

## **Automatyczne maszyny cyfrowe i ich zastosowanie w geodezji**

### **1. Zasady budowy i działania automatycznych maszyn cyfrowych**

#### *1.1. Wstęp*

Zbudowanie automatycznych maszyn cyfrowych jest jednym z najważniejszych osiągnięć myśli ludzkiej w ostatnich latach. Maszyny te zadziwiają nas niewiarygodną szybkością pracy, wyrażającą się dziesiątkami, a nawet setkami tysięcy operacji arytmetycznych i logicznych na sekundę. Zaprzężone do pracy elektrony, zgodnie z ułożonym przez człowieka programem, wykonują różnorodne obliczenia, tłumaczą na obce języki, kierują procesami produkcyjnymi fabryk, sterują lotem samolotów. Automatyczne maszyny cyfrowe coraz częściej zastępują człowieka w wykonywaniu pracy umysłowej, a dzięki swej uniwersalności i szybkości działania umożliwiają wykonanie zadań, o których realizacji trudno by było kiedyś pomyśleć. Przykładem tego może być rozwiązywanie szczególnie skomplikowanych układów równań różniczkowych, jakie występują przy badaniach rakiet sterowanych. Również w geodezji automatyczne maszyny cyfrowe znajdują duże zastosowanie. Przy ich pomocy mogą być rozwiązywane wszelkie zadania obliczeń geodezyjnych, ale najbardziej celowe ze względów ekonomicznych jest stosowanie tych maszyn do zadań skomplikowanych, w szczególności takich, w których ilość działań arytmetycznych jest dość znaczna w porównaniu do ilości danych liczbowych.

Od dawna stosuje się maszyny automatyczne do rozwiązywania układów równań normalnych, układania różnorodnych tablic, przeliczania współrzędnych itd. Obok maszyn uniwersalnych, przy pomocy których można rozwiązywać szeroki wachlarz problemów, buduje się maszyny specjalistyczne o stałych programach. W Stanach Zjednoczonych powstała maszyna przeznaczona do rozwiązywania układów równań liniowych (ilość równań może dochodzić do 1200). W Niemczech Zachodnich firma

Zuse produkuje przekątnikową, stosunkowo wolno liczącą maszynę Z11, która posiada kilkanaście stałych programów umożliwiających rozwiązywanie typowych zadań geodezyjnych, takich jak np. wcięcie w przód, wcięcie wstecz, transformacja, obliczenie powierzchni ze współrzędnych itd. W Polsce buduje się maszynę przekątnikową PARC, przeznaczoną specjalnie dla obliczeń geodezyjnych. Projekt maszyny opracowany został w Katedrze Obliczeń Geodezyjnych Akademii Górniczo-Hutniczej przez mgra G. Kudelskiego.

Niewątpliwie już w najbliższej przyszłości będziemy świadkami szerokiego rozpowszechnienia maszyn automatycznych w geodezji. Z tego względu warto, choć pobieżnie zapoznać się z zasadami ich działania, elementami konstrukcji i sposobami układania programów.

### 1.2. Klasyfikacja maszyn liczących

Istnieje wiele różnych typów maszyn liczących. Zapoznamy się z ich ogólną klasyfikacją, aby ustalić jakie miejsce w tej klasyfikacji zajmują maszyny automatyczne.

Wszystkie maszyny liczące dzielimy na dwie podstawowe grupy: maszyny o działaniu ciągłym, tzw. maszyny analogowe i maszyny o działaniu numerycznym, tzw. maszyny cyfrowe. Zasadnicza różnica między tymi dwoma grupami maszyn polega na tym, że w maszynach analogowych dane początkowe, przebieg i wyniki rachunku występują pod postacią funkcji ciągłych, zaś w maszynach cyfrowych w postaci nieciągłej, numerycznej.

