaj wartość bezwzględną liczby zmiennie- przecinkowej	6-2
SGN - podaj wartość funkcji signum	6-3
ENT - podaj wartość funkcji entier	6-4
SIN - podaj wartość funkcji sinus	6-5
COS - podaj wartość funkcji cosinus	6-6
SQRT - pierwiastek kwadratowy	6-7
LN - logarytm naturalny	6-8
EXP - funkcja e^x	6-9
ATG - arens tangens	6-10
ACTG - arens cotangens	6-11

PODAJ WARTOŚĆ BEZWZGLEDNĄ LICZBY ZMIENNOPRZECINKOWEJ

FORMA ROZKAZU:

EX <A> ABS.

- LICZBA /arg.funkcji/ - jest umieszczona w rejestrach 1, 2,3, /musi być znormalizowana/.
- WYNIK - jest umieszczony w rejestrach 1, 2,3.
- NADMIAR - 1/ adres programu obsługi błędu określa ekstrakod OBSŁUGA NADMIARÓW OV,
2/ jeśli ekstrakod OBSŁUGA NADMIARÓW OV nie był wcześniej wykonany, zadanie jest przerwane, a wydruk na monitorze określa przyczynę i adres ekstrakodu, w którym powstał nadmiar.

Patrz: Uwaga na str. 4.6.

PODAJ WARTOŚĆ FUNKCJI SIGNUM

FORMA ROZKAZU:

EX < A > SGN.

LICZBA /arg.funkcji/ - znormalizowana jest umieszczona w rejestrach 1,2,3,

WYNIK - jest umieszczony w rejestrach 1, 2,3, jako liczba zmiennoprzecinkowa:

-1.0 , gdy $LICZBA < 0$

0 , gdy $LICZBA = 0$

+1.0 , gdy $LICZBA > 0$

Patrz: Uwaga na str. 4.6.

PODAJ WARTOŚĆ FUNKCJI ENTIER

FORMA ROZKAZU:

EX <A> ENT.

- LICZBA /arg.funkcji/ - znormalizowana jest umieszczona w rejestrach 1,2,3,
- WYNIK - jest umieszczony w rejestrach 1, 2,3, jako liczba zmiennoprzecinkowa.

Patrz: Uwaga na str. 4.6.

SINUS

FORMA ROZKAZU:

EX <A> SIN.

ARGUMENT FUNKCJI
/w radianach/

- jest znormalizowaną liczbą
zmiennoprzecinkową umieszczoną
w rejestrach 1,2,3.

WYNIK

- jest umieszczony w rejestrach
1,2,3.

Patrz: Uwaga na str. 4.6.

COSINUS

FORMA ROZKAZU:

EX <A> COS.

ARGUMENT FUNKCJI
/w radianach/- jest liczbą znormalizowaną
umieszczoną w rejestrach 1,2,3,

WYNIK

- jest umieszczony w rejestrach
1,2,3.

Patrz: Uwaga na str. 4.6.

PIERWIASTEK KWADRATOWY

FORMA ROZKAZU:

EX < A > SQRT.

- LICZBA /arg.funkcji/ - znormalizowana jest umieszczona w rejestrach 1,2,3,
- WYNIK - jest umieszczony w rejestrach 1,2,3.
- WYJŚCIA Z EKSTRAKODU - wg ŚLADU gdy LICZBA < 0 /obliczenia nie są wykonywane/
wg ŚLAD + 1 gdy LICZBA 0 >

Patrz: przykład str. 9-6.

LOGARYTM NATURALNY

FORMA ROZKAZU:

EX <A> LN.

- LICZBA /arg.funkcji/ - znormalizowana jest umieszczona w rejestrach 1,2,3,
- WYNIK - jest umieszczony w rejestrach 1,2,3.
- WYJŚCIA Z EKSTRAKODU - wg ŚLADU gdy LICZBA 0 /obliczenia nie są wykonywane/,
wg ŚLADU + 1 gdy LICZBA 0

Współpraca z tym ekstrakodem jest analogiczna jak w przypadku ekstrakodu "sgrt".

