# 5. PROGRAMOWANIE

## 51. Pojęcia podstawowe

Maszyna cyfrowa wykonuje obliczenia na podstawie pewnego ogólnego planu działania. Plan ten obejmuje czynności przygotowawcze, jak np. ustalenie metody obliczenia, ustalenie przyrostów zmiennych niezależnych, obranie rozwinięcia obliczanej funkcji na szereg oraz szczegółowy wykaz czynności, które mają być wykonane przez maszynę. Ta ostatnia część planu nazywa się programem danego problematu rachunkowego.

Program jest ciągiem, czyli sekwencją określonych operacji. Większość operacji jest tego rodzaju, że każda z nich obejmuje bądź wykonanie działania arytmetycznego, bądź przeniesienie liczby z jednego organu maszyny do drugiego - np. z arytmometru do pamięci - bądź też zarówno działanie arytmetyczne, jak i przeniesienie liczby. Kilka zaledwie operacji - np. zatrzymanie maszyny - nie zawiera żadnej operacji wymienionych dwóch rodzajów.

Liczba operacji używanych do budowy najbardziej nawet skomplikowanych programów jest nieznaczna. Zawiera się ona w różnych maszynach przeważnie w granicach od kilkunastu do trzydziestu paru. Z tego wynika, że każdą operację można oznaczyć za pomocą dwucyfrowej liczby w układzie dziesiętnym lub pięciocyfrowej w układzie dwójkowym. Jest rzeczą dogodną przyporządkowywać poza tym poszczególnym operacjom również symbole literowe; w większości przypadków mogą to być symbole jednoliterowe.

Prawie wszystkie operacje dotyczą liczb. Na przykład operacja: "Dodać do zawartości sumatora liczbę zawartą w miejscu pamięciowym numer x" dotyczy jakiejś liczby umieszczonej w pamięci, bądź zawczasu, bądź w poprzedniej fazie obliczeń. Dlatego też do symbolu operacji dołączamy symbol miejsca

pamięciowego, otrzymując w ten sposób symbol czegoś, co ma charakter bardziej zakończony i konkretny i co nazywamy zazwyczaj rozkazem.

Symbol miejsca pamięciowego nazywamy adresem liczby w miejscu tym zawartej. Symbol rozkazu składa się zatem z dwóch części, z których pierwsza oznacza operację, druga zaś - adres. Dotyczy to typu rozkazów, które będziemy nazywali jednoadresowymi lub krócej rozkazami prostymi. W dalszym ciągu niniejszego pod słowem " rozkaz" bez bliższych omówień będziemy rozumieli rozkaz prosty.

Jeśli operację dodawania oznaczymy literą "A", to symbol "A 099" oznacza rozkaz dodania do zawartości sumatora liczby zawartej w miejscu pamięciowym 099. Przypuśćmy, że operacji dodawania przyporządkowaliśmy liczbę dziesiętną "01". Wówczas ten sam rozkaz możemy napisać w postaci

01 099
operacja adres

Ponieważ liczby miejsc znakowych, liczby symboli operacji i symboli adresów są stałe, wiemy zawsze, które cyfry w symbolu rozkazu dotyczą operacji, które zaś - adresu.

W układzie dwójkowym ten sam rozkaz ma postać

0000 1000 1100 011
operacja adres

Ponieważ zarówno symbol operacji, jak i symbol adresu, mają postać liczb, to i symbol rozkazu ma też postać liczby. Wynika z tego, że rozkaz może być traktowany w maszynie pod każdym względem, jak liczba, a m.in. to, że może on być lokowany w rejestrach pamięci. Jest to szczegół niezmiernie ważny. Czas upływający między zakończeniem jednej operacji a początkiem następnej powinien być możliwie mały, ponieważ jest to czas martwy. Wielkość jego zależy w znacznym stopniu od czasu otrzymania następnego rozkazu przez organ sterujący maszyny. Czas wykonania jednej operacji w maszynie o dużej szybkości jest rzędu co najwyżej paru milisekund, a w niektórych maszynach dla większości operacji czas ten jest kilkadziesiąt

mikrosekund. Czas otrzymania następnego rozkazu powinien być znacznie mniejszy od czasu wykonywania operacji, co może być osiągnięte tylko w ten sposób, że rozkazy będą lokowane w pamięci o małym czasie oczekiwania.

Rozkazy są lokowane w pamięci w takiej kolejności, w jakiej - w zasadzie - mają być wykonywane. Ograniczenie reguły kolejnego wykonywania rozkazów ma charakter następujący. Program można podzielić na części, które będziemy nazywali ciągami rozkazów, w taki sposób, że rozkazy należące do tego samego ciągu są wykonywane zawsze w tej kolejności, w jakie zostały one zarejestrowane w pamięci, kolejność natomiast wykonywania ciągów zależy nie tylko od programu, ale również od uzyskanych uprzednio wyników obliczeń. Niektóre z ciągów mogą być - i częstokroć są - przebiegane wielokrotnie, inne natomiast mogą być pomijane.

Program, który by nie zawierał przebieganego wielokrotnie ciągu rozkazów, nie ma żadnego znaczenia praktycznego; najłatwiej jest przekonać się o tym na podstawie przykładu.

Rozpatrzmy zadanie obliczenia sumy trzech liczb. Oznaczmy potrzebne nam operacje w sposób następujący.

A - dodawanie liczby z określonego miejsca pamięciowego do zawartości sumatora.

