

WOJSKOWA AKADEMIA TECHNICZNA  
im. Jarosława Dąbrowskiego

---

Jan Szmelter

JĘZYK SAS  
dla  
MASZYNY CYFROWEJ URAŁ-2

WARSZAWA - 1968

WOJSKOWA AKADEMIA TECHNICZNA  
im. Jarosława Dąbrowskiego

---

Jan Szmelter

JĘZYK SAS

dla  
MASZYNY CYFROWEJ URAL-2

Biblioteka Główna  
Wojskowej Akademii Technicznej

A I - 259



00-000636-0000

---

Warszawa - 1968

## SPIS TRESCI

CZESC I - OGOLNE WIADOMOSCI O MASZYNNIE "URAL-2" .....	str. 7
§ 1 Liczba całkowita .....	str. 7
§ 2 Liczba stałoprzecinkowa .....	str. 10
§ 3 Liczba zmienoprzecinkowa .....	str. 11
§ 4 Pamięć i rejestrty .....	str. 11
§ 5 Urządzenia wejścia i wyjścia .....	str. 13
 CZESC II - ROZKAZY MASZYNNY "URAL-2" .....	str. 17
§ 6 Budowa rozkazu .....	str. 19
§ 7 Rozkazy wejścia i wyjścia .....	str. 20
§ 8 Kolejność wykonywania pracy rozkazów .....	str. 20
§ 9 Rozkazy pobierania .....	str. 22
§ 10 Rozkazy przesyłania .....	str. 24
§ 11 Działania arytmetyczne .....	str. 25
§ 12 Przesuwanie .....	str. 27
§ 13 Operacje logiczne .....	str. 28
§ 14 Modyfikacja .....	str. 30
§ 15 Współpraca z bębnem .....	str. 30
§ 16 Inne rozkazy .....	str. 31
 CZESC III - OPIS JĘZYKA SAS .....	str. 33
§ 17 Elementy tabulogramu .....	str. 34
§ 18 Program, rozdział, podprogram .....	str. 35
§ 19 Podprogram .....	str. 36
§ 20 Licznik, etykieta, słownik .....	str. 36
§ 21 Liczby .....	str. 38
§ 22 Rozkaz .....	str. 39
§ 23 Deklarowanie miejsc roboczych .....	str. 40
§ 24 Podprogramy biblioteczne .....	str. 42
 CZESC IV - KORZYSTANIE Z JĘZYKA SAS .....	str. 51
§ 25 Wczytywanie programów .....	str. 52
§ 26 Wzbogacanie biblioteki podprogramów .....	str. 53
 CZESC V - PRZYKŁADY .....	str. 55

CZĘŚĆ I

OGÓLNE WIADOMOŚCI O MASZYNNIE URAŁ-2

## OGÓLNE WIADOMOŚCI O MASZYNIE URAŁ-2

### § 1. Liczba całkowita

W praktyce codziennej posługujemy się liczbami DZIESIĘTNYMI, których zapisu dokonuje się za pomocą dziesięciu cyfr

0 1 2 3 4 5 6 7 8 9

Pisząc program posługujemy się często zapisem OKTALNYM, który tym różni się od dziesiętnego, że zawiera tylko osiem cyfr

0 1 2 3 4 5 6 7

Jeszcze prostszy jest zapis binarny zawierający tylko dwie cyfry

0 1

zwane BITAMI. Zapis binarny jest stosowany powszechnie w technice maszynowej. Na pulpicie sterowniczym bitowi = 0 odpowiada wygaszona lampka, a bitowi = 1 lampka świecąca się.

A oto porównanie trzech zapisów

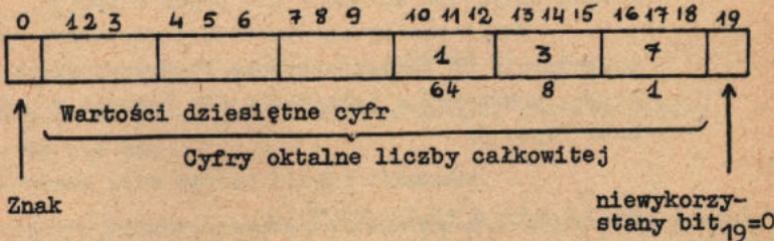
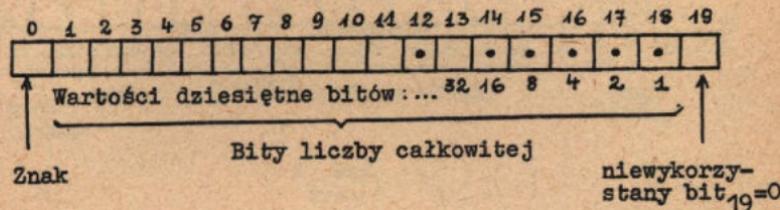
Dziesiętnie	Oktalnie	Binarnie
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	10	1000
9	11	1001
10	12	1010
11	13	1011
12	14	1100
13	15	1101
14	16	1110
15	17	1111
16	20	10000
17	21	10001
24	30	11000
32	40	100000
40	50	101000
48	60	110000
56	70	111000
64	100	1000000
72	110	1001000
128	200	10000000
320	500	101000000
511	777	111111111
512	1000	1000000000

Szczególnie proste są związki pomiędzy zapisem oktalnym i binarnym.

Poprostu każdej cyfrze oktalnej odpowiada trójka bitów, według następującego schematu:

Oktalnie - 0 1 2 3 4 5 6 7  
Binarnie - 000 001 010 011 100 101 110 111

W komórce maszyny liczby całkowite zapisują się binarnie. Komórka składa się z 20 bitów pomumerowanych od lewej do prawej, /rys. 1a/.



Rys. 1

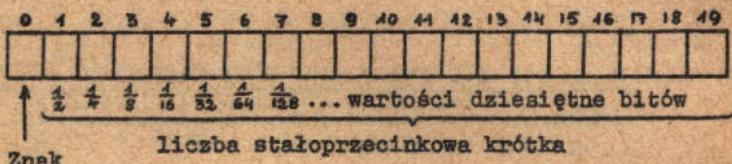
Na bicie zerowym zapisuje się znak liczby w postaci:

Plus	Bit = 0
Minus	Bit = 1

Bit<sub>19</sub> jest niewykorzystany i równy 0. Liczbę zapisaną binarnie, można interpretować także oktalnie /rys. 1b/, łącząc trójki bitów w cyfry zapisu oktalnego.

### § 2. Liczba stałoprzecinkowa

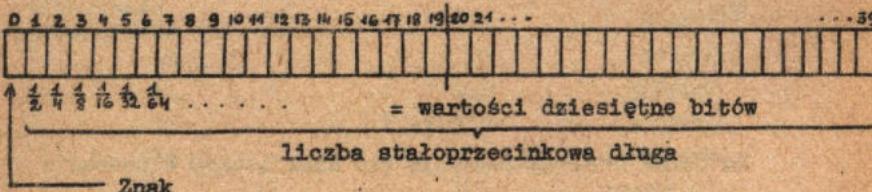
Liczby, których wartość bezwzględna jest mniejsza od jedności mogą być zapisywane w komórce maszyny w postaci stałoprzecinkowej jako liczby krótkie lub długie. Krótka liczba stałoprzecinkowa zapisana jest następująco: /rys. 2/



Rys. 2

Jak widać jest to liczba ułamkowa, której kropka pozycyjna znajduje się przed bitem 1.

Jeśli chcemy liczbę stałoprzecinkową wyrazić dokładniej, to zapisujemy ją w komórce zawierającej 40 bitów, powstałej z połączenia dwu komórek w jedną całość /rys. 3/



Rys. 3

### § 3. Liczba zmiennoprzecinkowa

Trzeci rodzaj liczb stosowanych w obliczeniach na maszynie URAL-2, są to liczby zmiennoprzecinkowe. Liczby te są zawsze zapisywane w komórce długiej w sposób podany na rys. 4.



Rys. 4

Wartość liczby zmiennoprzecinkowej określa się następująco:

$$\text{znak} \cdot \text{mantysa} \cdot 2^{\text{cecha}}$$

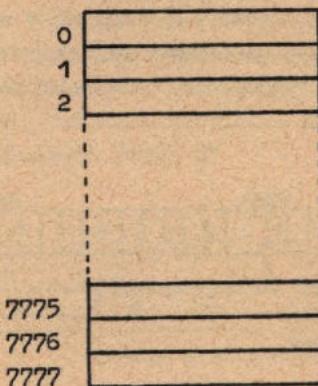
Jak widać cecha jest liczbą całkowitą zawartą między  $-63$  a  $+63$ .

Mantysa jest zawsze liczbą ułamkową.

Liczba zmiennoprzecinkowa nie równa zeru jest znormalizowana, gdy posiada pierwszy bit cechy równy 1. Liczba zmiennoprzecinkowa równa zeru ma wszystkie bity równe zero, liczby zmiennoprzecinkowe mniejsze bezwzględnie od  $2^{-63}$  traktowane są jako zero.

