

Podprogramy zamknięte wprowadzamy kolejno jeden za drugim (kolejność dowolna). Kolejne rozkazy programu ustawiającego, łącznie z rozkazami taśmowymi podane są w tabl. 5-11.

Ponadto program ustawiający umożliwia programiście autokontrolę, czy przypadkiem rozkazy programu głównego nie są wprowadzane na miejsca, na które poprzednio wprowadziliśmy podprogramy. W tym celu na końcu programu głównego należy umieścić grupę rozkazów taśmowych:

- 1) S4 g ,
- 2) 14 $g-1$,
- 3) M2 0141,

gdzie g oznacza najwyższy adres zajęty przez program główny. W przypadku pomyłki w podziale pamięci program główny nie zostanie wprowadzony do maszyny.

5.2. PROGRAMY WYPROWADZAJĄCE

Wyprowadzanie wyników obliczeń oraz programów zapisanych w pamięci dokonujemy za pomocą programów wyprowadzających. Przy normalnej eksploatacji maszyny musimy dysponować co najmniej czterema programami wyprowadzającymi:

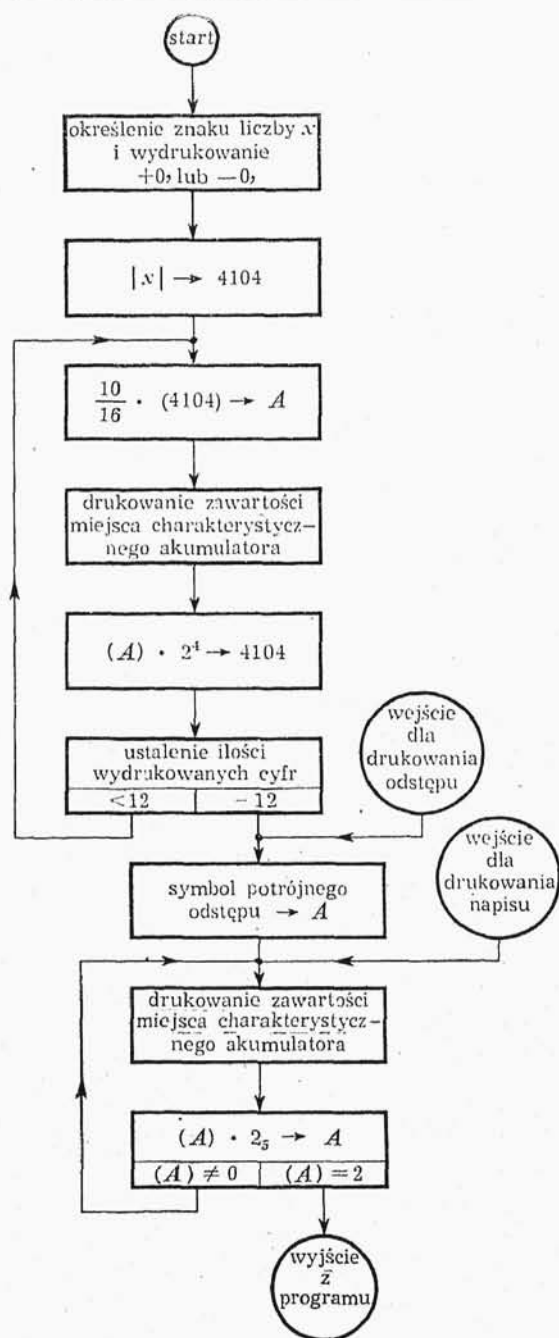
- 1) programem do wyprowadzania zawartości kolejnych miejsc pamięci w postaci rozkazów i pseudorozkazów,
- 2) programem wyprowadzająco-przeliczającym dla liczb całkowitych,
- 3) stałoprzecinkowym programem wyprowadzająco-przeliczającym,
- 4) zmiennoprzecinkowym programem wyprowadzająco-przeliczającym.