Patrz: Uwaga na str. 4.6.

FUNKCJA e^x

FORMA ROZKAZU:

EX <A> EXP.

ARGUMENT FUNKCJI

- jest liczbą znormalizowaną umieszczoną w rejestrach 1,2,3,

WYNIK

- jest umieszczony w rejestrach 1,2,3.

NADMIAR

- 1/ adres programu, obsługi błędu określa ekstrakod OBSŁUGA NADMIARÓW OV,
- 2/ jeśli ekstrakod OBSŁUGA NADMIARÓW OV nie był wcześniej wykonany, zadanie jest przerwane, a wydruk na monitorze określa przyczynę i adres ekstrakodu, w którym powstał nadmiar.

Patrz: Uwaga na str. 4.6.

ARCUS TANGENS

FORMA ROZKAZU:

EX <A> ATG.

LICZBA /arg.funkcji/ - znormalizowana jest umieszczona
w rejestrach 1,2,3,

WYNIK - jest umieszczony w rejestrach
1,2,3.

Patrz: Uwaga na str. 4.6.

ARCUS COTANGENS

FORMA ROZKAZU:

EX <A> ACTG.

LICZBA /arg.funkcji/

- znormalizowana jest umieszczona
w rejestrach 1,2,3,

WYNIK

, - jest umieszczony w rejestrach
1,2,3.

Patrz: Uwaga na str. 4.6.

Rozdział 7

ZMIANA LICZB

	str.
FLOW - zamień liczbę całkowitą krótką na zmiennoprzecinkową,	7-2
FLOT - zamień liczbę całkowitą długą na zmiennoprzecinkową	7-3
INTW - zamień liczbę zmiennoprzecinkową na całkowitą krótką	7-4
INTI - zamień liczbę zmiennoprzecinkową na całkowitą długą	7-5

ZAMIEŃ LICZBĘ CAŁKOWITĄ KRÓTKĄ NA ZMIENNOPRZECINKOWĄ

ZAMIEŃ LICZBĘ CAŁKOWITĄ KRÓTKĄ NA ZMIENNOPRZECINKOWĄ

FORMA ROZKAZU:

EX <A> FLOW.

A - jest liczbą całkowitą krótką,
którą należy zamienić na postać
zmiennoprzecinkową.

WYNIK - jest znormalizowaną liczbą
zmiennoprzecinkową w rejest -
rach 1,2,3.

ZAMIEŃ LICZBĘ CAŁKOWITĄ DŁUGA NA ZMIENNOPRZECINKOWĄ

FORMA ROZKAZU:

EX <A> FLOI.

LICZBA DŁUGA z rejestrów 1,2 jest zamieniona na znormalizowaną liczbę zmiennoprzecinkową w rejestrach 1,2,3.

Patrz: Uwaga na str. 4.6.

ZAMIEŃ LICZBĘ ZMIENNOPRZECINKOWĄ NA CAŁKOWITĄ KRÓTKĄ

FORMA ROZKAZU:

EX <A> INTW.

LICZBA ZMIENNOPRZECINKOWA z rejestrów 1,2,3, zostaje zaokrąglona i po zamianie na liczbę całkowitą krótką jest umieszczona w rejestrze wskazanym przez efektywny argument drugi.

Gdy podczas zamiany postaci liczby okazuje się, że powstała liczba krótka przekracza zakres liczb krótkich sygnalizowany jest nadmiar poprzez zapalenie bitu V w RO, działanie nie jest wykonywane, maszyna przechodzi do wykonania następnego rozkazu.

W przypadku gdy efektywny argument drugi operacji jest $\neq 1...7$ działanie programu zostaje zatrzymane, a na Teletype drukowany jest tekst:

ALARM < adres > WRONG PARAMETER

gdzie < adres > : = adres pierwszej komórki za ekstrakodem.