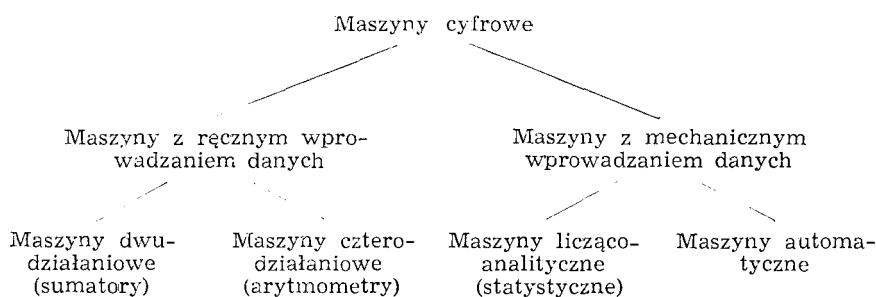
Maszyna analogowa przedstawia sobą fizyczny model, analogię zadania, do którego rozwiązania jest ona przeznaczona. Maszyny analogowe są zatem maszynami specjalistycznymi, każda z nich rozwiązuje zagadnienia należące do pewnej wąskiej klasy, np. buduje się maszyny rozwiązujące równania różniczkowe. Maszyny tego rodzaju są na ogół proste w budowie i stosunkowo tanie, natomiast ich wadą jest niewielka dokładność rachunku, wyrażająca się błędem względnym rzędu 0,1—1%.

Najprostszy i najbardziej rozpowszechniony przyrządem liczącymi, które można zaliczyć do pierwszej grupy maszyn, są suwaki logarytmiczne i planimetry.

W odróżnieniu od maszyn analogowych maszyny cyfrowe odznaczają się wysoką dokładnością rachunku, zależną tylko od liczby pozycji w licznikach.

Opierając się na sposobie wprowadzania danych (i otrzymywania wyników) podzielimy maszyny cyfrowe na grupy, tak jak to zostało pokazane na rys. 9.

Najprostszymi maszynami cyfrowymi są maszyny przeznaczone do wykonywania tylko dwóch działań: dodawania i odejmowania. Do maszyn dwudziałaniowych możemy zaliczyć sumatory, adiatory, a także liczydła. Maszyny te stosuje się przede wszystkim w księgowości. Natomiast w technice powszechne zastosowanie znalazły wszelkiego rodzaju arytmometry: dźwigniowe i klawiszowe, o napędzie ręcznym i elektrycznym. Maszyny tego typu produkowane są w wielkich ilościach. Ostatnio pojawiły się prototypy arytmometrów, przy pomocy których można wykonać więcej niż 4 działania. Prof. dr Ramsayer ze Stuttgartu skonstruował arytmometr umożliwiający obliczenie wartości funkcji  $\sin x$ ,  $\cos x$ ,  $\sec x$ ,  $\operatorname{tg} x$ ,  $\operatorname{arctg} x$ ,  $\operatorname{arcsin} x$  i  $\sqrt{x}$  z ośmiocyfrową dokładnością.



Rys. 9

Dane liczbowe na sumatorach i arytmometrach wprowadza się ręcznie przy pomocy szeregu dźwigni lub też odpowiedniej klawiatury.

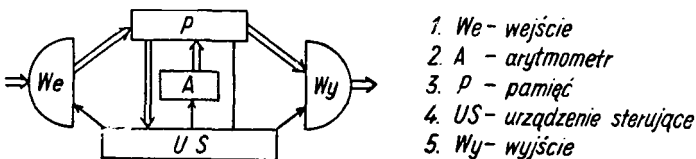
Maszyny licząco-analityczne (statystyczne), tworzące następną grupę w klasyfikacji, operują danymi liczbowymi wprowadzonymi mechanicznie na tzw. kartach dziurkowanych. Maszyny te użytkuje się w zespołach, w skład których wchodzi między innymi: dziurkująca karty dziurkarka, sortujący karty sorter, sumujący dane i drukujący wyniki tabulator oraz wykonująca szereg działań arytmetycznych mnożarka. Szybkość pracy poszczególnych maszyn zespołu zawiera się w granicach od kilku tysięcy do kilkudziesięciu tysięcy operacji na godzinę, a więc, w porównaniu z arytmometrami i sumatorami, jest dość znaczna. Maszyny licząco-analityczne przynoszą wielkie korzyści ekonomiczne w statystyce, księgowości i planowaniu, wszędzie tam, gdzie bogaty materiał liczbowy poddawany jest niezbyt skomplikowanym operacjom rachunkowym, informacje zaś są wielokrotnie opracowywane. Nie nadają się one jednak (ze względów ekonomicznych) do rozwiązywania zadań matematycznych, które cechuje duża ilość następujących po sobie różnych operacji oraz konieczność częstych zmian toku obliczeń, w zależności od wyników pośrednich. Zadaniom takim w pełni odpowiadają nowoczesne maszyny

automatyczne. Określenie „automatyczne” oznacza tu, że od momentu wprowadzenia na drodze mechanicznej (lub niekiedy ręcznej) wszelkich danych potrzebnych do obliczeń, aż do momentu wydrukowania wyników, maszyna nie potrzebuje ingerencji człowieka.