### § 4. Pamięć i rejestrzy

Pamięć maszyny składa się z 4096 komórek krótkich ponumerowanych oktalnie  $0, 1, 2, \dots, 7777$ . /rys. 5/



Rys. 5

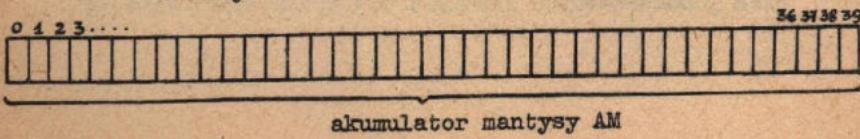
Numer komórki nazywamy jej **ADRESEM BEZWZGLĘDNYM**. Jeśli dwie kolejne komórki krótkie łączymy w jedną długą, to pierwsza z nich musi mieć adres parzysty, który jest traktowany jako bezwzględny adres całej komórki długiej.

Dodatkowo maszyna posiada pamięć bieżową, składającą się z komórek krótkich ponumerowanych oktalicie 0,1,2,.....,77777.

Można także korzystać z pamięci na taśmie magnetycznej. Pamięć jest uzupełniona przez specjalne komórki, zwane rejestrami. Działania arytmetyczne i operacje logiczne dokonują się w tych rejestrach. Dla programowania ważna jest znajomość działania następujących rejestrów:

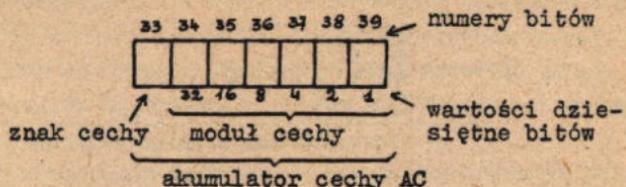
1/ AM = akumulator mantysy zawierający 40 bitów /rys.6/.

numery bitów



rys. 6

2/AC = akumulator cechy zawierający 7 bitów /rys.7/

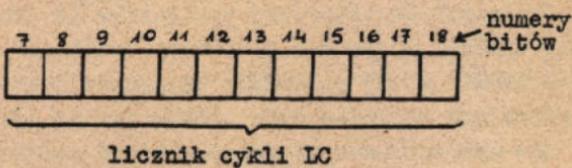


Rys. 7

3/  $\omega$  = jednobitowy akumulator znaku

4/  $\varphi$  = jednobitowy akumulator nadmiaru

5/ LC = licznik cykli zawierający 12 bitów /rys.8/



Rys. 8

6/  $LC_{19}$  = jeden bit długości licznika cykli.

#### § 5. Urządzenia wejścia i wyjścia

Aby wprowadzić informacje do maszyny trzeba je napisać na urządzeniu zwany DALEKOPISEM. Jest to rodzaj maszyny do pisania zaopatrzonej w klawisze, których naciśnięcie powoduje napisanie odpowiedniego znaku dalekopisowego na papierze. Treść przepisaną w ten sposób nazywamy TABULOGRAMEM.

Pisząc tabulogram otrzymujemy jednocześnie TAŚMĘ PERFOROWANĄ. Jest to wstęga papieru /rys. 9/ wzduż której w równych odstępach biegną dziurki prowadzące.



Rys. 9

Przy każdym uderzeniu klawisza, w rzędzie obok kolejnej dziurki prowadzącej, na taśmie drukuje się układ 5 dziurek. Każdej kombinacji dziurek odpowiada odpowiedni znak na tabulogramie, co zestawiono w tablicy 1.

Większość kombinacji dziurek może być traktowana dwojako w zależności od tego, czy traktujemy ją jako cyfry, czy jako litery. O tym jak je należy traktować, decydują znaki 33 = CYFRY i 37 = LITERY. Wszystkie kombinacje dziurek następujące po znaku cyfr traktowane są jako cyfry tak dugo, aż nie wystąpi znak liter, po którym kombinacje dziurek traktowane są jako litery.

Znak PUSTA TASMA nie odgrywa żadnej roli.

Znak KARETKA odpowiada przesunięciu karetki dalekopisu do początku wiersza tabulogramu. Znak LINIA odpowiada obrotowi wałka karetki o jedną linię. Znak SPACJA odpowiada pozostawieniu pustego odstępu w tekście tabulogramu.

Mając taśmę perforowaną można ją odczytać posługując się tablicą 1. Znacznie łatwiejsze i szybsze jest mechaniczne odczytywanie taśmy. Można ją bowiem wprowadzić do specjalnego urządzenia, w którym czujniki według układu dziurek sterują klawiszami dalekopisu, który wydrukuje tabulogram automatycznie.

Wprowadzenie informacji do maszyny dokonuje się za pośrednictwem CZYTNIAKA. Czytnik przesuwa taśmę, bieżąc odpowiednimi czujnikami układ dziurek, które przekształca na układy impulsów elektrycznych wprowadzonych do komórek pamięci maszyny.

Wyrowadzenie informacji z maszyny dokonuje się za pośrednictwem PERFORATORA. Jest to urządzenie sterowane impulsami elektrycznymi maszyny, dziurkujące taśmę według tych impulsów. Na podstawie tej taśmy otrzymujemy automatycznie tabulogram wyników.

Jak widać taśma perforowana odgrywa rolę pośrednią. Istotną rolę ma postać tabulogramu. Dlatego dalszy opis języka SAS będzie opisem tabulogramu.

TABLICA 1. ZNACZENIE ZNAKOW DALEKOPISOWYCH

NR OKTALNY	UKLAD DZIUREK	CYFRY	ZNACZENIE PO WCZYTANIU ZNAKU
00	:		
01	: 0	5	PUSTA TASMA T
02	: 0		KARETKA 0
03	: 00	9	
04	: 0		SPACJA
05	: 0 0		
06	: 00		H
07	: 000		N M
10	0:		LINIA
11	0: 0	)	L R
12	0: 0	4	G
13	0: 00	]	
14	0-0	8	I
15	0-0 0	0	P C
16	0-00	:	V
17	0-000	=	
20	0 :	3	E
21	0 : 0	+	Z
22	0 : 0	*	D
23	0 : 00	*	B
24	0-0	,	S
25	0-0 0	6	Y F
26	0-00	[	X
27	0-000	/	
30	00 :	-	A W
31	00 : 0	2	J
32	00 : 0	:	
33	00 : 00		CYFRY
34	00-0	7	
35	00-0 0	1	U O K
36	00-00	(	
37	00-000		LITERY

CZĘŚĆ II

ROZKAZY MASZYNY URAL-2

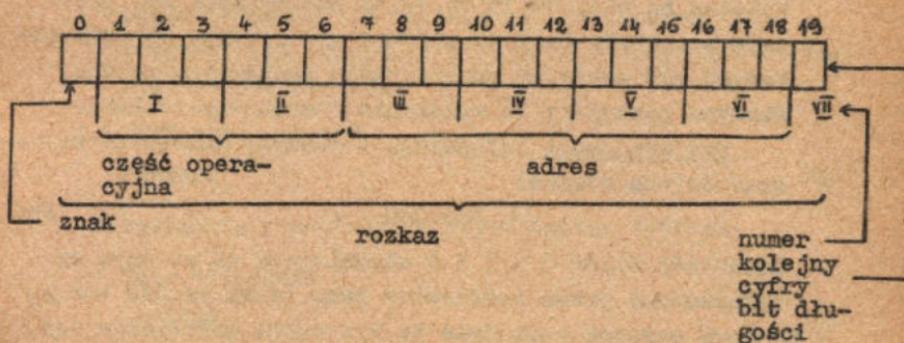
ROZKAZY MASZYNY URAL-2

§ 6. Budowa rozkazu

ROZKAZ na tabulogramie ma postać liczby oktalnej siedmiocyfrowej, np.

a/ 0237764 /1/  
która może ewentualnie być poprzedzona znakiem minus, np.

b/ -0237764 /2/  
Ostatnia cyfra rozkazu, tzw. BIT DŁUGOŚCI może być równa tylko 4 lub 0. Rozkaz /b/ wprowadza się do krótkiej komórki pamięci zgodnie z rys. 10



Rys. 10

Dwie pierwsze cyfry rozkazu stanowią tzw. CZĘŚĆ OPERACYJNĄ, która określa rodzaj czynności, jaką maszyna ma wykonać. Cyfry IIII IV V VI stanowią liczbę oktalną zwaną ADRESEM. Określa ona oktalicie numer komórki pamięci, do której odnosi się dany rozkaz. Jeśli przed rozkazem napisany jest znak minus, to rozkaz będzie dotyczył adresu napisanego na tabulogramie zmniejszonemu o aktualną zawartość licznika cykli LC./rys.8/.

Rozkazy zapisane są jeden po drugim w komórkach pamięci operacyjnej maszyny, w takiej samej kolejności jak są napisane na tabulogramie.