W odróżnieniu od programów wprowadzających, programy wyprowadzające mają prostą strukturę. Ograniczymy się do omówienia tylko dwóch programów wyprowadzających, a mianowicie stałoprzecinkowego programu wyprowadzająco-przeliczającego (bardzo uproszczonego) i programu dla wyprowadzania zawartości kolejnych miejsc pamięci w postaci rozkazów i pseudorozkazów.

Stałoprzecinkowy program wyprowadzająco-przeliczający łącznie z podstawowym programem wprowadzającym i parametrami jest nagrany na stałe na ścieżce nr zero. Program ten służy do drukowania liczb stałoprzecinkowych co do modułu mniejszych od jedności w układzie dziesiętnym, ponadto służy do drukowania odstępów potrojnych i napisów trzyznakowych.

Stałoprzecinkowy program wyprowadzająco-przeliczający zastosowany do liczb x ($|x| < 1$), drukuje symbol $+0$, lub -0 , w zależności od znaku liczby x oraz 12 cyfr dziesiętnych liczby x . Cyfry te uzyskuje się przez wielokrotne mnożenie x przez $2^{-4} \cdot 10$, drukowanie cyfry, która znalazła się na miejscu charakterystycznym akumulatora oraz arytmetyczne przesuwanie wyniku o 4 miejsca w lewo (czyli mnożenie wyniku przez 2^4). Po wydrukowaniu liczby program drukuje potrójny odstęp. Odstęp taki można również drukować bez drukowania jakiegokolwiek liczby.

Uwaga: Napis -0 , tworzymy przez dodanie do napisu $+0$, liczby ósemkowej



Rys. 5-4. Schemat blokowy stałoprzecinkowego programu wprowadzająco-przeliczającego

01 0000, gdyż minus w kodzie wewnętrznym ma postać ósemkową 13 0000, zaś plus ma postać 12 0000.

Wywołanie stałoprzecinkowego podprogramu wyprowadzająco-przeliczającego dla przeliczania i drukowania liczby x (gdzie $|x| < 1$) wygląda następująco:

$a+0000$	14	4104,
1	12	4070,
2	14	4102,
3	04	0101.

Uwaga: Zakładamy tu, że liczba przeznaczona do drukowania znajduje się w akumulatorze; w przypadku gdy liczba ta znajduje się pod jakimś adresem różnym od 4104, do wywołania musimy dołączyć jeszcze jeden rozkaz. W przypadku, gdy liczba przeznaczona do drukowania znajduje się już pod adresem 4104, wywołanie skraca się o jeden rozkaz.

Parametr występujący w wywołaniu podprogramu dla drukowania liczb ósemkowych składa się z dwóch słów krótkich, pierwsze z nich (o adresie krótkim parzystym) określa *wejście* do programu, drugie (o adresie krótkim nieparzystym) jest skalą logiczną dla pętli drukującej.

Wywołanie podprogramu dla drukowania odstępu wygląda następująco:

$d+0000$	12	0072
1	14	0102
2	04	0101

Wywołanie podprogramu dla drukowania dowolnego napisu ma postać:

$e+0000$	12	0073
1	14	0102
2	12	< napis >
3	04	0101

W przypadku gdy napis jest postaci „+0,“ wywołanie wygląda następująco:

$e+0000$	12	0073
1	14	0102
2	12	0064
3	04	0101

START

0,25 EXP	+0,000000000000	=	+0,250000000000
0,25 EXP	+0,001999999978	=	+0,250500500318
0,25 EXP	+0,003999999957	=	+0,251002002623
0,25 EXP	+0,005999999935	=	+0,251504508894

STOP

Rys. 5-5. Przykład drukowania wyników i napisów za pomocą stałoprzecinkowego programu przeliczająco-wyprowadzającego