Pierwsze automatyczne maszyny cyfrowe budowano przy pomocy elementów przekaźnikowych, były to więc maszyny elektromechaniczne, o małej szybkości pracy, rzędu kilku operacji na sekundę. Osiągnięcie wielkich szybkości cechujących współczesne maszyny było możliwe dopiero po zastosowaniu techniki elektronowej. Początkowo w konstrukcjach maszyn elektronowych używano lamp, podobnych do tych, jakie znajdują się w odbiornikach radiowych, obecnie coraz częściej wprowadza się elementy półprzewodnikowe, osiągając przez to znaczne korzyści, polegające przede wszystkim na zmniejszeniu wagi maszyny i zużycia energii, a także na zwiększeniu pewności jej pracy.

### 1.3. Ogólny opis maszyny automatycznej

Zastanowimy się teraz nad pytaniem: Z jakich podstawowych urządzeń składa się automatyczna maszyna cyfrowa? Maszyna ta zastępuje rachmistrza w jego pracy; rachmistrza, który odczytuje dane, liczy na arytmometrze, zapamiętuje wyniki pośrednie (lub zapisuje je sobie dla pamięci na papierze), zważa na kolejność wykonywanych czynności w ramach realizowanego algorytmu, wreszcie zapisuje wynik ostateczny. Wszystkie wymienione powyżej czynności maszyna musi wykonać w sobie właściwy sposób, przy pomocy specjalnych urządzeń.



1. We - wejście
2. A - arytмомetr
3. P - pamięć
4. US - urządzenie sterujące
5. Wy - wyjście

Rys. 10

Odczytanie danych, wprowadzenie ich do maszyny odbywa się w tzw. wejściu, obliczenia wykonuje arytмомetr, wszystkie niezbędne informacje potrzebne do przeprowadzenia obliczeń, w tym również wyniki pośrednie, przechowuje pamięć, nad kolejnością i charakterem wykonywanych operacji czuwa urządzenie sterujące, wyniki zaś ostateczne są drukowane przez urządzenie zwane wyjściem.

Praca maszyny polega na współdziałaniu ze sobą tych pięciu urządzeń.

Na rys. 10 pokazano przebiegi danych liczbowych i „rozkazów” do-

starczanych maszynie poprzez wejście, przy czym strzałkami cieńszymi przedstawiono połączenia z urządzeniem sterującym. „Rozkazy” są sygnałami, które powodują wykonanie przez maszynę ściśle określonych operacji, jak np. dodawanie, mnożenie, drukowanie itp. Warto zauważyć, że liczby i rozkazy przedstawia się w maszynie w jednakowy sposób — przy pomocy cyfr. Z kolei cyfry reprezentowane są przez różne wartości obranych parametrów elektrycznych, np. napięcia.

Ilość rozkazów — tworzących tzw. listę rozkazów — jest różna dla różnych typów maszyn; na ogół maszyny posiadają około 30 rozkazów. Im trudniejsze i bardziej skomplikowane operacje potrafi wykonać maszyna pod wpływem działania pojedynczego rozkazu, tym łatwiejsze jest programowanie, tzn. układanie programów. *P r o g r a m e m* nazywamy tu uporządkowany zespół rozkazów umożliwiający rozwiązanie danego maszynie zadania.

W większości stosowanych obecnie maszyn automatycznych operacje wykonuje się na liczbach przedstawionych w systemie dwójkowym, tj. w systemie o zasadzie równej 2. Przed bardziej szczegółowym opisem podstawowych urządzeń maszyn automatycznych podane zostaną pewne wiadomości o tym systemie.

#### 1.4. System dwójkowy

W powszechnym użyciu znajduje się system dziesiętny, nazywany tak dlatego, że każdą liczbę w tym systemie przedstawiamy w postaci sumy całkowitych potęg liczby dziesięć, np.:

$$115,2 = 1 \cdot 10^2 + 1 \cdot 10^1 + 5 \cdot 10^0 + 2 \cdot 10^{-1}$$

Liczbę 10 nazywamy tu zasadą systemu.