§ 7. Rozkazy wejścia i wyjścia

Wprowadzenie informacji do maszyny następuje za pośrednictwem rozkazu

4000010

/3/

Wykonują się następujące czynności:

Czytnik czyta jeden rządka taśmy. Jeśli była to pusta taśma, to przesuwa się ona do następnego rządka i czytanie przebiega od początku dalej. Jeśli wczytany zostanie rządak nie pusty, to jego wartość zostanie wpisana na bity 14 15 16 17 18 akumulatora AM, przy czym pozostałe bity AM zostaną wyzerowane. Czytnik zatrzyma się gotów do przeczytania następnego rządka. Maszyna przejdzie do wykonania następnego rozkazu.

Wyprowadzenie informacji z maszyny następuje za pośrednictwem rozkazu

3300020

/4/

Zawartość bitów 0 1 2 3 4 akumulatora AM zostaje wydziurkowana przez perforator jako jeden rządak taśmy, po czym maszyna przejdzie do wykonania następnego rozkazu.

§ 8. Kolejność wykonywania pracy rozkazów

W zasadzie rozkazy są wykonywane w takiej kolejności, w jakiej są napisane w pamięci /a tym samym i na tabulogramie/. Kolejność tą zmieniać mogą następujące rozkazy:

a/ Skok bezwzględny

22NNNN0

/5/

Maszyna przechodzi do wykonywania dalszej części programu począwszy od rozkazu napisanego w komórce o adresie oktalnym NNNN.

b/ Skok ze Śladem

22NNNN4

/6/

Maszyna przechodzi do wykonywania dalszej części programu począwszy od rozkazu zapisanego pod adresem NNNN+1. Jednocześnie do komórki o adresie NNNN wpisuje się SLAD skoku, to znaczy rozkaz  
22~~MMMM~~MO.

w którym ~~MMMM~~ oznacza powiększony o jednośc adres komórki, w której zapisany jest rozkaz /6/

c/ Skok przy  $\omega = 1$

21NNNNNO

/7/

Jeśli  $\omega = 1$ , maszyna przechodzi do wykonywania rozkazów począwszy od zapisanego pod adresem NNNN.

W przeciwnym przypadku kolejność wykonywania rozkazów nie zmienia się.

d/ Skok przy  $\omega = 0$

21NNNN4

/8/

Działanie przeciwe do poprzedniego rozkazu.

e/ Skok cykliczny

24NNNN4

/9/

Jeśli licznik cykli LC = 0, to kolejność wykonywania rozkazów nie ulega zmianie. Jeśli LC ≠ 0 to maszyna przechodzi do wykonywania rozkazów począwszy od napisanego pod adresem NNNN. Jednocześnie zawartość licznika cykli zmniejsza się o jednośc gdy LC<sub>19</sub> = 0 lub o dwa, gdy LC<sub>19</sub> = 1

f/ Skok przy N-tym kluczu

23000NO

/10/

Na pulpicie sterowniczym znajdują się klucze ponumerowane 1 2 3 4 5 6 7.

Jeśli klucz o numerze N jest włączony, zostaje opuszczony rozkaz występujący po /10/. Przy wyłączonym kluczu N kolejność wykonywania rozkazów jest niezmieniona.

g/ Stop

37NNNNO

/11/

Maszyna zatrzymuje się. Przy ponownym naciśnięciu przycisku START, dalsza praca maszyny zaczyna się od wykonywania następnego rozkazu.

Adres NNNN nie odgrywa roli.

h/ Stop lub skok przy  $\Psi = 1$

Jeśli znajdujący się na pulpicie klucz BLOKADA  $\Psi$  jest włączony, maszyna nie reaguje na wartość rejestru  $\Psi$ .

Jeśli klucz BLOKADA  $\Psi$  jest wyłączony i włączony jest klucz "STOP przy  $\Psi = 1$ ", to maszyna zatrzymuje się po każdym rozkazie, po którym w rejestrze  $\Psi$  pojawia się wartość 1. Ponowne naciśnięcie przycisku START zeruje rejestr  $\Psi$  i uruchamia pracę maszyny poczawszy od następnego rozkazu po zatrzymaniu.

Jeśli klucz BLOKADA  $\Psi$  i "STOP przy  $\Psi = 1$ " są wyłączone, to przy pojawianiu się wartości jeden w rejestrze  $\Psi$ , maszyna opuszcza jeden rozkaz i rejestr  $\Psi$  zostaje wyzerowany.

§ 9. Rozkazy pobierania

Pobieranie stałoprzecinkowe liczby krótkiej

02NNNNNO

/12/

Zawartość komórki krótkiej o adresie NNNN przepisuje się na bity 0 do 19 akumulatora, a na pozostałe bity 20 do 39 i do akumulatora cechy wpisują się zera. Do rejestrów  $\omega$  przepisuje się zawartość bitu zerowego akumulatora,  $\Psi = 0$ .

Pobieranie stałoprzecinkowe liczby długiej

02NNNN4

/13/

Zawartość komórki o parzystym adresie NNNN oraz następnej, traktowana jako liczba stałoprzecinkowa dłuża /rys.3/ przepisuje się do akumulatora mantysy.

Reszta jak po rozkazie /12/

Pobieranie zmienoprzecinkowe liczby długiej /14/

42NNNN4

Zawartość komórki o parzystym adresie NNNN oraz następnej traktowane jest jako liczba zmienoprzecinkowa /rys. 4/.

Bit od 0 do 32 przepisują się do akumulatora mantysy, którego bity 33 do 39 zostają wyzerowane. Cecha wraz ze znakiem przepisuje się do akumulatora cechy. Dalej następuje normalizacja liczby w akumulatorze.

Bit zerowy akumulatora przepisuje się do rejestru  $\omega$ . Jeśli cecha pbranej liczby jest mniejsza od -63, lub większa od +63 to rejestr  $\varphi = 1$ . W przeciwnym przypadku  $\varphi = 0$ .

Pobieranie zmienoprzecinkowe liczby krótkiej

42NNNNNO

/15/

Bit 0 - 19 liczby zmienoprzecinkowej pobierane są z komórki o adresie NNNN a pozostałe są zerami.

Dalej jak w rozkazie /14/

Pobranie cyklu krótkie

25NNNNNO

/16/

Liczniak cykli przyjmuje wartość równą NNNN. Bit<sub>19</sub> licznika cykli przyjmuje wartość = 0.

Pobranie cyklu długie

25NNNN4

/17/

Liczniak cykli przyjmuje wartość równą NNNN. Bit<sub>19</sub> licznika cykli przyjmuje wartość = 1.

## § 10. Rozkazy przesyłania

### Przesyłanie stałoprzecinkowe liczby krótkiej

16NNNNNO

/18/

Zawartość bitów 0 do 19 akumulatora mantysy przepisuje się do komórki pamięci o adresie NNNN, a do rejestru wpisuje się zawartość bitu zerowego  $\Psi = 0$ .

### Przesyłanie stałoprzecinkowe liczby długiej

16NNNNN4

/19/

Zawartość akumulatora mantysy przepisuje się do dwu kolejnych komórek pamięci, z których pierwsza ma parzysty adres NNNN. Rejestry  $\omega$  i  $\Psi$  zachowują się jak po rozkazie /18/.

### Przesyłanie zmienoprzecinkowe liczby długiej

56NNNNN4

/20/

Zawartość akumulatora normalizuje się, a potem ulega zaokrągleniu, które polega na tym, że do modułu mantysy dodaje się jedynka na bicie 32 gdy bit 33 był jedynką, oraz bity 33 do 39 zostają wyzerowane. Potem bity 0 do 32 akumulatora mantysy i 33 do 39 akumulatora cechy przepisują się do dwóch kolejnych komórek pamięci, z których pierwsza ma parzysty adres NNNN. Rejestr  $\omega$  przyjmuje wartość bitu 0. Jeśli przy zaokrągleniu wystąpi nadmiar  $\Psi = 1$ , poza tym  $\Psi = 0$ .

### Slad cyklu

27NNNNNO

/21/

Jeśli zawartość licznika cykli wynosi MMMM, a zawartość bitu<sub>19</sub> licznika cykli B, /B=0 lub B=4/ to do komórki krótkiej o adresie NNNN wpisuje się rozkaz

25MMMNBB

11. Działania arytmetyczne

Dodawanie stałoprzecinkowe liczby długiej

01NNNN4

/22/

Do zawartości akumulatora mantysy dodaje się dłuża liczba stałoprzecinkowa zapisana w dwu kolejnych komórkach pamięci, z których pierwsza posiada parzysty adres NNNN. Wynik zapisuje się w akumulatorze mantysy, znak w komórce zerowej akumulatora i w rejestrze  $\omega$ . Rejestr przyjmuje wartość 0, gdy wynik mieści się w akumulatorze, a wartość 1 gdy nie mieści się.

Akumulator cechy zeruje się.

Dodawanie zmiennoprzecinkowe liczby krótkiej

01NNNN0

/23/

Analogicznie jak poprzednio, z tym że starsza część dodawanej liczby długiej jest pobrana z komórki o adresie NNNN /który może być nieparzysty/, a młodsza jest zerem.