C - przeniesienie zawartości sumatora do określonego miejsca pamięciowego i ustawienie sumatora na zero

W - wydrukowanie wyniku zarejestrowanego w określonym miejscu pamięciowym

Z - zatrzymanie maszyny.

Przypuśćmy, że użyta maszyna ma pamięć o pojemności tysiąca liczb, wobec czego adres w każdym rozkazie jest trzycyfrowy. Oznaczmy przez $a_1$, $a_2$, $a_3$ liczby, które mają być dodane, i załóżmy, że są one umieszczone w pamięci w sposób niżej podany:

| | |
|---|---|
| w miejscu pamięciowym numer | 008 - $a_1$ |
| " " " | 009 - $a_2$ |
| " " " | 010 - $a_3$. |

Przy tych założeniach program dodania tych liczb przedstawia się, jak podano w tablicy 5-1.

T a b l i c a 5-1

Program obliczania $a_1+a_2+a_3$

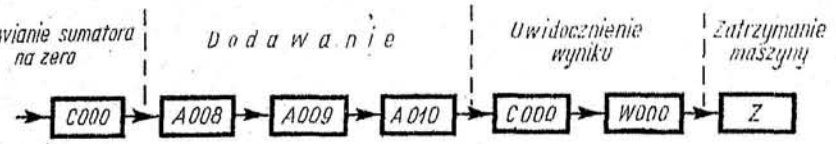
| N | Rozkaz | Wynik operacji |
|---|---|---|
| 001 | C 000 | Ustawienie sumatora na zero |
| 002 | A 008 | Umieszczenie $a_1$ w sumatorze |
| 003 | A 009 | Utworzenie sumy $a_1+a_2$ |
| 004 | A 010 | Utworzenie sumy $a_1+a_2+a_3$ |
| 005 | C 000 | Przeniesienie ostatniej sumy do pamięci |
| 006 | W 000 | Wydrukowanie wyniku |
| 007 | Z | Zatrzymanie maszyny |

W rubryce zatytułowanej N podane są numery miejsc pamięciowych, w których ulokowane są poszczególne rozkazy. Program zaczyna się od rozkazu C 000. Operacja C zawiera przeniesienie zawartości sumatora do pamięci oraz ustawienie sumatora na zero. W danym przypadku tylko ta druga czynność jest ważna. Zaczynamy program od operacji C, ponieważ nie wiemy czy w sumatorze nie pozostała z poprzedniego programu liczba różna od zera. Miejsce pamięciowe 000 gra w tym przypadku rolę jak gdyby kosza na śmiecie, do którego wyrzucamy poprzednią zawartość sumatora.

Rozkazy 002, 003, 004 kierują właściwym sumowaniem, po ukończeniu którego maszyna powinna uwidocznić wynik. Konstrukcje maszyn przeważnie nie przewidują możliwości przenoszenia zawartości sumatora bezpośrednio do urządzenia drukującego na wyjściu, wobec tego przeniesienie to należy zaprogramować za pomocą dwóch rozkazów. Rozkaz 005 przenosi sumę $a_1+a_2+a_3$ do pamięci, rozkaz zaś 006 powoduje wydrukowanie tej sumy. Przy przenoszeniu mo-

żemy wykorzystać ponownie miejsce pamięciowe 000. Możność ta wynika z okoliczności, że wprowadzenie nowego zapisu do miejsca pamięciowego kasuje automatycznie zapis poprzedni.

Zrozumienie programu, jak również jego układanie, ułatwia znakomicie tzw. wykres programu. W omawianym przypadku wykres programu przedstawia się jak podano na rys. 5-1. Jak widać, wszystkie rozkazy układają się tu na odcinku prostej.




Rys. 5-1. Wykres programu dodawania trzech liczb

Brak znaczenia praktycznego takiego programu wynika z tego, że jego napisanie i przeniesienie do pamięci maszyny trwałoby wiele tysięcy razy dłużej, niż samo obliczenie. Są przypadki, gdy program ma wartość, jako tzw. subprogram, to znaczy, jako samodzielna część obszerniejszego programu. Rozpatrzony program nie jest jednakże interesujący nawet jako subprogram, ponieważ dotyczy on dodania określonej liczby składników. Wartościowy byłby natomiast taki program dodawania, którego postać nie zależałaby od liczby składników i który mógłby być zatem stosowany natychmiast, gdy zostałyby dane składniki.

Ułożenie takiego uniwersalnego programu dodawania i w ogóle układanie programów o realnym znaczeniu dla eksploatacji maszyn cyfrowych jest możliwe dopiero przy użyciu dwóch specjalnych chwytów programowania, jakimi są stosowanie rozkazów warunkowych oraz obliczanie rozkazów.

## 52. Rozkazy warunkowe

Zasadniczy tok kolejnego wykorzystywania rozkazów może być przerwany przez wtrącenie rozkazu, który sam przez się nie zawiera operacji, uzależnia natomiast

numer miejsca pamięciowego, z którego ma być pobrany następny rozkaz, od określonego wyniku dotychczasowych obliczeń. W ten sposób, najogólniej sformułowany rozkaz warunkowy miałby brzmienie następujące: "Jeśli liczba zawarta obecnie w tym a w tym miejscu maszyny ma pewną właściwość, to następny rozkaz pobierz z miejsca pamięciowego o numerze takim a takim, jeśli natomiast, omawiana liczba właściwości tej nie ma, użyj kolejny rozkaz programu".