ZAMIEŃ LICZBĘ ZMIENNOPRZECINKOWĄ NA CAŁKOWITĄ DŁUGĄ

FORMA ROZKAZU:

EX < A > INTI.

LICZBA ZMIENNOPRZECINKOWA z rejestrów 1,2,3, jest zaokrąglona i po zamianie na liczbę całkowitą DŁUGĄ umieszczona w rejestrach 1,2.

W przypadku, gdy powstała w wyniku zamiany liczba długa przekracza zakres dopuszczalny dla liczb długich sygnalizowany jest nadmiar przez zapalenie bitu V w RO.

Działanie nie jest wykonywane. Maszyna przechodzi do pobrania i wykonania następnego rozkazu programu.

Patrz: Uwaga na str. 4.6.

Rozdział 8

KOMUNIKACJA Z SYSTEMEM

	str.
CHAR - umieść ostatni przeczytany znak w rejestrze	8-2
ESYS - wróć do systemu	8-3

UMIEŚĆ OSTATNI PRZECZYTANY ZNAK W REJESTRZE

FORMA ROZKAZU:

EX < A > CHAR.PARAMETR.

<A> = 0...k

- jest symbolicznym numerem urządzenia wejścia,

PARAMETR = 1....7 ,

- jest numerem rejestru, w którym ma być umieszczony ostatni znak alfanumeryczny wczytany z urządzenia wskazanego przez <A> .

Ekstrakod ten pozwala odzyskać ostatni wczytany przez dane urządzenie znak, niezależnie od tego jakim ekstrakodem ręcznie było ostatnie czytanie z danego urządzenia.

WRÓĆ DO SYSTEMU OPERACYJNEGO

FORMA ROZKAZU:

EX < A > ESYS.

Przerywa wykonywanie programu, drukuje na monitorze:

ALARM < LICZBA OKT. 1 > < LICZBA OKT. 2 > FINISH

gdzie: < LICZBA OKT. 1 > := adres następnej komórki za ekstrako-
dem,

< LICZBA OKT. 2 > := efektywny argument drugi.

Rozdział 9

PRZYKŁADY EKSTRAKODÓW

	Str.
EKSTRAKODY WEJŚCIA I WYJŚCIA	9-2
EKSTRAKODY DZIAŁAJĄCE NA LICZBACH ZMIENNOPRZECINKOWYCH	9-3
DZIAŁANIE EKSTRAKODU OV	9-5
FUNKCJE	9-6

EKSTRAKODY WEJŚCIA I WYJŚCIA

Przykład 1: ex, 1. rec. 7.

R1 = 2, znak wczytywany: a/65 w kodzie ISO-7/
z wejścia oznaczonego numerem symbolicznym 2 zo
stanie wczytany do rejestru 7 kod litery a.

Stan R7 po wykonaniu ekstrakodu:

R7 = 65

Stan pozostałych rejestrów nie ulegnie zmianie.

Przykład 2: ex, 1 & 2. prc.7.

R1 = 25 R2 = 550 S/575/ = 2 R7 = 65

Litera a /65 w kodzie ISO-7/ zostanie wyprowa -
dzona na wyjście o numerze symbolicznym 2.

Przykład 3: ex/nrwe/ ret.buf.!!+.

S/nrwe/ = 1

buf = 500

tekst na taśmie: czytaj do + a dalej nie.

Po wykonaniu ekstrakodu bufor będzie zapełniony
w następujący sposób:

S/500/: c z

S/501/ : y t

S/502/ : a j

S/503/ : d

S/504/ : o

S/505/ : 128 128

a R4 = 506.

Przykład 4: ex/buf/ reow.1.

S/buf/ : 0 1

S/buf + 1/ : 0 2

Po wykonaniu ekstrakodu:

R1 = 12

R4 = buf + 2

Pozostałe rejestry nie ulegają zmianie.