Odejmowanie stałoprzecinkowe liczby długiej

03NNNN4

/24/

przebiega analogicznie do /22/.

Odejmowanie stałoprzecinkowe liczby krótkiej

03NNNN0

/25/

przebiega analogicznie do /23/

Mnożenie stałoprzecinkowe przez liczbę długą

06NNNN4

/26/

Analogicznie do /22/. Wynik zawiera 32 bity dokładne

Mnożenie stałoprzecinkowe przez liczbę krótką

06NNNN0

/27/

Analogicznie do /23/. Wynik zawiera 19 bitów dokładnych. Jeśli jednak poprzednio w akumulatorze była liczba krótka, to wynik zawiera 38 bitów dokładnych.

Dzielenie stałoprzecinkowe przez liczbę długą

07NNNN4

/28/

Analogiczne do /22/. Wynik zawiera 33 bity dokładne.  
Przy dzieleniu przez zero maszyna zatrzymuje się.

Dzielenie stałoprzecinkowe przez liczbę krótką

07NNNNNO

/29/

Analogiczne do /23/. Dzielna powinna zawierać tylko 33 bity. Wynik dzielenia ma 19 bitów dokładnych. Przy dzieleniu przez zero maszyna staje.

Dodawanie zmennoprzecinkowe liczby dłużej

41NNNN4

/30/

Dla liczby zmennoprzecinkowej w akumulatorze dodaje się zmennoprzecinkowa pobrana z dwu kolejnych komórek pamięci, z których pierwsza ma parzysty adres NNNN. Wynik ulega normalizacji i zapisuje się w akumulatorze, a jego znak na zerowym miejscu akumulatora i w rejestrze  $\omega$ .

Jeśli wynik mieści się w akumulatorze, to  $\psi = 0$ , inaczej  $\psi = 1$ .

Dodawanie zmennoprzecinkowe liczby krótkiej

41NNNNNO

/31/

Analogicznie do /30/, z tym że starsza część pobieranej liczby brana jest z komórki pamięci o adresie NNNN /niekoniecznie parzystym/ a młodsza /a więc także cecha/ traktowana jest jako zero.

Odejmowanie zmennoprzecinkowe liczby dłużej

43NNNN4

/32/

analogicznie do /30/

Odejmowanie zmennoprzecinkowe liczby krótkiej

43NNNNNO

/33/

analogicznie do /31/

Odejmowanie wartości bezwzględnych liczb zmienno-przecinkowych

44NNNN4

/34/

analogicznie do /30/.

Mnożenie zmennoprzecinkowe przez liczbę długą

46NNNN4

/35/

analogicznie do /30/

Mnożenie zmennoprzecinkowe przez liczbę krótką

46NNNNNO

/36/

analogicznie do /31/

Dzielenie zmennoprzecinkowe przez liczbę długą

47NNNN4

/37/

analogicznie do /30/. Przy dzieleniu przez zero maszyna staje. Dzielna musi być liczbą znormalizowaną.

Dzielenie zmennoprzecinkowe przez liczbę krótką

47NNNNNO

/38/

analogicznie do /31/. Uwagi te same co przy /37/.

Formowanie znaku

10NNNNNO lub 10NNNN4

/39/

Na zerowy bit akumulatora i do rejestru  $\omega$  wprowadza się zerowy bit komórki pamięci o adresie NNNN.

Sumowanie kontrolne

26NNNNNO

/40/

Do zawartości akumulatora mantysy dodaje się stałoprzecinkowa dłuża liczba dodatnia, której bity 0 do 19 są zerami, a następne 20 do 39 są zawartością komórki pamięci o adresie NNNN. Dalej analogicznie do /22/.

§ 12. Przesuwanie

Arytmetyczne przesunięcie w lewo

1100NN4

/41/

Akumulator cechy zeruje się. Znak akumulatora mantysy

nie zmienia się. Bity 1 do 39 akumulatora mantysy przesuwają się o NN /oktalnie/ pozycji w lewo. Na najmłodsze bity wprowadzają się zera. Jeśli przesunięta liczba mieści się w akumulatorze, to  $\varphi = 0$ , inaczej  $\varphi = 1$ . Rejestr  $\omega$  przyjmuje wartość zerowego bitu akumulatora mantysy.

Arytmetyczne przesunięcie w prawo

1101NN4

/42/

Jak wyżej, tylko że przesunięcie następuje o NN /oktalnie/ miejsc w prawo. Bit najmłodsze giną, a wchodzące najstarsze są zerami.

Logiczne przesunięcie w lewo

1100NN0

/43/

Akumulator cechy zeruje się. Znak akumulatora mantysy traktowany jest jako jedna całość wraz z pozostałymi bitami akumulatora mantysy i przesuwa się razem z nimi o NN /oktalnie/ miejsc w lewo. Wychodzące bity giną, wchodzące bity są zerami. Rejestr  $\varphi = 0$ . Rejestr  $\omega$  przyjmuje wartość zerowego bitu akumulatora mantysy.

Logiczne przesunięcie w prawo

1101NNO

/44/

Analogiczne do 43, z tym że przesunięcie następuje w prawo o NN /oktalnie/ miejsc.

§ 13. Operacje logiczne

Koniunkcja liczbą długą

12NNNN4

/45/

Koniunkcji podlegają zawartość akumulatora mantysy AM i stałoprzecinkowa liczba dłuża pobrana z dwu kolejnych komórek pamięci, z których pierwsza ma parzysty adres NNNN. Wynik koniunkcji zapisuje się w akumulatorze mantysy AM, akumulator cechy AC nie zmienia się.

Każdy k-ty bit wyniku jest iloczynem k-tych bitów obu czynników koniunkcji /niezależnie od pozostałych bitów/. To samo dotyczy także bitu zerowego.

Rejestr  $\varphi = 0$ . Rejestr  $\omega = 1$  gdy wynik koniunkcji jest zerem. W przeciwnym przypadku  $\omega = 0$ .

#### Koniunkcja liczba krótka

12NNNN

/46/

Podobnie jak poprzednio, z tym że pobierana z pamięci liczba długa w starszej części równa jest zawartości komórki NNNN pamięci a w młodszej równa jest zeru.

#### Alternatywa liczba długa

13NNNN4

/47/

Alternatywie podlega zawartość akumulatora mantysy AM i liczba długa pobrana stało przecinkowo z dwu kolejnych komórek pamięci, z których pierwsza ma adres parzysty NNNN. Wynik zapisuje się w AM, przy czym akumulator cechy AC nie zmienia się.

Każdy k-ty bit wyniku /także i zerowy/ jest sumą logiczną k-tych bitów obu składników, według reguły:

$$0+0=0, \quad 0+1=1, \quad 1+0=1, \quad 1+1=1$$

Rejestr  $\varphi=0$ . Rejestr  $\omega=1$  gdy wynik alternatywy jest zerem. W przeciwnym przypadku  $\omega=0$

#### Alternatywa liczba krótka

13NNNN

/48/

Podobnie jak poprzednio, z tym że pobrana z pamięci liczba długa w starszej części równa jest zawartości komórki pamięci o adresie NNNN, a młodsza jest zerem.

#### Porównanie z liczbą długą

14NNNN4

/49/

W działaniu biorą udział te same składniki co w /47/.

Każdy k-ty bit wyniku jest zerem, gdy obie k-te bity składników są zgodne. W przeciwnym przypadku bit ten jest jednością. Wynik zapisuje się do akumulatora mantysy AM, akumulator cechy AC nie zmienia się.

Rejestr  $\psi = 0$ , rejestr  $\omega = 0$  gdy wynik operacji jest zerem, a w przeciwnym przypadku  $\omega = 1$ .

Porównanie z liczbą krótką

14NNNNNO

/50/

Jak poprzednio, z tym że pobierana z pamięci liczba dłuża w starszej części równa jest zawartości komórki o adresie NNNN, a w młodszej jest zerem.

§ 14. Modyfikacja

30NNNNNO

/51/

Powoduje zmodyfikowanie rozkazu napisanego bezpośrednio po /51/. Zamiast tego modyfikowanego rozkazu zostanie wykonany rozkaz będący sumą rozkazu modyfikowanego i zawartości komórki o adresie NNNN. Sumowanie dokonuje się następująco:

Rozkaz modyfikowany traktowany jest jako liczba dodatnia, której początkową cyfrą jest bit znaku. Zawartość komórki NNNN traktuje się tak samo, z tym jednak, że gdy zerowy bit jest jedynką pozostałe trzeba zastąpić ich negacjami. Te dwie krótkie liczby dodaje się binarnie. Jeśli powstaje nadmiar, to go się pomija dodając liczbę 0000004.

Wynik modyfikacji jest nowym rozkazem, który zostanie wykonany zamiast rozkazu modyfikowanego. Zapis w pamięci ferrytowej rozkazu modyfikowanego nie zmienia się przy tej operacji.