Przy zapisie liczb w systemie dziesiętnym używamy 10 różnych cyfr:

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9$$

Wspominaliśmy już o tym, że w maszynach cyfrowych każdej cyfrze przyporządkowuje się pewną wartość obranego parametru elektrycznego. Budowa maszyn pracujących w systemie dziesiętnym wymagałaby wobec tego użycia elementów posiadających dziesięć różnych stanów elektrycznych odpowiadających dziesięciu cyfrom \*). Nieporównanie prostsze i pewniejsze w działaniu są elementy konstrukcyjne wyróżniające tylko dwa stany elektryczne, np. przewodzenie i nieprzewodzenie. Z tego właśnie względu w budowie maszyn cyfrowych największe zastosowanie znalazł

---

\*) Można by też przyporządkować każdej cyfrze systemu dziesiętnego kilka elementów dwuwartościowych, jednakże komplikuje to budowę maszyny.

system, w którym występują jedynie dwie cyfry: 0 i 1, odpowiadające dwóm stanom elektrycznym. System taki nazywa się dwójkowym lub binarnym.

Maszyny cyfrowe, zbudowane w oparciu o system binarny, przeliczają dane liczbowe wprowadzone do maszyny w systemie dziesiętnym na system dwójkowy, rachunek prowadzą w tym systemie, po czym wyniki ostateczne przeliczają znów na system dziesiętny i w tej postaci je drukują. Przeliczanie następuje automatycznie lub przy pomocy specjalnych programów.

W systemie dziesiętnym liczby przedstawiamy w postaci sumy całkowitych potęg liczby dziesięć, w systemie dwójkowym natomiast — w postaci sumy całkowitych potęg liczby dwa. Tak więc samą liczbę dwa możemy zapisać wyraźnie w postaci:

$$\text{dwa} = 1 \cdot 2^1 + 0 \cdot 2^0 = 10$$

Liczba dwa jest w systemie dwójkowym liczbą dwucyfrową; do jej zapisania użyliśmy dwie cyfry: 10. Do zapisania liczby dziewięć potrzebowalibyśmy cztery cyfry:

$$\text{dziewięć} = 1001 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

A oto kilka innych przykładów:

$$\text{trzy} = 11 = 1 \cdot 2^1 + 1 \cdot 2^0$$

$$\text{piętnaście} = 1111 = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$\text{trzydzieści dwa} = 10000 = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$$

Działania arytmetyczne w systemie dwójkowym odznaczają się wyjątkową prostotą. Przy dodawaniu należy pamiętać tylko o tym, że:  $0 + 0 = 0$ ,  $1 + 0 = 1$ ,  $0 + 1 = 1$  i  $1 + 1 = 10$  (dwa).

Dla przykładu dodamy dziewięć i trzynaście:

$$1001 \text{ (dziewięć)}$$

$$1101 \text{ (trzynaście)}$$

$$10110 \text{ (dwadzieścia dwa)}$$

Przeliczanie z systemu dwójkowego na dziesiętny wygodnie jest wykonywać wypisując kolejne potęgi dwóch pod przeliczaną liczbą, np.:

$$10110$$

$$168421$$

i tworząc sumę tych potęg, którym odpowiadają jedynki liczby przeliczanej:

$$10110 = 16 + 4 + 2 = 22$$

Mnożenie w układzie dwójkowym polega na stosowaniu następującej tabliczki mnożenia:

$$\begin{array}{ll} 0 \cdot 0 = 0, & 0 \cdot 1 = 0, \\ 1 \cdot 0 = 0, & 1 \cdot 1 = 1. \end{array}$$

Pomnożymy dla przykładu dziewięć przez trzynaście:

$$\begin{array}{r} \text{mnożna } 1001 \text{ (dziewięć)} \\ \text{mnożnik } 1101 \text{ (trzynaście)} \\ \hline 1001 \\ 0000 \\ 1001 \\ 1001 \\ \hline \end{array}$$

iloczyn 1110101 (sto siedemnaście).

Jak widać, mnożenie sprowadza się do pojedynczego dodawania mnożonej na tych pozycjach, na których w mnożniku znajdują się jedynki.

W podobnie prosty sposób wykonuje się w systemie dwójkowym odejmowanie i dzielenie.