EKSTRAKODY DZIAŁAJĄCE NA LICZBACH ZMIENNOPRZECINKOWYCH

Argument pierwszy działań znajduje się w rejestrach R1, R2, R3. Argument drugi działań znajduje się w trzech kolejnych komórkach pamięci. Oba argumenty są w postaci znormalizowanych liczb zmiennoprzecinkowych.

Przykład 5: ex, 5. addf.

$$R5 = 500, R4 = 37.$$

Zawartość trzech kolejnych komórek pamięci o adresach 500, 501, 502 będzie traktowana jako drugi argument działania. Stan rejestrów po wykonaniu działania:

R1, R2, R3 := wynik, R4 = 500, R5, R6, R7 - bez zmiany.

Wynik jest liczbą zmiennoprzecinkową znormalizowaną.

Przykład 6: ex, 4. subf.

$$R4 = 700.$$

Zawartość trzech kolejnych komórek pamięci o adresach 700, 701, 702 będzie traktowana jako drugi argument działania. Stan rejestrów po wyjściu z ekstrakodu:

R1, R2, R3 := wynik w postaci liczby zmiennoprzecinkowej znormalizowanej.

R4, R5, R6, R7 - bez zmiany.

Przykład 7: ex, 6'. mplf.

$$R6 = 1000. S/R6/ = 400.$$

Zawartość komórek 400, 401, 402 będzie traktowana jako argument drugi. Stan rejestrów po wyjściu z ekstrakodu:

R1, R2, R3 := wynik w postaci liczby zmiennoprzecinkowej,

$$R4 := 400$$

R5, R6, R7 - bez zmiany.

Przykład 8: ex, 5 & 4. divf.

R4 = 10, R5 = 390

Zawartość komórek 400, 401, 402 zostanie pobrana, jako drugi argument działania.

Stan rejestrów po wyjściu z ekstrakodu:

R1,R2,R3:= wynik - liczba zmiennoprzecinkowa znormalizowana,

R4:= 400

R5,R6,R7 - bez zmiany.

Przykład 9: ex, 5, & 4. addf.

R4 = 10, R5 = 390. S/400/ = 500

Zawartość komórek o adresach 500, 501, 502 zostanie pobrana jako drugi argument działania.

Stan rejestrów po wyjściu z ekstrakodu:

R1, R2, R3:= wynik

R4 = 500

R5,R6,R7 - bez zmiany.

Przykład 10: ex(500)subf.

R4 = 0

Jednostka centralna potraktuje zawartość komórek pamięci o adresach 500, 501 i 502 jako argument drugi działania. Po wyjściu z ekstrakodu stan rejestrów będzie następujący:

R1,R2,R3:= wynik

R4:= 500

R5,R6,R7 - bez zmiany.

Przykład 11: ex(400)mlf.

R4 = 10, S/400/ = 1000

Zawartość komórek pamięci o adresach 1000, 1001, 1002 będzie pobrana jako drugi argument działania. Stan rejestrów po zakończeniu ekstrakodu:

R1,R2,R3:= wynik R4 = 1000

R5,R6,R7 - bez zmiany.

Przykład 12: $\text{ex}(4 \& 4) \text{ divf.}$

$R4 = 400$

Argument drugi działania znajduje się w komórkach pamięci o adresach 404, 405, 406. Stan rejestrów po wyjściu z ekstrakodu:

$R1, R2, R3 := \text{wynik}$

$R4 := 404$

$R5, R6, R7$ - bez zmian.

Przykład 13: $\text{addf}(4 \& 4^{\wedge})$

$R4 = 1006, S/1010/ = 400$

Argument drugi działania znajduje się w komórkach pamięci o adresach 408, 401, 402. Stan rejestrów po wyjściu z ekstrakodu:

$R1, R2, R3 := \text{wynik}$

$R4 := 1010$

$R5, R6, R7$ - bez zmian.