Liczbę decydującą o dalszym przebiegu obliczenia będziemy nazywali wyróżnikiem rozkazu warunkowego. Jej miejscem jest z reguły sumator, właściwością decydującą o dalszym biegu obliczenia - jest zazwyczaj znak wyróżnika.

Druga część rozkazu warunkowego może być pomijana, ponieważ opisuje ona przebieg regularny wykonywania programu. Wobec tego rozkaz warunkowy o treści, takiej, jaka spotyka się w wielu maszynach, może być sformułowany następująco: "Jeśli zawartość sumatora jest ujemna, to następny rozkaz pobierz z miejsca pamięciowego o podanym numerze". Rozkaz warunkowy wymienionej treści będziemy oznaczali literą X wraz z numerem następnego rozkazu, który powinien być użyty, jeśli zawartość sumatora jest ujemna. Dalszy tok obliczeń opiera się już na ogólnej regule następstwa rozkazów aż do napotkania następnego rozkazu warunkowego.

Przebieg rozkazów w kolejności, w której są one umieszczone w pamięci, nazywa się przebiegiem bezwarunkowym, którykolwiek natomiast przebieg, w ciągu którego działa chociażby jeden rozkaz warunkowy nazywa się przebiegiem warunkowym.

Jeśli chwilowa wartość wyróżnika rozkazu warunkowego powoduje przebieg warunkowy, będziemy mówili, że rozkaz warunkowy działa, jeśli natomiast przebieg pozostaje bezwarunkowy, będziemy mówili, że rozkaz warunkowy nie działa.

Rozpatrzmy następujący prosty przykład programu. Niechaj zadaniem maszyny będzie umieszczenie w N 006 wartości bezwzględnej zawartej tam liczby a. Jasne jest, że przebieg rozwiązania musi mieć charakter warunkowy. Jeśli a jest dodatnie, to w N 006 maszyna powinna pozostawić a, jeśli a jest ujemne, - umieścić - a. Dla sformułowania programu potrzebny nam jest,

oprócz niektórych rozkazów wymienionych w rozdziale 51, rozkaz warunkowy i rozkaz odejmowania o treści następującej: "Odejmij od zawartości sumatora liczbę zawartą w miejscu pamięciowym o podanym numerze". Rozkaz ten będziemy oznaczali literą B wraz z numerem miejsca pamięciowego.

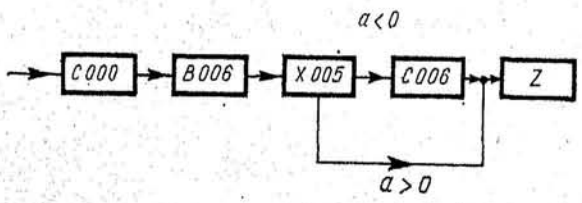
Program określania wartości bezwzględnej zawarty jest w tablicy 5-2, wykres zaś programu - na rys. 5-2.

Tablica 5-2

Program wyznaczania wartości bezwzględnej

| N | Rozkaz | Wynik operacji |
|---|---|---|
| 001 | C 000 | Ustawienie sumatora na zero |
| 002 | B 006 | Obliczenie -a |
| 003 | X 005 | Przeskok do rozkazu 005, jeśli a>0 |
| 004 | C 006 | Przeniesienie -a do pamięci |
| 005 | Z | Zatrzymanie maszyny |

Jako drugi przykład rozpatrzymy określanie największej spośród czterech liczb, które oznaczymy literami a, b, c, d. Rozwiązanie oprzemy na następują-




Rys. 5-2. Wykres programu wyznaczania wartości bezwzględnej

cym schemacie. Przed rozpoczęciem obliczenia porównywane liczby lokujemy na określonych miejscach pamięciowych, następnie obieramy pewne dodatkowe miejsce pamięciowe, np. 021, w którym będziemy lokowali. każdorazowo największą z dotychczas porównanych liczb. Zaczynamy od tego, aby liczbę a przenieść z jej włas-

nego miejsca na miejsce 021; z kolei przenosimy do sumatora liczbę b i tworzymy różnicę b-a, następnie zaś stosujemy rozkaz warunkowy. Jeśli b=a, to obliczona różnica jest nieujemna i rozkaz warunkowy nie działa. W dalszym ciągu programu przewidujemy umieszczenie b zamiast a na miejscu 021. Jeśli a<b, to rozkaz warunkowy będzie działał, co powinno być wykorzystane do ominięcia rozkazów umieszczających b na miejscu 021. W wyniku na miejscu pamięciowym 021 pozostaje zarejestrowane sup/a,b/, czyli większa z dwóch liczb a i b. Podobnie postępujemy z liczbą c, następnie zaś z d, w wyniku czego na miejscu 021 otrzymujemy sup/a,b,c,d/, czyli największą spośród porównywanych liczb. Program zagadnienia podany jest w tabl. 5-3, wykres programu - na rys. 5-3.