§ 15. Współpraca z bębnem

Zapis na bęben

52NNNNNO, OMMMMM, OOKKKKO

/52/

Są to trzy kolejne rozkazy. Zawartość komórki pamięci ferrytowej o adresie NNNN przepisze się do komórki pamięci bębnowej o adresie MMMMM. Zawartości następnych komórek pamięci ferrytowej przepiszą się do następnych komórek pamięci bębnowej w tej samej kolejności. Adresem ostatniej komórki pamięci ferrytowej, której zawartość ma się przepisać jest KKKK.

Na początku operacji /52/ akumulator samoczynnie zeruje się, po czym w trakcie przepisywania tworzy się w nim samoczynnie suma kontrolna zawartości przepisywanych komórek /według zasad działania /40/.

Licznik cykli zeruje się. Rejestry  $\varphi$  i  $\omega$  nie ulegają zmianom.

Uwaga: W serii rozkazów /52/ jedynie pierwszy wolno modyfikować operacją /51/ lub licznikiem cykli /2/.

Czytanie z bębna

51NNNNNO, OMMMMMO, OOKKKKO.

/53/

Wszystkie czynności i oznaczenia analogiczne do /52/ z tym, że teraz zawartość komórek pamięci bębnowej przepisuje się do komórek pamięci ferrytowej.

Czytanie kontrolne z bębna

51NNNN4, OMMMMMO, OOKKKKO,

/54/

Wszystkie oznaczenia analogiczne do /52/, z tym, że zawartości komórek pamięci bębnowej, są sumowane kontrolnie, ale nie zapisują się w komórkach pamięci ferrytowej.

Inne rozkazy

Istnieje jeszcze szereg rozkazów kierujących współpracą z mało używanymi urządzeniami maszyny. Aby nie zatrzeć opisu, tutaj zostały one pominięte.

CZĘŚĆ III

OPIS JĘZYKA SAS

OPIS JĘZYKA SAS

§.17. Elementy tabulogramu

Opis języka SAS jest opisem tabulogramu sporządzonego na dalekopisie i posiadającego swój odpowiednik na taśmie dalekopisowej.

- a/. Znak "PUSTA TASMA" nie odgrywa roli.
- b/. Znaki "CYFRY" i "LITERY" są interpretowane tak samo jak w dalekopisie. Zbędne znaki "CYFRY" i "LITERY" są pomijane.
- c/. Znaki "SPACJA" /odstęp/ "LINIA" i "KARETKA" odgrywają rolę symboli rozdzielających poszczególne wyrażenia tabulogramu. Zbędne znaki "SPACJA" i "KARETKA" są pomijane.
- d/. Dowolny ciąg znaków dalekopisowych zawarty między dwoma apostrofami jest traktowany jako komentarz i jest pomijany przez maszynę /z apostrofami włącznie/.
- e/. Średnik ";", ósemka "8", i dziewiątka "9" nie zawarte w nawiasach okrągłych stanowią sygnał przerwy wczytywani taśmy. Po ponownym naciśnięciu klucza "Start" wczytywanie przebiega dalej /i nic się nie psuje/.
- f/. Dowolny nieprzerwany ciąg złożony z liter:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
tworzy słowo. Pierwsze cztery litery słowa stanowią nazwę, w której pozostałe litery są pomijane. Dla krótkich słów nazwa może być jedno, dwu lub trzyliterowa.

- g/. Dowolny nieprzerwany ciąg cyfr

0 1 2 3 4 5 6 7

stanowi oktalna. Jest to liczba całkowita w zapisie oktalnym.

- h/. Ujęty w nawiasy okrągle "(") i napisany zgodnie z zasadami ALGOLU ciąg złożony ze znaków

0 1 2 3 4 5 6 7 8 9 + - .

stanowi liczbę zmiennoprzecinkową. Największy moduł zmiennoprzecinkowej  $10^{20}$ . Liczba mniejsza od  $10^{-20}$  traktowana jest jak 0/zero/

/Uwaga: niedozwolone są spacje pomiędzy znakami two- rzącymi liczbę/.

i/. Jeśli w zmiennoprzecinkowej nie występuje kropka po- zycyjna, to liczba przedstawia liczbę całkowitą.

Największy moduł całkowitej

260000.

#### Program, rozdział, podprogram

Programem nazywamy zbiór instrukcji, które maszyna ma wykonać celem rozwiązania interesującego nas zadania.

Program dzielimy na rozdziały ponumerowane oktalnie numerami 2,4,6,...72,

Na tabulogramie początek rozdziału zaznaczamy pisząc w osobnym wierszu

ROZDZIAŁ                    numer rozdziału                    /55/

a koniec rozdziału zaznaczamy pisząc

KONIEC                      /56/

Po wczytaniu kolejnego rozdziału maszyna zapisuje go na bęben idąc od końca, jeden po drugim. Przed pierw- szym rozdziałem programu należy umieścić w osobnym wierszu napis:

KASUJ ROZDZIAŁY                    /57/

Powoduje to wymazanie z bębna rozdziałów, które były nagrane w poprzednich programach. Po słowie KONIEC ostatniego rozdziału piszemy instrukcję.

START                    numer rozdziału                    /58/

Powoduje to zatrzymanie maszyny. Po naciśnięciu klucza START, następuje przepisanie z bębna żądanego rozdzia- łu do pamięci operacyjnej od jej miejsca zerowego i dalej skok do tego miejsca.

Jeśli chcemy aby w trakcie wykonywania działań przejść do innego rozdziału, należy w tym miejscu napisać rozkazy  
02 N, 22 7762.

/59/

gdzie N = adres komórki zawierającej numer rozdziału. Powoduje to przepisanie z bębną żądanego rozdziału do pamięci operacyjnej począwszy od jej miejsca zerowego i skok do tego miejsca.

§ 19. Podprogramem nazywamy fragment programu, który w wielu programach może się powtarzać. Podprogramy odróżniamy od siebie nazwami. Na tabulogramie początek podprogramu zaznaczamy pisząc w ostatnim wierszu

PODPROGRAM

nazwa

/60/

a koniec podprogramu pisząc w osobnym wierszu

KONIEC

/61/

Po wczytaniu kolejnego podprogramu maszyna zapisuje go na bębnie, jeden za drugim idąc od początku.

Aby podprogram dodać do rozdziału, należy w tym miejscu napisać

FUNKCJA

nazwa podprogramu

/62/

Wynika z tego, że podprogram, z którego chcemy korzystać musi być nagrany na bębnie wcześniej niż rozdział.

Jeśli chcemy skasować już nagrane podprogramy, /za wyjątkiem bibliotecznych/, to piszemy rozkaz

KASUJ PODPROGRAM

Przy pracy maszyny według danego programu można korzystać z całego bębną za wyjątkiem końcowych miejsc, na których zapisane są rozdziały programu.

§ 20. Licznik, etykieta, słownik

W trakcie wczytywania programu, kolejne rozkazy i informacje zapisują się do kolejnych komórek pamięci, począwszy od komórki o adresie zerowym. Równocześnie należy tworzyć licznik, który powinien być równy oktalnemu adresowi komórki pamięci, do której wpisują się wspomniane wyżej informacje. Licznik ten zmienia swoją wartość w miarę wczytywania programu według następujących reguł.

- a/. Po deklaracji /55/ i /60/ licznik rozkazów automatycznie przyjmuje wartość 0
- b/. Po wczytaniu rozkazu lub liczby całkowitej licznik wzrasta o jednośc
- c/. Po wczytaniu symbolu  
      `##`) /63/  
licznik doznaje zaokrąglenia w górę do wartości parzystej a na ewentualnie opuszczone puste miejsce wpisuje się zero.
- d/. Po wczytaniu liczby zmennoprzecinkowej, /która zapisuje się w dwu komórkach, przy czym pierwsza ma numer parzysty /licznik wzrasta o dwa
- e/. Po deklaracji /62/ licznik doznaje najpierw zaokrąglenia jak w pkt. c/, a potem zwiększa się o ilość komórek zajętych przepisywanym podprogramem.
- f/. Po wczytaniu symbolu  
      `[liczba oktalna]`) /64/  
licznik przyjmuje wartość napisanej liczby oktalnej
- g/. Po wczytaniu symbolu  
      `+ [liczba oktalna]`) /65/  
do licznika dodaje się napisana liczba oktalna
- h/. Po wczytaniu symbolu  
      `- [liczba oktalna]`) /66/  
licznik maleje o napisaną wartość oktalną

Etykieta nazywamy symbol w postaci

`[NAZWA])` /67/

albo

`[NAZWA] J` /68/

Każdy rozdział lub każdy podprogram ma swój osobny słownik. W słowniku każdej nazwie etykiety przypisuje się liczbę oktalną równą wartości licznika w miejscu napisania etykiety. Oczywiście w jednym słowniku nie może być dwu jednakowych nazw. Natomiast w różnych słownikach te same nazwy mogą oznaczać co innego.

W rozdziale obie formy /67/ i /68/ traktowane są jednakowo. W podprogramie etykieta /67/ zapisuje się tylko w jego słowniku, a etykieta /68/ dodatkowo przepisuje się do słownika rozdziału, do którego program był przepisany deklaracją /62/.