Tablica 5-12

Stałooprzecinkowy program wyprowadzająco-przeliczający

Kolejny adres	Kolejny rozkaz		Uwagi
0032	12	4104	początek programu drukowania liczb
3	20	0041	
4	22	0004	
5	10	0064	
6	25	0000	
7	23	0006	
0040	30	0074	
1	03	0036	
2	12	4104	
3	26	0000	
4	14	4104	
5	27	0065	
6	16	4104	
7	25	0000	
0050	20	0004	
1	14	4104	
2	12	0103	
3	23	0001	
4	14	0103	
5	03	0046	
6	12	0004	początek podprogramu drukowania odstępu
7	25	0000	początek podprogramu drukowania napisu
0060	23	0006	(¹)
1	31	0074	
2	03	0057	
3	02	0101	
4	12	0025	+ 0, (²)
5	12	0000	
6	01	0000	
7	00	0002	
0070	02	0032	} parametr dla drukowania liczb parametr dla drukowania odstępu parametr dla drukowania napisu parametr
1	17	7740	
2	02	0056	
3	02	0057	
4	37	7777	

(¹) Pod adresem 0004 znajduje się parametr 24 2424 (patrz tabl. 5-6) będący potrójnym symbolem odstępu (patrz tabl. 2-6).(²) Patrz tabl. 2-6

Na rysunku 5-4 podany jest schemat blokowy stałooprzecinkowego programu wyprowadzająco-przeliczającego.

Na rysunku 5-5 pokazana jest postać wyników obliczeń, wyprowadzonych na zewnątrz za pomocą omawianego wyżej podprogramu.

Podamy jeszcze kilka danych dotyczących tego programu — długość 38 słów krótkich, liczba wykonanych rozkazów 130 dla liczby, 13 dla trzyznakowego napisu. Kolejne rozkazy naszego programu podano w tabl. 5-12.

Tablica 5-13

Program drukowania kolejnych miejsc pamięci w kodzie rozkazowym

Kolejny adres	Kolejny rozkaz		Objaśnienia			
$k + 0000$	12	<16 0000>	nastawienie dalekopisu na drukowanie cyfr i znaków			
1	25	0000				
2	12	<22 0000>	wykonanie dodatkowego „wysuwu“			
3	25	0000				
4	12	<21 0000>	ustawienie nowego wiersza			
5	25	0000				
6	12	<22 0000>				
7	25	0000				
$k + 0010$	12	0120	drukowanie adresu			
1	10	$k + 4054$				
2	04	$k + 0043$				
3	12	<24 0000>	drukowanie podwójnego odstępu			
4	25	0000				
5	25	0000				
6	00	0120	drukowanie części operacyjnej			
7	30	<37 0000>				
$k + 0020$	22	0006				
1	10	<00 0040>				
2	04	$k + 0043$				
3	12	<24 0000>	drukowanie pojedynczego odstępu			
4	25	0000				
5	00	0120	drukowanie części adresowej			
6	10	$k + 4054$				
7	04	$k + 0043$				
$k + 0030$	12	0120	przeadresowanie			
1	10	0076				
2	14	0120				
3	11	0121				
4	03	$k + 0036$				
5	05	0120	stop			
6	12	0120	sprawdzenie końca „bloczka“ złożonego z ośmiu słów			
7	11	<00 0007>				
$k + 0040$	30	<00 0007>				
1	03	$k + 0004$				
2	02	$k + 0002$	skok do drukowania dodatkowego „wysuwu“			
3	77	7777	podprogram			
4	30	$k + 4052$				
5	06	$k + 0051$				
6	23	0003				
7	25	0005				
$k + 0050$	02	$k + 0044$				
1	01	$k + 0043$				
2	20	0000	5 zer 0...0	13 jedynek 1...1	16 zer 0...0	parametr
3	00	7777	17 zer 0...0	16 zer 0...0	1	parametr
4	20	0000				
5	00	0000				

W tablicy 5-13 podany jest program przeznaczony do drukowania zawartości kolejnych krótkich miejsc pamięci począwszy od n -tego, a skończywszy na m -tym (gdzie $n \leq m$), w postaci rozkazów i pseudorozkazów. Program ten wywołujemy w następujący sposób:

- 1) umieszczamy pod adresami 0120 i 0121 parametry

$$\begin{array}{rcl} 0120 & 12 & n, \\ & 1 & 12 \quad m+1, \end{array}$$

gdzie n i m pierwszy i ostatni adres krótkich słów przeznaczonych do drukowania;

- 2) wykonujemy rozkaz

$$02 \quad k+0000.$$

Program podany w tabl. 5-13, drukuje kolejne rozkazy i pseudorozkazy, począwszy od zawartości adresu n , a skończywszy na zawartości adresu m .