T a b l i c a 5-3

Program znajdowania największej z czterech liczb

| N | Rozkaz | Wynik operacji |
|---|---|---|
| 001 | C 000 | Ustawienie sumatora na zero |
| 002 | A 022 | Przeniesienie a do sumatora |
| 003 | C 021 | Przeniesienie a do pamięci |
| 004 | A 023 | Przeniesienie b do sumatora |
| 005 | B 021 | Utworzenie różnicy b-a |
| 006 | X 009 | Przeskok do rozkazu 009, jeśli b<a |
| 007 | A 021 | Obliczenie b |
| 008 | C 021 | Przeniesienie b do pamięci |
| 009 | A 024 | Przeniesienie c do sumatora |
| 010 | B 023 | Utworzenie różnicy c-sup/a,b/ |
| 011 | X 015 | Przeskok do rozkazu 015, jeśli c<sup/a,b/ |

Tablica 5-3 (c.d)

| N | Rozkaz | Wynik operacji |
|---|---|---|
| 012 | A 021 | Obliczenie c |
| 013 | C 021 | Przeniesienie c do pamięci |
| 014 | A 025 | Przeniesienie d do sumatora |
| 015 | B 021 | Utworzenie różnicy d-sup/a,b,c/ |
| 016 | X 019 | Przeskok do rozkazu 021, jeśli d-sup/a,b,c/ |
| 017 | A 021 | Obliczenie d |
| 018 | C 021 | Przeniesienie d do pamięci |
| 019 | W 021 | Wydrukowanie sup/a,b,c,d/ |
| 020 | Z | Zatrzymanie maszyny |

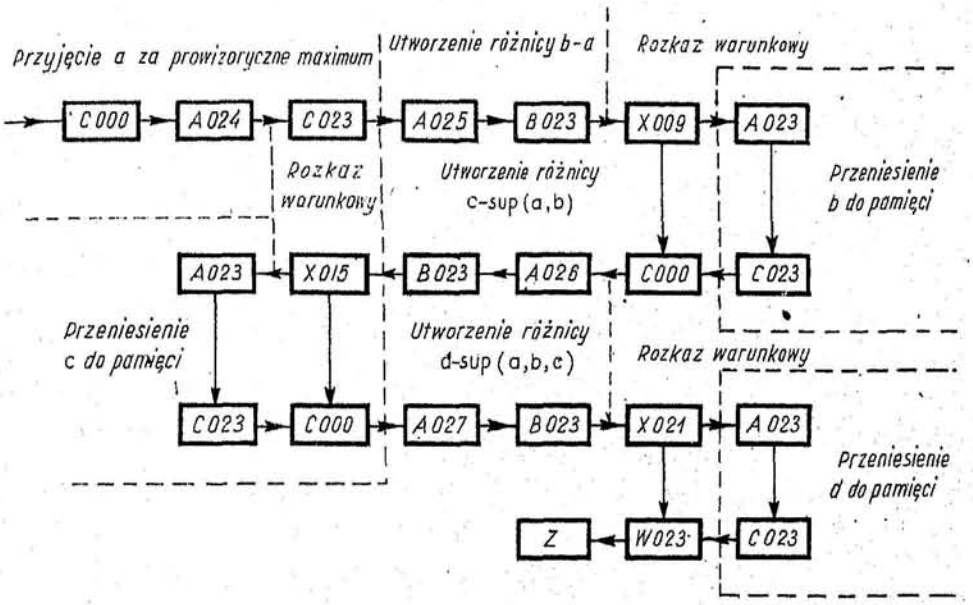
Lokata danych i wyników obliczeń

| N | Wielkość |
|---|---|
| 021 | liczba największa |
| 022 | a |
| 023 | b |
| 024 | c |
| 025 | d |

liczby porównywane

W programie tym na uwagę zasługują następujące szczegóły. Po rozkazie X 009 w sumatorze tkwi różnica b-a. Jeśli b>a, to należy przenieść b do N 023. Można to wykonać w ten sposób, żeby ustawić sumator na zero, przenieść b do sumatora, stąd zaś do N 023. Wymaga to trzech operacji. Można to samo osiągnąć w dwóch dodając a do zawartości sumatora

i przenosząc wynik do N 023. Jeśli b < a, to przed przystąpieniem do porównywania c z a należy ustawić sumator na zero za pomocą rozkazu C 000. Analogiczna sytuacja zachodzi po rozkazie X 015 i częściowo po X 021.




Rys. 5-3. Wykres programu znajdowania największej z czterech liczb

Jako trzeci przykład stosowania rozkazów warunkowych rozpatrzymy program obliczania wartości funkcji danej wzorem

$$y = ax^2+bx+c$$

dla wartości x zmieniających się co h, od m do n. Zakładamy, że (n-m)/h jest liczbą całkowitą. Dla napisania programu będziemy potrzebowali jeszcze dwóch operacji, mianowicie D: "Przeniesienie do rejestru mnożenia liczby ze wskazanego miejsca pamięciowego" i E: "Pomnożenie zawartości rejestru mnożenia przez liczbę umieszczoną we wskazanym miejscu pamięciowym i dodanie iloczynu do zawartości sumatora".