### § 21. Liczby

Liczba w programie może mieć postać liczby oktalnej całkowitej, oktalnej długiej, dziesiętnej całkowitej, dziesiętnej zmiennoprzecinkowej.

- a/ Liczba oktalna całkowita /krótka/. Zapisujemy ją w postaci siedmiocyfrowej liczby oktalnej zaczynającej się od znaku i zakończonej kropką, np

+0200224.

Liczba ta wpisuje się do komórki pamięci tak jak rozkaz /\$22/.

- b/ Liczba oktalna dłuża

Zapisujemy ją w postaci dwóch kolejnych liczb oktalnych całkowitych, przy czym pierwsza z nich musi być umieszczona w komórce o adresie parzystym.

- c/ Liczba całkowita dziesiętna

Zapisujemy ją normalny sposób, poprzedzając ewentualnie znakiem ujmując w nawiasy okrągłe, np.

(1968) (-25)

Uwaga: Cyfr tej liczby oraz znaku nie można rozdzielać spacją.

- d/ Liczba zmiennoprzecinkowa dziesiętna

Zapisujemy ją zgodnie z zasadami ALGOLU ujętą w nawiasy okrągłe. Liczbę tę trzeba napisać do dwóch komórek, z których pierwsza ma adres parzysty.

Przykłady:

(3.14) (+0.314<sub>10</sub>1) (311.0<sub>10</sub> -2)

Liczba ta musi zawierać kropkę pozycyjną.

Nie wolno jej znaków rozdzielać spacją.

2. Rozkaz

Budowa rozkazu została opisana w § 6. Obecnie podamy sposób zapisania rozkazu na tabulogramie.

Najprostszy zapis rozkazu ma postać liczby oktalnej,  
np:

znak      0200224      kropka lub przecinek  
operacja      adres      bit długości

/6

W tym zapisie nie wolno go rozdzielić spacją  
/w szczególności nie może być spacji między ostatnią  
cyfrą a kropką lub przecinkiem/.

- a/. Znak może być + lub -. Znak + można opuścić
- b/. Bit długości może być 4 lub 0
- c/. Opuścić można początkowe zera liczby oktalnej
- d/. W rozdzielaniu role kropki i przecinka są jednakowe.  
W podprogramie rozkazy zakończone kropką przy prze-  
-pisywaniu deklaracją /62/ nie ulegają zmianie,  
natomiast do modułów rozkazów zakończonych przecinkiem  
dodaje się licznik tego miejsca rozdziału, począwszy  
od którego przepisuje się program.

Istnieje drugi sposób zapisu rozkazu, w którym  
znak, operacja, bit długości, kropka lub przecinek  
zachowują poprzednią postać i znaczenie.

Natomiast adres może przyjmować następujące formy:

- a/. Adres jest liczbą oktalną od 0 do 7777 oddzieloną  
od operacji i bitu długości spacją / lub kilku  
spacjami/.
- b/. Adres jest symboliczny, w postaci nazwy. W ostatecznym  
efekcie nazwa zostanie przez maszynę zastąpiona liczbą  
oktalną zaczepniętą ze słownika.
- c/. Adres jest symboliczny z indeksem w postaci

[ Nazwa ] [ znak ] [ oktalna ]

Indeks dodatkowo dodaje się do modułu rozkazu

Znak + można opuścić.

- d/. Adres jest względny w postaci

Znak    Oktalna

Znak jest + lub - i nie wolno go opuścić.

W ostatecznej postaci adres równy będzie sumie liczni-ka rozkazu i liczbie oktalnej /z uwzględnieniem znaku/.

Występujący po adresie względny bit długości oddzie-łony musi być co najmniej jedną spacją.

Przy tym drugim sposobie wolno stosować następujące opuszczenia:

a/.można opuścić bit długości = 0

b/.można opuścić bit długości = 0

i adres oktalny = 0. Trzeba zachować co najmniej jedną spację między operacją a kropką lub prze-cinkiem.

c/.można opuścić cały rozkaz = 0, zachowując jedynie kropkę lub przecinek.

§ 23. Deklarowanie miejsc roboczych

Miejsca programu od 7762 do 7777 zajęte są przez program sprowadzający rozkazy z bębna i nie można ich wyko-rzystywać.

Rozdział zapisuje się począwszy od miejsca zerowego pamięci aż do swego końca zaznaczonego deklaracją KONIEC.

Miejsca robocze należy deklarować wolnej części pamięci, najlepiej idąc od jej komórki 7761 wstecz. Np. gdy w rozdziale 6 chcemy zadeklarować 6 miejsc roboczych dla zmiennej A i po 1 dla X i Y piszemy

#### ROZDZIAŁ 6

7761) X) -1) Y) -6) A)

o) **rozkazy**

KONIEC

Jeśli w rozdziale występują dodatkowo rozkazy zawierające adresy symboliczne, dla których brak etykiet, to dla nich w słowniku zostaną zarezerwowane kolejne parzyste adresy bezpośrednio po KONCU rozdziału.

Miejsca robocze podprogramu powinny być w nim zawarte. Deklarujemy je przed końcem podprogramu, pisząc po etykiecie tyle kropek ile rezerwujemy krótkich komórek pamięci. Jeśli np. chcemy w podprogramie XYZ zarejestrować 8 miejsc dla zmiennej A i po 1 miejscu dla X i Y robimy to następująco:

#### PODPROGRAM XYZ

**rozkazy**

X). Y). A).....

KONIEC

Jeśli w podprogramie występują zakończone przecinkiem rozkazy zawierające adresy symboliczne, dla których brak etykiet, to etykiety te dopisują się automatycznie

na kolejnych parzystych miejscach bezpośrednio przed końcem podprogramu, a sam KONIEC się przesuwa.

W podprogramie adresy symboliczne rozkazów zakończonych kropką odnoszą się do nazw zapisanych w słowniku rozdziału, do którego podprogram został sprowadzony deklaracją /9/. Dla tych nazw nie powinno być etykiet w rozpatrywanym podprogramie. Rozkazy te mogą natomiast odnosić się także do etykiet /15/ występujących w innych podprogramach.

#### S 24. Podprogramy biblioteczne

Język SAS zawiera w sobie bibliotekę podprogramów, które są nagrane na bęben wraz z TRANSLATOREM /tzn. programem przetłumaczającym język SAS na język maszyny/. Podprogramy te wywołujemy stosując deklarację /62/. Podamy opis i sposób korzystania z tych podprogramów. Uwaga: Jeśli inaczej nie podajemy, korzystanie z podprogramów nie zmienia licznika cykli.

#### FUNKCJA CZYTAJ

Jeśli w programie zastosujemy wywołanie

22 CZYTAJ 4.

to maszyna przeczyta kolejną informację z taśmy danych.

Informacją tą może być liczba dziesiętna napisana przy użyciu symboli

+ - . 1 0 1 2 3 4 5 6 7 8 9

nie rozdzielonych spacją, zgodnie z zasadami ALGOLU.

(np. liczbę 314 można napisać  $3.14 + 003.14 \cdot 314_{10}^{-2}$

$+31400_{10}^{-4} 0.00314_{10}+3$  itd.),

liczba ta zostanie zinterpretowana jako liczba zmienno-przecinkowa /§2/ i zapamiętana w akumulatorze tak jak przy opisie rozkazu /14/. Po wykonaniu tych czynności maszyna przechodzi do wykonania rozkazu następującego po /71/. (Jeżeli zapis liczby zawierał kropkę pozycyjną to komórka CZYTAJ [40] jest zerem, z czego możemy czasem korzystać dla odróżnienia danych całkowitych od niecałkowitych).

Wczytywaną informacją może być słowo złożone z liter

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Znaki dalekopisowe kolejnych liter wczytywanego słowa

zostaną przepisane do kolejnych bitów akumulatora

/rys. 11/ mantysy

0 1 2 3 4 5 6 7 8 9 10 . . . .

39

I	II	III	IV	V	VI	VII	VIII	VIII
---	----	-----	----	---	----	-----	------	------

KOLEJNE LITERY SŁOWA

Rys. 11

Jeśli słowo przekracza 8 liter, to pozostałe zostaną pominięte. Jeśli tych liter jest mniej, to pierwsze zostaną zapisane na początku, a reszta zastąpiona zerami. Po wczytaniu słowa rozkazem /71/, maszyna opuści jeden rozkaz.

Wczytaną informacją może być także jeden ze znaków pisarskich:

, : ; '/ () [ ] = # \*

oraz + - . o ile nie wchodzą w skład liczby dziesiętnej. Maszyna wczytuje pięć bitów oznaczających ten znak w kodzie dalekopisowym i zapisuje je na bitach 14, 15, 16, 17, 18 akumulatora mantysy, którego pozostałe bity są zerami. Po wczytaniu znaku pisarskiego rozkazem /71/ maszyna opuszcza jeden rozkaz.