Uwaga: Należy pamiętać, że po wyjściu z ekstrakodu w rejestrze $R4$ znajduje się zawsze efektywny argument drugi. Wynik znajduje się w trzech pierwszych rejestrach i jest liczbą znormalizowaną, o ile argumenty działania były liczbami zmiennoprzecinkowymi znormalizowanymi. Pozostałe rejestry nie zmieniają zawartości.

DZIAŁANIE EKSTRAKODU OV

Przykład 14: $S/400/ : \emptyset.$

$S/401/ : \text{jpt}, 11.$

$S/402/ : \text{jpt}, 30.$

$S/403/ : \text{ex}, 4. \text{ esys.}$

$S/500/ : \text{ex}/400/\text{OV.}$

$S/503/ : \text{divf}, 7.$

$S/505/ : \text{ad}, 7, 6.$

Zawartość rejestrów przed wejściem do ekstrakodu dzielenia:

R1,R2,R3=dzielnia, w postaci liczby zmiennoprzecinkowej.

R4 = 0

R5 = 1, R6 = 2, R7 = 600

S/600/ = 0, S/601/, S/602/ = 0

Zawartość komórek pamięci o adresach 600, 601, 602 będzie traktowana jako dzielnik.

W tym przypadku dzielnik równa się 0. Zostaje wykryta chęć dzielenia liczby zmiennoprzecinkowej przez 0. Wcześniej w programie został wykonany ekstrakod obsługi nadmiarów, zatem adres komórki następnej po ekstrakodzie dzielenia /505/ zostanie zapamiętany w pamięci pod adresem 400, a do rejestru rozkazów zostanie pobrany rozkaz z komórki 402. Stan rejestrów w tym momencie:

R1,R2,R3 - bez zmiany

R4:=600

R5,R6,R7 - bez zmiany

S/400/:=505

IC:= 402.

FUNKCJE

Przykład 15: ex, 4. flow.

R4 = 6, R1 = R2 = R3 = 0

Zawartość rejestru R4 zostanie zamieniona na liczbę zmiennoprzecinkową i umieszczona w rejestrach R1,R2,R3.

Po wyjściu z ekstrakodu stan rejestrów będzie następujący:

R1:= 03

R2:= 0600000

R3:= 0

R4,R5,R6,R7 - bez zmiany.

Przykład 16: ex, 4. flow.

R4 = 6 ; S/6/ = 4 ; R1 = R2 = R3 = 10

Zawartość komórki pamięci o adresie 6 zostanie zamieniona na liczbę zmiennoprzecinkową.

Stan rejestrów po zakończeniu ekstrakodu będzie następujący:

R1:= 03

R2:= 040000

R3:= 0

R4:= 4

R5,R6,R7 - bez zmiany.

Przykład 17: ex(4) flow.

Liczba 4 zostanie zamieniona na postać zmiennoprzecinkową. Stan rejestrów po wyjściu z ekstrakodu:

R1:= 03

R2:= 040000

R3:= 0

R4:= 4

R5,R6,R7 - bez zmiany.

Przykład 18: ex, 4. sqrt.

a : jpt, 20.

b : ex, 4. esys.

1/ R1,R2,R3:= 0. 4E + 1

Ze względu na to, że liczba pierwiastkowana jest dodatnia maszyna wykona działanie i po wyjściu z ekstrakodu pobierze do wykonania rozkaz z komórki oznaczonej etykietą "b".

2/ R1,R2,R3:= - 1,44E + 2

Liczba pierwiastkowana ujemna. Maszyna nie wykona żadnych obliczeń - stan rejestrów R1, R2, R3 nie ulegnie zmianie, nastąpi wyjście z ekstrakodu i pobranie zawartości komórki oznaczonej w przykładzie etykietą "a", jako następnego rozkazu.

Patrz: Uwaga na stronie 4.6.

SIMP-ZEDOK z/ 507/72 n. 500