Tablica 5-4

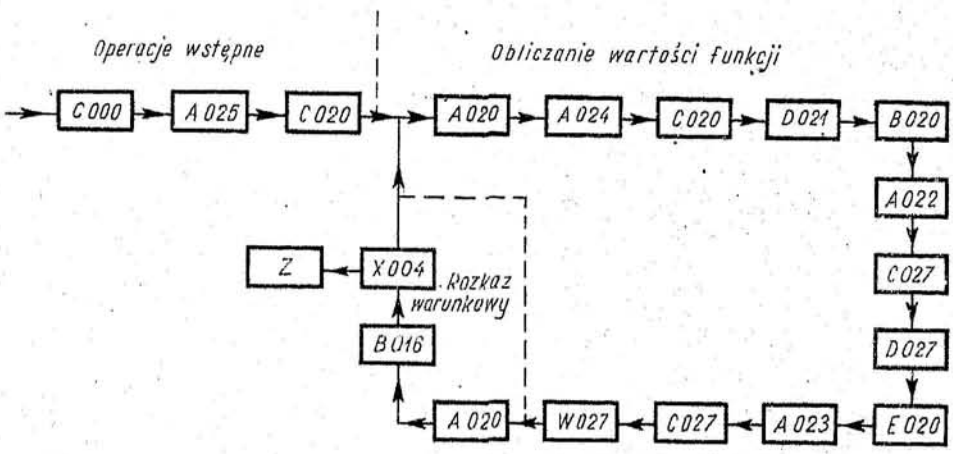
Program obliczania wartości funkcji $y = ax^2+bx+c$

| ES | N | Rozkaz | Wynik operacji |
|---|---|---|---|
| 01 | 001 | C 000 | Ustawienie sumatora na zero |
| | 002 | A 025 | Przeniesienie m-h do sumatora |
| | 003 | C 020 | Położenie x = m-h |
| 02 | 004 | A 020 | Przeniesienie x-h do sumatora |
| | 005 | A 024 | Obliczenie x |
| | 006 | C 020 | Przeniesienie x do pamięci |
| | 007 | D 021 | Przeniesienie a do rejestru mnożenia |
| | 008 | E 020 | Obliczenie ax |
| | 009 | A 022 | Obliczenie ax+b |
| | 010 | C 027 | Przeniesienie ax+b do pamięci |
| | 011 | D 027 | Przeniesienie ax+b do rejestru mnożenia |
| | 012 | E 020 | Obliczenie $ax^2+bx$ |
| | 013 | A 023 | Obliczenie $ax^2+bx+c$ |
| | 014 | C 027 | Przeniesienie wyniku do pamięci |
| | 015 | W 027 | Wydrukowanie wyniku |
| 03/02 | 016 | A 020 | Przeniesienie x do sumatora |
| | 017 | B 026 | Obliczenie różnicy x-n |
| | 018 | X 004 | Powrót do rozkazu 004, jeśli x<n |
| 04 | 019 | Z | Zatrzymanie maszyny |

Lokata danych i wyników obliczeń

| N | Wielkość ulokowana | N | Wielkość ulokowana |
|---|---|---|---|
| 020 | x - zmienne | 024 | h - stałe |
| 021 | a - stałe | 025 | m-h - stałe |
| 022 | b - stałe | 026 | n - stałe |
| 023 | c - stałe | 027 | y - zmienne |

Program jest podany w tablicy 5-4, a wykres programu na rys. 5-4. W odróżnieniu od dotychczasowych programów mamy tu ciąg piętnastu rozkazów od 004 do 018, które są przebiegane wielokrotnie. Ciąg taki będziemy nazywali zamkniętym lub też cyklem rozkazów. Na wykresie programu cykl uwidacznia się istnieniem pętli. W programie zawartym w tablicy 5-4 możemy wyróżnić dwie części cyklu, mianowicie część operatywną obejmującą rozkazy od 004 do 015, w ramach których następuje obliczenie wartości zmiennych x i y, oraz część warunkową, zawierającą obliczenie wyróżnika rozkazu warunkowego (016 i 017) oraz sam rozkaz warunkowy (018).




Rys. 5-4. Wykres programu obliczania wartości trójmianu kwadratowego

Podobieństwa i różnice między programami można bardzo ogólnie charakteryzować mówiąc, że mają one takie same lub różne struktury. Pod strukturą programu należy rozumieć te informacje o kolejności obiegania przez maszynę ciągów rozkazów, które pozostają po abstrahowaniu od treści poszczególnych rozkazów. Do bliższego sformułowania pojęcia struktury programu możemy dojść w sposób następujący.

Wyodrębniamy w programie takie ciągi rozkazów, z których każdy kończy się rozkazem warunkowym i zawiera ponadto te wszystkie rozkazy, które poprzedzają rozkaz warunkowy i zawierają obliczenie jego

wyróżnika. Każdy taki ciąg rozkazów będziemy nazywali elementem kierującym struktury programu.

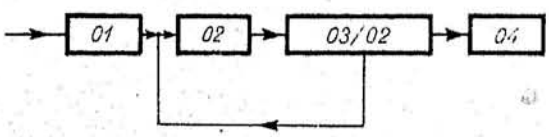
Wyodrębniamy następnie z programu ciągi rozkazów, które będziemy nazywali elementami wykonawczymi struktury programu. Każdy element wykonawczy składa się z kolejnych rozkazów wykonywanych jednakową liczbę razy.

Zastępując w wykresie programu poszczególne ciągi rozkazów przez odpowiednie elementy struktury otrzymamy wykres struktury programu. W wykresie struktury numerujemy wszystkie elementy kolejnymi liczbami, przy czym każdy element kierujący oznaczamy dodatkowo numerem tego elementu wykonawczego, do którego maszyna wraca, gdy rozkaz warunkowy działa.

Zatem element wykonawczy jest określony w ramach danego programu przez jedną liczbę, element zaś kierujący - przez parę uporządkowaną liczb. Podanie oznaczonych w ten sposób elementów struktury wyznacza jednoznacznie strukturę programu. Na przykład, jeśli dane są elementy

01, 02, 03/02, 04,

to wykres struktury może być natychmiast narysowany i obieg rozkazów programu tym samym całkowicie określony (rys. 5-5).