Jeśli w programie maszyny napiszemy rozkaz,

22 TYTUL 4.

to z tabulogramu danych zostanie przepisany jeden wiersz na tabulugram wyników.

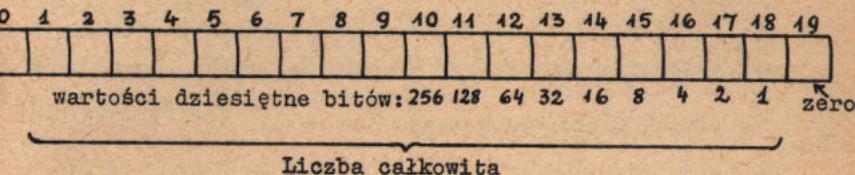
#### FUNKCJA DENORMALIZACJI

Jeśli w akumulatorze znajduje się liczba zmiennoprzecinkowa, to po zastosowaniu wywołanie

22 DENORMALIZACJA 4.

liczba ta zostanie zaokrąglona do najbliższej wartości całkowitej. Ta liczba całkowita zostaje następnie przepisana w postaci stałoprzecinkowej wg. rys. 12,

w pierwszej połowie akumulatora mantysy.



Rys. 12.

#### FUNKCJA DRUKUJ

Do komórki DRUKUJ [2] należy wprowadzić stałoprzecinkową liczbę całkowitą /napisaną jak na rys. 12/ równą ilości żądanego cyfr przed kropką pozycyjną. Do komórki DRUKUJ [3] podobnie wprowadzamy żądaną liczbę cyfr po kropce pozycyjnej. W akumulatorze należy umieścić zmienoprzecinkowo /14/ liczbę, którą każemy maszynie wydrukować pisząc rozkaz

22 DRUKUJ 4.

Jeśli liczba miejsc po przecinku nie jest zerem, to na tabulogramie wyników wydrukuje się:

Znak, część całkowita, kropka, część ułamkowa, przy czym wielkość liczby będzie zaokrąglona do tylu miejsc dziesiętnych, ile ma być wydrukowane. Początkowe zera w części całkowitej zostaną zastąpione spacjami, a znak wydrukuje się tuż przed liczbą. Jeśli część całkowita okaże się większa, niż zarezerwowano dla niej miejsc, to kropka się odpowiednio przesunie.

i zmniejszy się o tyle samo miejsc liczba miejsc po przecinku. Jeśli i to nie wystarczy, liczba zostanie napisana w postaci:

znak, liczba całkowita, 10, potęga, tak jednak, aby cały zapis zajął tyle miejsca na tabulogramie ile przewidziano, to znaczy

**DRUKUJ [2] + DRUKUJ [3] + 2**

Jeśli do DRUKUJ [3] wprowadzimy zero, to nie będzie się drukować kropka pozycyjna, i znak + przed liczbą zastąpiony zostanie spacją. Wydrukowana liczba zajmie miejsc na tabulogramie

**DRUKUJ [2] + 1**

#### FUNKCJA WYDAWNICTWO

W tym przypadku korzysta się z licznika cykli, i jego stan po zakończeniu wykonywania rozkazów podprogramu jest zerem.

Wywołanie

22 BLANKI 4.

Powoduje wydrukowanie tylu znaków pustej taśmy, ile wynosi przed wywołaniem zawartość licznika cykli.

Rozkaz

22 SPACJA 4.

powoduje wydrukowanie tylu spacji, ile wynosi początkowa zawartość licznika cykli.

Rozkaz

22 LINIA 4.

/80/

powoduje wydrukowanie jednego znaku karetki, a po nim  
tylko znaków linii, ile wynosi początkowa zawartość  
licznika cykli.

FUNKCJA TRYGONOMETRYCZNA

/81/

Do akumulatora należy wprowadzić zmiennoprzecinkowo /14/ argument funkcji. Po wykonaniu niżej podanych rozkazów otrzymamy w akumulatorze odpowiednią wartość zmiennoprzecinkową. Przewiduje się następujące rozkazy:

22 SIN 4.

/82/

22 COS 4.

/83/

22 TG 4.

/84/

22 CTG 4.

/85/

FUNKCJA ODWROTNIA

/86/

Działanie analogiczne jak przy /81/. Przewiduje się możliwość korzystania z następujących rozkazów:

22 ASIN 4. dla funkcji arc sin /87/

22 ACOS 4. " " arc cos /88/

22 ATG 4. " " arc tg /89/

22 ACTG 4. " " arc ctg /90/

22 PWK 4. dla pierwiastka kwadratowego /91/

Błędne wywołanie /np. pierwiastek liczby ujemnej/  
powoduje zatrzymanie maszyny i wydrukowanie skrótowej  
informacji o błędzie. Po naciśnięciu guzika START, maszy-  
na będzie pracowała dalej.

FUNKCJA WYKŁADNICZA

Działanie /łącznie z sygnalizacją błędów/ jak wyżej.

Mögliwe rozkazy:

22 LN 4. dla logarytmu naturalnego /93

22 EXP 4. dla funkcji  $e^x$  /94

FUNKCJA URAL

Podprogram służy do rozwiązywania układu równań  
algebraicznych liniowych /metodą rugowania z wyborem  
największego elementu/.

Począwszy od zakończonego adresu pamięci operacyjnej  
należy zapisać jedną po drugiej kolumny współczynników  
przy kolejnych niewiadomych. Za nimi bezpośrednio  
należy zapisać kolumnę wyrazów wolnych. Jeśli to samo  
zadanie rozwiążujemy dla kilku kolumn wyrazów wolnych,  
to następne kolumny, piszemy także bezpośrednio  
jedne po drugiej.

Wszystkie występujące liczby zapisywane są jako  
zmiennoprzecinkowe.

Do komórki URAL [2] wprowadzamy adres zerowego  
wyrazu zerowej kolumny / o której była mowa wyżej/

Do komórki URAL [3] wprowadzamy liczbę równań.

Jest to liczba całkowita w postaci z rys.12.

Do komórki URAL [4] wprowadzamy analogicznie liczbę kolumn wyrazów wolnych.

Po tym przygotowaniu, wykonanie rozkazu

22 URAL 4

/96/

spowoduje zastąpienie wartości wpisanych w kolumny wyrazów wolnych przez odpowiednie wartości niewiadomych.

Macierz współczynników ulegnie zniszczeniu.

#### FUNKCJA TEKST

/97/

Jeśli chcemy aby został wydrukowany określony przez nas tekst, to piszemy w programie

TEKST # dowolny tekst

nie zawierający

znaku # #

/98/

CZĘŚĆ IV

KORZYSTANIE Z JĘZYKA SAS

C Z E S C IV

KORZYSTANIE Z JĘZYKA SAS

§ 25. Wczytywanie programów.

Przed wczytaniem programu należy:

- a/. odblokować bębnny
- b/. odblokować oba wskaźniki blokady 4
- c/. odblokować wszystkie klucze
- d/. nagrać SAW
- e/. wczytać taśmę autokodu SAS

Taśma ta wczytuje się bezgłośnie, etapami, które po sprawdzeniu sumy kontrolnej zapisują się na bębnie. Jeśli suma kontrolna się nie zgadza trzeba wczytywanie zacząć od początku.

Taśma zatrzymuje się na odcinku pustej taśmy, po którym następuje krótki odcinek, którego nie należy narazie wczytywać.

f/. Dalej należy podłożyć do wczytywania program w języku SAS i nacisnąć guzik START.

Kolejno napisane podprogramy i rozdziały zapisuje się według reguł podanych w części III. Wczytywanie kończy się po deklaracji /58/. Maszyna wydrukowuje krótką informację o wolnych miejscach na bębnie oraz kawałek pustej taśmy, po czym staje.

g/. Należy podłożyć taśmę z danymi i nacisnąć guzik "START", wtedy rozpocznie się liczenie.

Jeśli w trakcie tłumaczenia maszyna natrafi na błędy formalne, to wydrukuję wiersz zawierający ten błąd oraz ewentualną uwagę o rodzaju błędu. Po wczytaniu błędного podprogramu lub rozdziału maszyna zatrzymuje się. Uruchomienie następuje przez dalsze naciśnięcie guzika START. Błędne podprogramy i rozdziały nie są zapisywane na bębnie.

Jeśli przy tłumaczeniu włączymy klucz 1, to maszyna wydrukuję słownik symboli rozdziału lub podprogramu. Jeśli włączymy klucz 2 to otrzymamy

podprogram lub rozdział przetłumaczony oktalicie. To ostatnie zaleca się stosować do wypróbowanych programów, gdyż taśma tak otrzymana zawiera sumy kontrolne a przetłumaczone rozdziały zawierają już dopisane podprogramy /62/. Taśma ta może być wczytywana zamiast pierwszej, oryginalnej.

Jeśli chcemy skorzystać z nagranego autokodu, bez ponownego nagrywania, należy wyzerować akumulator i zastartować od miejsca 7762. Czasem może to zawiść. Wtedy nagrywamy SAW i wczytujemy koniec taśmy autokodu, o którym była mowa w punkcie f.