Tablica 5-14

Parametry rozkazowe

Adres parametru	Postać parametru w kodzie rozkazowym	
0066	01	0000
0021	02	0000
0065	12	0000
0011	23	0000
0007	24	0000
0036	25	0000
0043	26	0000

Tablica 5-15

Parametry arytmetyczne

Adres parametru	Postać parametru w kodzie rozkazowym		Wartość arytmetyczna parametru
0077	00	0000	0 (zero krótkie)
0076	00	0001	2^{-16}
0074	37	7777	-2^{-16}
0067	00	0002	2^{-15}
0075	17	7777	$1-2^{-16}$
4076	—	—	2^{-33}
4074	—	—	$1-2^{-33}$
0066	01	0000	2^{-4}
0021	02	0000	2^{-3}
0065	12	0000	$10 \cdot 2^{-4}$

W tablicach 5.14, 5.15 podane są parametry zapisane na stałe na ścieżce zerowej; z parametrów tych będziemy wielokrotnie dalej korzystali.

5.3. URUCHOMIENIE MASZINY

W punkcie tym nie będziemy zajmowali się bliżej stroną techniczną uruchomienia, sprawy te omawia szczegółowo instrukcja techniczna obsługi danej maszyny. Natomiast zakładając, że maszyna jest włączona do sieci oraz że wszystkie napięcia przyjmują prawidłowo wartości, omówimy czynności wykonywane przez matematyka dla rozpoczęcia obliczeń.

1. Za pomocą przycisku W15 gasimy neonówkę S2 (zegar), w przypadku gdy neonówka S2 nie zgasła lub zapaliła się ponownie, nie można prowadzić obliczeń, gdyż nastąpiło uszkodzenie maszyny (patrz Załącznik 2, pulpit sterowania).

2. Zerujemy wszystkie rejestry maszyny przez przyciśnięcie przycisku W4 (patrz jak wyżej).

3. Ustawiamy przełącznik W3 (I-II takt pracy) w położeniu górnym „I takt pracy” (patrz jak wyżej).

4. Ustawiamy przełącznik W1 (praca ciągła — praca krokowa) w położeniu górnym „praca ciągła” (patrz jak wyżej).

5. Zakładamy taśmę perforowaną z testem lub programem, który chcemy wykonać, pod głowicę czytnika, zamykamy głowicę czytnika, włączamy oświetlenie czytnika za pomocą przełącznika znajdującego się pod czytnikiem. Wówczas czytnik samoczynnie przesuwa taśmę zatrzymując się na rzędku pierwszym przeznaczonym do czytania.

6. Przyciskamy przycisk W2 (start) (patrz jak wyżej). Wówczas maszyna zaczyna wykonywać rozkazy umieszczone pod kolejnymi adresami począwszy od adresu 0000. Jak już wiemy, pod adresem 0000 znajduje się początek programu wprowadzającego, który spowoduje odczytanie grupy rozkazów taśmowych początku czytania (punkt 5.1.1, podstawowy program wprowadzający), perforowanych na taśmie, a następnie zacznie wprowadzać kolejne rozkazy i pseudorozkazy programu. Jak mówiliśmy już w punktach 5.1.2 i 5.1.3, wywołania programów wprowadzająco-przeliczających dokonujemy za pomocą rozkazów taśmowych, oczywiście po uprzednim wprowadzeniu tych programów do maszyny.

Uwaga 1: W przypadku gdy chcemy uruchomić maszynę, która zatrzymała się samoczynnie po wykonaniu pewnego etapu obliczeń, w zależności od budowy wykonywanego programu nastawimy przełącznik W3 w którymś z położen: „I takt pracy” lub „II takt pracy” i naciskamy przycisk „start”.