Rys. 5-5. Struktura programu obliczania wartości trójmianu kwadratowego

Wspomniane poprzednio pojęcie subprogramu ma duże znaczenie użytkowe, nie wyjaśnia jednakże w sposób przejrzysty działania programu i nie da się jednoznacznie powiązać z pojęciami elementów struktury. Pod subprogramem rozumie się na ogół ciąg rozkazów, służący do obliczania wartości jednego lub kilku nieznanych parametrów występujących w to-

ku wykonywania programu. Subprogram zawiera z reguły co najmniej jeden element strukturalny wykonawczy i jeden element kierujący, może zawierać ponadto ciągi rozkazów stanowiące części elementów wykonawczych.

Praktyczne znaczenie subprogramów tkwi w możności stosowania gotowych programów lub ich części do budowy bardziej skomplikowanych programów, praktyczne znaczenie pojęcia struktury ma charakter metodologiczny i polega na ułatwianiu analizy istniejących programów oraz na ułatwianiu projektowania nowych programów.

Stosowanie rozkazów warunkowych nie rozwiązuje jeszcze zasadniczych trudności programowania. Na przykład program obliczania wartości trójmianu kwadratowego opiera się na okoliczności, że wartość zmiennej niezależnej była obliczana w każdym cyklu przez maszynę, dzięki czemu adres zmiennej mógł być niezmienny. Gdyby wartości x były dane z góry, to każda z nich miałaby inny adres i za pomocą dotychczas rozpatrzonych metod programowania nie umielibyśmy napisać programu obliczenia y.

## 53. Obliczanie adresów

Wiemy o tym, że każdemu rozkazowi jest przyporządkowana w maszynie pewna liczba. Będziemy mówili, że liczba ta określa rozkaz. Kilka jej pierwszych znaków określa operację, pozostałe - adres.

Oznaczmy przez r liczbę określającą jakiś rozkaz. Wówczas liczby r+1, r+2 itd. aż do r+s określają pewne inne rozkazy, które jeśli s jest dostatecznie małe, zawierają tę samą operację, lecz różnią się adresami. Wynikające stąd możliwości programowania są wyzyskiwane w sposób następujący. Przypuśćmy, że mamy obliczenie, które da się przeprowadzić za pomocą programu zawierającego jeden lub więcej cykli. W obliczenie wchodzą wielkości, z których co najmniej jedna w każdym cyklu przybiera inną wartość. Taką wielkość lub takie wielkości będziemy nazywali zmiennymi. Załóżmy, że wielkości zmienne nie są obliczane przez maszynę, lecz dane z góry. Wówczas operacje obliczeniowe w każdym cyklu poprzedzamy obliczeniami adresów wielkości zmiennych.

Tablica 5-5

Program obliczania $\sum_{1}^{n} a_i$. Metoda obliczania:

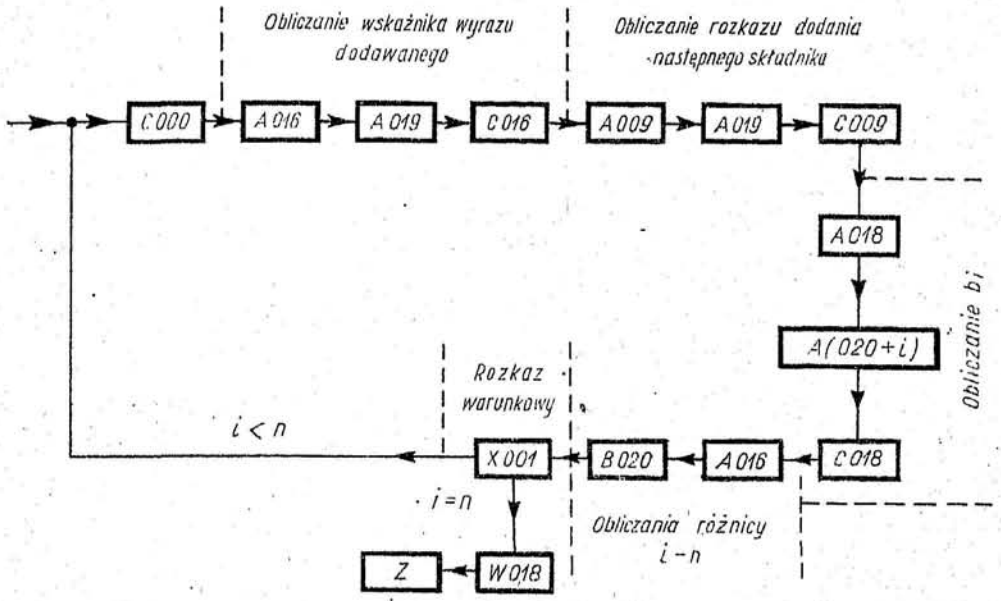
$b_i = b_{i-1} + a_i$ ; $b_0 = 0$.