Znaki plus i minus w maszynie cyfrowej przedstawia się z reguły przy pomocy dodatkowej (pierwszej) pozycji cyfrowej. Przyjęto oznaczać zerem plus, jedynką zaś — minus.

Wszystkie liczby, którymi operuje maszyna znajdują się w pewnym przedziale, zwanym zakresem liczenia. W niektórych maszynach zakres liczenia obejmuje tylko liczby całkowite zawarte w przedziale od pewnej liczby  $-N$  do  $+N$ , przy czym liczba  $N$  określona jest ilością pozycji cyfrowych w maszynie. Pewną odmianą maszyn tego rodzaju są maszyny, których zakres liczenia obejmuje liczby mniejsze co do bezwzględnej wartości od jedności, ale większe od pewnej liczby  $x$ , określonej również ilością pozycji cyfrowych w maszynie. Obydwa typy maszyn nazywają się maszynami ze stałym przecinkiem. Liczenie ze stałym przecinkiem następuje wiele trudności, które wynikają między innymi z tego, że w stosunkowo wąskim zakresie liczenia trzeba zmieścić zarówno dane początkowe jak też wyniki pośrednie i ostateczne. Wiąże się to z koniecznością przeprowadzenia szczegółowej analizy procesu rachunkowego i wprowadzenia odpowiednich współczynników redukcyjnych w tych przypadkach, gdy liczby przekraczają zakres liczenia.

Oprócz maszyn ze stałym przecinkiem buduje się bardziej skomplikowane maszyny z przecinkiem zmiennym. W maszynach takich liczbę  $L$  przedstawia się na ogół w postaci:

$$L = d \cdot 10^p$$

gdzie  $d$  jest ułamkiem właściwym (mantysą), zawartym w granicach od  $\frac{1}{10}$  do 1, zaś  $p$  — wykładnikiem potęgi.

Dla przykładu weźmy zapisaną w systemie dziesiętnym liczbę  $L = 1015,21$ . Mamy tu  $d = 0,101521$  i  $p = 4$ :

$$1015,21 = 0,101521 \cdot 10^4$$

Tak więc w tym sposobie liczenia liczby przedstawia się przy pomocy dwóch grup cyfr odpowiadających mantysie i wykładnikowi potęgowemu. Przyjmując, że dla mantysy przewidziano 6 cyfr, dla wykładnika zaś potęgowego 2 cyfry (prócz tego 2 pozycje na znaki plus—minus) będziemy mogli zapisać liczbę 1015,21 jako:

$$+ 101521 + 04$$

Maszyny ze zmiennym przecinkiem mają większy zakres liczenia i ich eksploatacja jest łatwiejsza.

Istnieje jeszcze jeden sposób uwzględniania położenia przecinka, polegający na zastosowaniu odpowiednich programów elementarnych. Nie powoduje to komplikacji w budowie maszyny, ale zmniejsza szybkość rachunku i pojemność pamięci.

Liczby w maszynie automatycznej mają na ogół ustaloną ilość znaków cyfrowych. Maszyny liczące w układzie dwójkowym posiadają około 30—40 znaków binarnych (bitów), co odpowiada 9—12 znakom dziesiętnym.

### 1.5. Arytmometr

Arytmometr jest urządzeniem maszyny, wykonującym operacje arytmetyczne, a także niektóre operacje logiczne. Wyróżnić w nim należy sumator, którego zadaniem jest wykonywanie działań dodawania i odejmowania oraz układ mnożący.

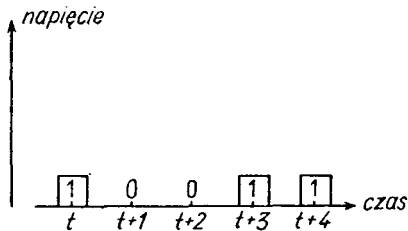
Budowa sumatora i układu mnożącego zależy przede wszystkim od sposobu przesyłania liczb w maszynie. Rozróżniamy dwa takie sposoby. W sposobie szeregowym liczbę przesyła się przy pomocy jednego przewodu cyfra za cyfrą poczynając od pozycji najniższej. Przyjmując, że cyfrze 1 odpowiada impuls elektryczny, cyfrze zaś 0 — brak impulsu, możemy przedstawić szeregowe przesłanie liczby 11001 jak na rys. 11.