Uwaga 2: W przypadku gdy w czasie obliczeń zapali się neonówka S2 (zegar), należy zatrzymać maszynę i sprawdzić, czy neonówka S2 daje się zgasić przy użyciu przycisku W15. Gdy neonówka nie daje się zgasić, należy przystąpić do kontroli technicznej, w przypadku zaś gdy neonówka S2 daje się zgasić, należy sprawdzić maszynę testem (programem kontrolnym), po czym należy powtórzyć obliczenia.

5.4. SZUKANIE BŁĘDÓW W PROGRAMACH

Po wprowadzeniu programu do maszyny należy przeprowadzić szczegółową kontrolę programu, następnie zaś wykonać próbne obliczenia np. dla takich wartości argumentów, dla których znamy wyniki obliczeń. W przypadku gdy próbne obliczenia wypadną dobrze, przystępujemy do właściwych obliczeń. Sprawą próbných obliczeń nie będziemy się zajmowali bliżej ze względu na trudność spowodowaną wielką różnorodnością metod numerycznych stosowanych w obliczeniach automatycznych. Natomiast sprawę formalnej kontroli programu, to co Rosjanie nazywają „otładka programy“, omówimy dosyć dokładnie.

Zakładając, że schemat blokowy programu jest ułożony prawidłowo, kontrolujemy, czy nie powstały błędy w czasie kodowania, dziurkowania lub wprowadzania programu do maszyny. Najprostszą metodę takiej kontroli zastosowano przy eksploatacji angielskiej maszyny cyfrowej EDSAC, polega ona na drukowaniu części operacyjnej kolejno wykonywanych rozkazów programu, przy czym kontrola ta była jednocześnie próbnym obliczeniem. W przypadku wykonywania rozkazu skokowego program kontroli działania uruchamianego programu drukował wielokrotny odstęp między kodami operacji. Metodę tę można zmodyfikować o tyle, że w przypadku gdy program główny wywołuje podprogram, program kontroli działania uruchamianego programu sygnalizuje ten fakt, po czym nie kontrolując podprogramu (zakładając, że jest on poprawny) dalej kontroluje program główny. Program dla tego typu kontroli programów jest prostym programem interpretacyjnym (punkt 6.1), pobierającym kolejne rozkazy programu kontrolowanego do wykonania z drukowaniem w części operacyjnej wykonywanego rozkazu w przypadku, gdy jest to rozkaz skokowy; program interpretacyjny poza kodem operacji drukuje wielokrotny odstęp, w przypadku zaś wywołania podprogramu drukuje również część adresową rozkazu wywołującego, po czym przechodzi do następnego rozkazu programu głównego. Omówiony powyżej program kontroli jest prosty i dosyć uniwersalny, wymaga natomiast dużo czasu dla samej kontroli i starannego przygotowania próbných obliczeń, łącznie z obliczeniem wartości funkcji obliczanych przez podprogramy.

Poza omówioną wyżej metodą, bywają używane metody dające dużo dokładniejszą kontrolę, metody te mają jednak tę wadę, że wymagają dużego nakładu pracy i trwają stosunkowo długo, znacznie dłużej niż metoda stosowana na maszynie EDSAC. Ogólna charakterystyka tych metod jest następująca:

- 1) bardziej uniwersalna, polegająca na układaniu przez program kontrolowany schematu blokowego kontrolowanego programu,
- 2) indywidualna, polegająca na kontroli programu i porównywaniu go z zadanyim schematem blokowym.

W praktyce najwygodniej jest po prostu uruchamiać program przez wykonywanie próbných obliczeń, a następnie wyprowadzenie za pomocą programu podanego w tabl. 5-13 program z maszyny (lub części programu, zawierające liczniki cykli i ślady). Następnie kontrolujemy w wyprowadzonym programie stany liczników, na podstawie których szukamy błędów.