| ES | N | Rozkaz | Wynik operacji |
|---|---|---|---|
| 01 | 001 | C 000 | Ustawienie sumatora na zero |
| | 002 | A 016 | Przeniesienie i-1 do sumatora |
| | 003 | A 018 | Obliczenie i |
| | 004 | C 016 | Przeniesienie i do pamięci |
| | 005 | A 009 | Przeniesienie rozkazu A(019+i) do sumatora |
| | 006 | A 018 | Obliczenie rozkazu A(020+i) |
| | 007 | C 009 | Przeniesienie rozkazu A(020+i) do pamięci |
| | 008 | A 017 | Przeniesienie $b_{i-1}$ do sumatora |
| | 009 | A(020+i) | Obliczenie $b_i$ |
| | 010 | C 017 | Przeniesienie $b_i$ do pamięci |
| 01/02 | 011 | A 016 | Przeniesienie i do sumatora |
| | 012 | B 019 | Obliczenie i-n |
| | 013 | X 001 | Decyzja powrotu do rozkazu 001, jeśli i<n |
| 03 | 014 | W 017 | Wydrukowanie $b_n$ |
| | 015 | Z | Zatrzymanie maszyny |

Lokata danych i wyników obliczeń

| N | Wielkość | N | Wielkość |
|---|---|---|---|
| 016 | i, na początku 0 | 021 | $a_1$ |
| 017 | $b_i$, na początku 0 | 022 | $a_2$ |
| 018 | 1 | 023 | $a_3$ |
| 019 | n | 024 | $a_4$ |
| 020 | obojętna | itd. | itd. |

Przykładem programu zawierającego obliczanie adresów może być program dodawania n liczb (tabl. 5-5 i rys. 5-6). Zrozumienie programu korzystającego




Rys. 5-6. Wykres programu obliczania sumy n liczb

z metody obliczania rozkazów ułatwia znakomicie tablica zmian zawartości miejsc pamięciowych (tabl. 5-6). Struktura tego programu jest jeszcze prostsza niż programu z tabl. 5-4. Jest ona określona mianowicie elementami strukturalnymi

01, 02/01, 03.

Tablica 5-6

Zmiany zawartości miejsc pamięciowych w programie obliczania sumy n liczb

| N | Zawartość | | | |
|---|---|---|---|---|
| | na początku programu | po pierwszym cyklu | po drugim cyklu | po n-tym cyklu |
| 009 | rozkaz A 020 | A 021 | A 022 | A(020+n) |
| 016 | 0 | 1 | 2 | n |
| 017 | 0 | $b_1$ | $b_2$ | $b_n$ |

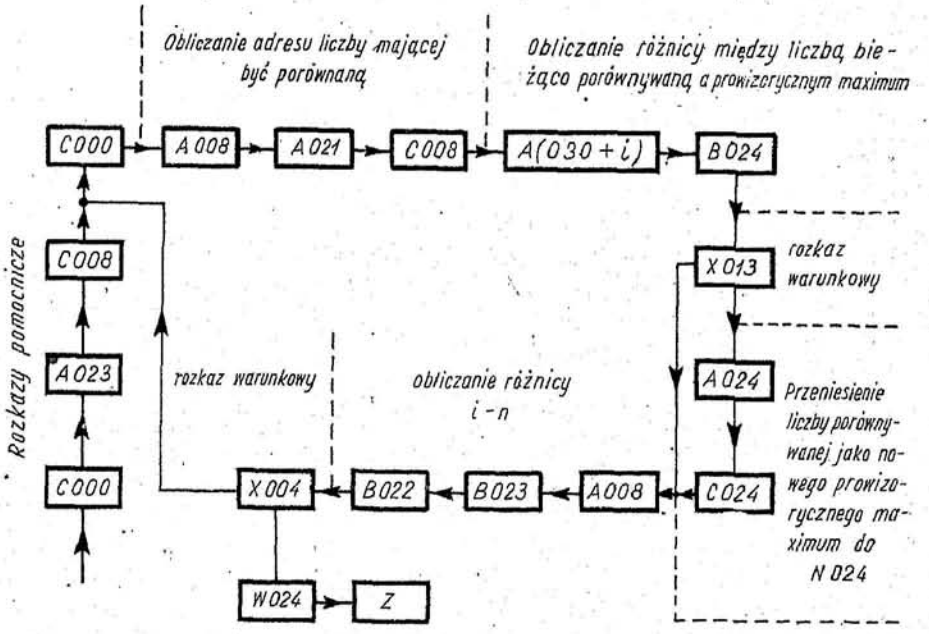
Tablica 5-7

Program znajdowania największej spośród n liczb

| ES | N | Rozkaz | Wynik operacji |
|---|---|---|---|
| 01 | 001 | C 000 | Ustawienie sumatora na zero |
| | 002 | A 023 | Przeniesienie rozkazu A 030 do sumatora |
| | 003 | C 008 | Przeniesienie rozkazu A 030 do pamięci |
| 02/04 | 004 | C 000 | Ustawienie sumatora na zero |
| | 005 | A 008 | Przeniesienie rozkazu A 030+i-1 do sumatora |
| | 006 | A 021 | Utworzenie rozkazu A 030+i |
| | 007 | C 008 | Przeniesienie rozkazu A 030+i do pamięci |
| | 008 | A 030+i | Przeniesienie $a_i$ do sumatora |
| | 009 | B 024 | Odjęcie prowizorycznego maximum |
| | 010 | X 013 | Przeskok do rozkazu 013, jeśli $a_i \leq a_{pm}$ |
| 03 | 011 | A 024 | Obliczenie $a_i$ |
| | 012 | C 024 | Przeniesienie $a_i$ do pamięci |
| 04/02 | 013 | A 008 | Przeniesienie rozkazu A 030+i do sumatora |
| | 014 | B 023 | Obliczenie i |
| | 015 | B 022 | Obliczenie różnicy i-n |
| | 016 | X 004 | Powrót do rozkazu 004, jeśli i < n |
| 05 | 017 | W 024 | Wydrukowanie $a_{max}$ |
| | 018 | Z | Zatrzymanie maszyny |

| N | Wielkość | N | Wielkość |
|---|---|---|---|
| 020 | 0 | 031 | $a_1$ |
| 021 | 1 | 032 | $a_2$ |
| 022 | n | 033 | $a_3$ |
| 023 | Rozkaz A 030 | itd. | itd. |
| 024 | Liczba największa | | |