#### § 26. Wzbogacanie biblioteki podprogramów

Jeśli chcemy do biblioteki autokodu na stałe dołączyć nowy podprogram, to postępujemy następująco:

- a/. Wykonujemy czynności a,b,c,d,e poprzedniego paragrafu
- b/. Podkładamy do czytnika taśmę z nowymi podprogramami.

Po ostatnim z nich powinien występować w osobnym wierszu średnik i karetka. Spowoduje to zatrzymanie taśmy po wczytaniu.

c/. Podkładamy do czytnika taśmę z programem:

WYPROWADZENIE AUTOKODU i naciskamy guzik START.

- d/. Po wczytaniu tej taśmy, za każdym naciśnięciem guzika START maszyna wyprowadzi nową taśmę autokodu.

CZĘŚĆ V

PRZYKŁADY

PRZYKŁADY

Rodzaj działania	Zapis w SAS	Na końcu programu muszą być zapisane rozkazy
0	1	2
Przepisz jeden wiersz danych na tabulogram wyników	22 TYTUL 4.	FUNKCJA CZYTAJ
Wczytaj dwie kolejne liczby z tabulogramu danych i zapisz do komórek A,B	22 CZYTAJ 4. 56 A 4, 22 CZYTAJ 4. 56 B 4,	FUNKCJA CZYTAJ
Wczytaj kolejną liczbę tabulogramu danych pomijając teksty komentarza na tym tabulogramie i zapisz ją do komórki A	22 CZYTAJ 4. 22+2, 22-1, 56 A 4,	FUNKCJA CZYTAJ
Wczytaj kolejną liczbę tabulogramu danych potraktuj ją jako całkowitą i zapisz do komórki N	22 CZYTAJ 4. 22 DENORM 4. 16 N,	FUNKCJA CZYTAJ
Wczytaj kolejną informację z tabulogramu danych. Jeśli to jest liczba, to skocz do etykiety LICZBA), jeśli to jest średnik to skocz do etykiety SREDNIK). Inne informacje pomin.	22 CZYTAJ 4. 22 LICZBA, 14 SR, 22 SREDNIK 4, 22-4, LICZBA)  SREDNIK)  SR) 00 0032 0.	FUNKCJA CZYTAJ

0	1	2
<p>Wykonaj następujące działania, w których A,B,C,D,X są zmienno przecinkowymi I,J,K,L,M,N, są całkowitymi:</p> <p>A = B + C  A = B - C  A = B * C  A = B / C</p>	<p>42B4, 41C4, 56A4,  42B4, 43C4, 56A4,  42B4, 46C4, 56A4,  42B4, 47C4, 56A4,</p>	
<p>I = J + K  I = J - K  I = J * K  I = J / K</p>	<p>02J, 01K, 16I,  02J, 03K, 16I,  02J, 06K, 11 22 4, 16I,  02J, 11 122 4, 07K, 16I,</p>	
<p>A [4] = B[2+I] * C</p>	<p>30I, 42 B[2]4, 46C4,  56A[4]4,</p>	
<p>A = B[I * 4 + K]</p>	<p>02I, 06 CZTERY  11 22 4, 01 K, 16 N,  30 N, 42B4, 56A4,</p>	<p>CZTERY) (4)</p>
<p>A = <math>\sqrt{1 - (B \cdot \sin D)^2}</math></p>	<p>42D4, 22SIN4,  46B4, 56C4,  46C4, 56C4,  42JEDEN4, 43C4,  22PWK4, 56A4,</p>	<p>**) JEDEN) (1.0)  FUNKCJA TRYGON  FUNKCJA ODWROTNIA</p>
<p>A = B<sup>C</sup></p>	<p>42B4, 22LN4,  46C4, 22EXP4,  56A4,</p>	<p>FUNKCJA WYKŁAD</p>

0	1	2
A=ARC SIN (B <sup>2</sup> +C <sup>2</sup> )	42B4, 46B4, 56D4, 42C4, 46C4, 41D4, 22 ASIN 4, 56A4,	FUNKCJA ODWROTNÄ
A = I	02I, 46NORMA4, 56A4,	#) NORMA) 40 0.0 11 4.
I = A	42A4, 22DENORM4, 16I,	FUNKCJA DENORM
A = B+3*I	02I, 46NORMA4, 46TRZY4, 41B4, 56A4	#) NORMA) 40 0.0 11 4. TRZY) (3.0)
Powtórz następującą sekwencję rozkazów dla I=J, J+K, J+2K, ..., L.	02 J, POWT) 16K, <div style="border: 1px solid black; padding: 5px; display: inline-block;">Rozkazy powtarzane</div> 02 J, 14 L, 21+4, 02 J, 01 K, 22POWT,	
Powtórz następującą sekwencję rozkazów dla A=B, B+C, B+2C, ..., D, (Ostatnie powtóżenie następuje dla  D-A  <  Z  po czym następuje A = D)	42B4, POWT) 56A4, <div style="border: 1px solid black; padding: 5px; display: inline-block;">Rozkazy powtarzane</div> 42D4, 43A4, 46DWA4, 44B4, 21+4, 42A4, 41B4, 22POWT, 42D4, 56A4,	#) DWA) (2.0)

0	1	2
Zapisz na bęben od adresu bębnowego 100 zawartości komórek J[0], J[1],...,J[6],	'TWORZENIE NEGACJI SUMY KONTROLNEJ W I[7]' 25 6. 02 ZERO, -26J[6], 24-1, 11 24. 14 7777. 16J[7]; 'ZAPIS NA BEBEN' 52J[0] ,00 100. 00 J[7], 'CZYTANIE KONTROLNE' 51J[0] 4, 00 100. 00J[7], 11 24. 14 7777.21-10,	Uwaga: w komórce 7777 znajduje się zawsze liczba -77 7777 4. ZERO) (0)
Przepisz wyżej zapisaną informację z bębna do komórek J[0], J[1],...,J[6].	'PRZEPISYWANIE' 51 J[0] , 00 100. 00 J[7], 'SPRAWDZANIE SUMY KONTROLNEJ' 11 24. 14 7777.21-5,	
Wydrukuj odpowiedź w postaci  A = <u>wartość liczbowa</u>  B = <u>wartość liczbowa</u>  przy czym wartość liczbowa ma zawierać 3 znaki dziesiętne przed kropką pozycyjną i dwa po kropce	TEKST # A =# 02TRZY, 16DRUKUJ [2]. 02DWA, 16DRUKUJ [3]. 42A4, 22DRUKUJ 4.  TEKST # B =# 42B4, 22 DRUKUJ 4,	FUNKCJA TEKST FUNKCJA DRUKUJ TRZY) (3) DWA) (2)
Wydrukuj liczbową wartość zmiennej całkowitej I, przewidując najwyżej 5 znaków dziesiętnych.	02PIEC, 16DRUKUJ [2]. 02ZERO, 16drukuj [3]. 02I, 46NORMA4, 22DRUKUJ4.	#)NORMA) 40 0,0 11 4. FUNKCJA DRUKUJ PIEC) (5) ZERO) (0)
Wydrukuj 5 linii	25 5.22 LINIA 4.	FUNKCJA WYDAWNICTWO
Wydrukuj 64 rządów pustej taśmy	25 100.22BLANKI 4.	FUNKCJA WYDAWNICTWO

0	1	2
Wydrukuj 3 spacje	25 3.22 SPACJA 4.	FUNKCJA WYDAW
Rozwiąż układ równań $A_0X_0 + A_6X_2 + A_{14}X_4 = A_{22}$ $A_2X_0 + A_{10}X_2 + A_{16}X_4 = A_{24}$ $A_4X_0 + A_{12}X_2 + A_{20}X_4 = A_{26}$ (Uwaga: indeksy są zapisane oktalicie i zwiększały się do dwa, bo każde dwie komórki zawierają jedną liczbę zmennoprzecinkową)	00 A,02-1,16URAL[2]. 02TRZY,16URAL[3], 02RAZ,16URAL[4], 22URAL4. 42A[22] 4,56X [0] 4, 42A[24] 4,56X [2] 4, 42A[26] 4,56X [4] 4,	TRZY)(3) RAZ)(1) FUNKCJA URA
Skocz do miejsca oznaczonego etykietą E	22 E ,	
Jeśli $A > B$ skocz do E inaczej, wykonaj rozkaz następnego	42A4,43B4,21E4,	
Jeśli $A < B$ skocz do E, inaczej do rozkazu następnego	42A4,43B4,21E,	
Jeśli $A > B$ skocz do E, inaczej do rozkazu następnego	42B4,43A4,21 E ,	
Jeśli $A \leq B$ skocz do E, inaczej do rozkazu następnego	42B4,43A4,21E4,	
Jeśli J=K, skocz do E, inaczej do rozkazu następnego	02 J,14 K, 21 E 4,	
STOP	37 O.	

10-000599-0000



A-599

Wojskowej Akademii Technicznej

Biblioteka Główna