W sposobie równoległym przesyłania liczb każdej pozycji odpowiada oddzielny przewód, co zostało pokazane schematycznie (rys. 12) na przykładzie tej samej liczby 11001.

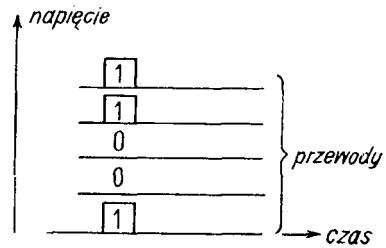
Maszyny o równoległym przesyłaniu liczb pracują szybciej niż maszyny, w których zastosowano przesyłanie szeregowe, są one jednak bardziej skomplikowane pod względem konstrukcyjnym.



Istnieje wiele różnych układów sumacyjnych i mnożących. Przedstawimy tutaj przykładowo budowę sumatora, jaki może być zastosowany w maszynie o równoległym przesyłaniu liczb.



Rys. 11

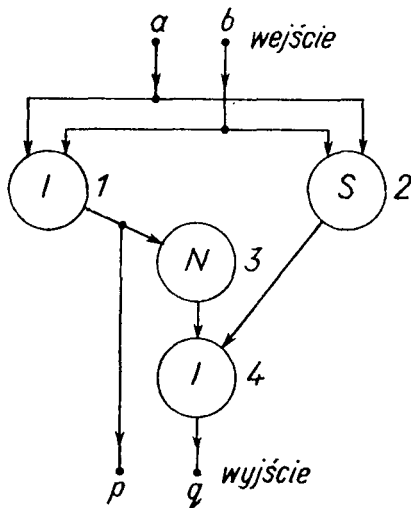


Rys. 12

Rozważmy najpierw zagadnienie sumowania liczb jednocyfrowych. Oznaczmy cyfry składników przez  $a$  oraz  $b$ , cyfry zaś sumy przez  $p$  i  $q$ . Mamy wobec tego:  $a + b = p \cdot 2^1 + q \cdot 2^0$ , przy czym mogą zajść cztery przypadki:

$$a + b = p q$$

- 1)  $0 + 0 = 00$
- 2)  $1 + 0 = 01$
- 3)  $0 + 1 = 01$
- 4)  $1 + 1 = 10$






Rys. 13

W czwartym przypadku mamy do czynienia z przeniesieniem jedynek do pozycji wyższej w stosunku do pozycji sumowanych cyfr.

Na rys. 13 widzimy schemat sumatora jednocyfrowego realizującego działanie  $a + b = pq$ :

Zastosowane tu zostały oznaczenia trzech układów elektronowych, które w określony poniżej sposób kierują przebiegiem impulsów prądu:

| Oznaczenie  | Objaśnienie   |
|---|---|
|  | <p>Impuls pojawia się na wyjściu z tylko wtedy, jeżeli choć na jednym z wejść <math>x, y</math> jest impuls (układ ten nazywamy sumą logiczną).</p>   |
|  | <p>Impuls pojawia się na wyjściu z tylko wtedy, jeżeli jest impuls na wejściu <math>x</math> i na wejściu <math>y</math> (układ ten nazywamy iloczynem logicznym).</p>  |
|  | <p>Impuls pojawia się na wyjściu z tylko wtedy, gdy na wejściu <math>x</math> nie ma impulsu — i odwrotnie, tj. na wyjściu <math>z</math> nie ma impulsu tylko wtedy, gdy na wejściu <math>x</math> jest impuls (układ ten nazywamy negacją).</p> |

Analizując połączenia układów elektronowych na podanym schemacie stwierdzimy <sup>\*)</sup>, że istotnie w każdym z czterech możliwych przypadków spełnione jest równanie  $a + b = pq$ . Szczegółowo opiszemy przypadek czwarty, dla którego  $a = b = 1$ . Przyjmiemy, że jedynie odpowiada impuls prądu oraz, że brak impulsu oznacza zero. Wobec tego na obydwóch przewodach wejścia mamy impulsy prądu dochodzące do układów, które na rys. 13 oznaczono dodatkowo cyframi 1 i 2. Zgodnie z podanym określeniem, z układu 1 wychodzi impuls doprowadzany do wyjścia  $p$ , co oznacza, że  $p = 1$ . Prócz tego impuls ten dochodzi do układu 3 oznaczonego literą  $N$ . Jeżeli do układu  $N$  dochodzi impuls, to na wyjściu  $z$  tego układu brak jest impulsu. Tak więc, do układu 4 nie dochodzi impuls z układu 3, a zatem na wyjściu  $q$  nie ma impulsu, co oznacza, że  $q = 0$ . W ten sposób sprawdziliśmy działanie sumatora jednocyfrowego dla przypadku  $1 + 1 = 10$ .