Drugi przykład nieskomplikowanego programu stosującego obliczanie rozkazów dotyczy znajdowania największej spośród n liczb. Program tego zagadnienia




Rys. 5-7. Wykres programu znajdowania największej spośród n liczb

podany jest w tabl. 5-7, wykres programu - na rys. 5-7, zmiany zawartości miejsc pamięciowych - w tablicy 5-8. Struktura tego programu jest określona przez następujące elementy

01, 02/04, 03, 04/02, 05.

Tablica 5-8

Zmiany zawartości miejsc pamięciowych w programie znajdowania największej spośród n liczb

| N | Zawartość | | | |
|---|---|---|---|---|
| | na początku programu | po pierwszym cyklu | po drugim cyklu | po n-tym cyklu |
| 008 | rozkaz A 030 | A 031 | A 032 | A(030+n) |
| 024 | 1 | $a_1$ | sup($a_1$, $a_2$) | sup(n liczb) |

Na podstawie wykresu struktury i tablicy zaliczenia rozkazów do elementów struktury można łatwo narysować wykres programu. Posiłkowanie się obu metodami naraz jest na ogół bezcelowe. Do interpretacji prostych programów lepiej nadają się wykresy programów, do bardziej skomplikowanych, gdy liczba rozkazów jest znaczna i wykres programów staje się nieprzejrzysty, lepiej nadają się wykresy struktury.

O efektywności programów stosujących rozkazy warunkowe i obliczanie adresów świadczy np. to, że program liniowy porównywania czterech liczb wymaga dwudziestu dwóch rozkazów, program zaś cykliczny*) porównywania dowolnej liczby liczb - tylko osiemnastu rozkazów.

## 54. Komplety rozkazów

W dotychczas rozpatrywanych przykładach programów używaliśmy kilku operacji, powiększając ich liczbę w miarę komplikacji zagadnień. Dla programowania dowolnego zagadnienia numerycznego komplet operacji powinien być nieco obszerniejszy. Lista potrzebnych operacji nie da się jednakże ustalić jednoznacznie. Chodzi o to, że niewiele jest operacji bez których programowanie rozwiązania problematu numerycznego byłoby w ogóle niemożliwe. Nawet operacja mnożenia nie jest niezbędna, ponieważ można ją zastąpić ciągiem odpowiednio dobranych dodawań i przesunięć. Niewątpliwie jednakże programowanie mnożenia byłoby w tym przypadku niezmiernie uciążliwe i możliwość popełniania błędów odpowiednio większa.

W tym właśnie tkwi sedno sprawy. Dobór kompletu operacji jest w znacznej części zagadnieniem kompromisu między komplikacjami i kosztami wbudowania w maszynę wielu operacji a trudnościami programowania za pomocą niewielkiej liczby operacji. Obiektywny dobór mógłby być oparty na materiale statystycznym ale materiał taki byłby trudny do zdobycia i trudny do przeanalizowania. W praktyce decyduje tu przybliżona ocena i intuicja konstruktora. Na ogół duże maszyny są zaopatrywane w duże komplety operacji, mniejsze zaś - w skromniejsze komplety. W rozdziale

*) Program cykliczny bywa nazywany również indukcyjnym lub iteracyjnym.

62 podany jest komplet operacji maszyny typu 701, który należy do raczej obszernych.

Do celów ćwiczebnych będziemy się posiłkowali kompletem podanym w tabl. 5-9.

Tablica 5-9

Zestawienie rozkazów ćwiczebnych

| Symbol | Treść rozkazu |
|---|---|
| A | Dodaj do zawartości sumatora liczbę pobraną z miejsca pamięciowego N. |
| B | Odejm z zawartości sumatora liczbę pobraną z miejsca pamięciowego N. |
| C | Przenieś zawartość sumatora do miejsca pamięciowego N i ustaw sumator na zero. |
| D | Przenieś do rejestru mnożenia liczbę pobraną z miejsca pamięciowego N. |
| E | Pomnóż liczbę pobraną z miejsca pamięciowego N przez zawartość rejestru mnożenia i dodaj iloczyn do zawartości sumatora. |
| G | Podziel zawartość sumatora i rejestru mnożenia, traktując je, jako jedną liczbę, przez liczbę pobraną z miejsca pamięciowego N. Iloraz pozostaw w rejestrze mnożenia, resztę - w sumatorze. |
| H | Przenieś zawartość rejestru mnożenia do miejsca pamięciowego N. |
| W | Wydrukuj wynik zarejestrowany w miejscu pamięciowym N. |
| X | Jeśli zawartość sumatora jest ujemna, przejdź do rozkazu zawartego w miejscu pamięciowym N, jeśli jest nieujemna, przejdź do rozkazu kolejnego. |