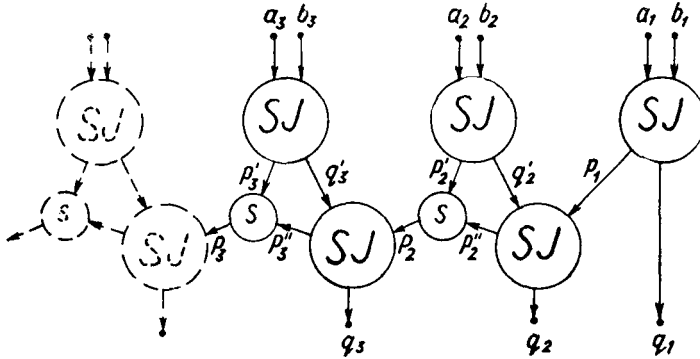
Zajmiemy się teraz sumowaniem liczb wielocyfrowych. Przypuśćmy, że mamy zsumować 2 liczby  $a = \dots a_3 a_2 a_1$  i  $b = \dots b_3 b_2 b_1$  tworzące sumę  $q = \dots q_3 q_2 q_1$  <sup>\*\*)</sup>). Na rys. 14 podano schemat sumatora realizującego takie działanie.

<sup>\*)</sup> Dla uniknięcia nieporozumień przy odczytywaniu schematów należy zaznaczyć, że połączenie między krzyżującymi się przewodami istnieje tylko wtedy, gdy zaznaczono to kropką.

<sup>\*\*)</sup> Litery ze wskaźnikami oznaczają kolejne cyfry liczb  $a, b$  i  $q$ .

Wprowadziliśmy tutaj dodatkowe oznaczenie (rys. 14a), które symbolizuje sumator jednocyfrowy, tj. układ przedstawiony na rys. 13.

W przedstawionym sumatorze każdej pozycji cyfrowej, z wyjątkiem pierwszej, odpowiadają dwa sumatory jednocyfrowe oraz jeden układ oznaczony literą  $S$ . Taka budowa sumatora wynika z konieczności uwzględnienia ewentualnego przeniesienia jedynek z niższych pozycji do wyższych.



Rys. 14



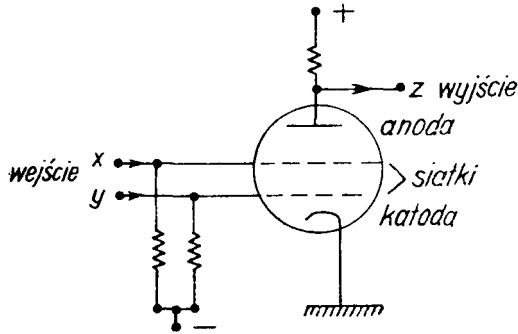
Rys. 14a

W pierwszej pozycji cyfrę sumy  $q_1$  oraz przeniesienia  $p_1$  otrzymujemy bezpośrednio z pojedynczego sumatora jednocyfrowego. Natomiast już w drugiej pozycji analogiczne sumowanie daje nam cyfrę  $q'_2$  i cyfrę przeniesienia  $p'_2$ , które to cyfry mogą ulec zmianie w przypadku, gdy  $p_1 = 1$ . Zadaniem drugiego sumatora jest uwzględnienie jedynek z przeniesienia, to jest utworzenie sumy cyfr  $q'_2$  i  $p_1$  o cyfrach  $p'_2$  (przeniesienie) i  $q_2$  (druga cyfra sumy). Jak z tego widać, przeniesienie do trzeciej pozycji może wyjść z jednego lub drugiego sumatora jednocyfrowego drugiej pozycji; przeniesienia te oznaczyliśmy  $p'_2$  i  $p''_2$ . Łatwo wykazać, że co najwyżej jedno z tych przeniesień może być równe jedności, wobec tego rola układu  $S$  polega tylko na przepuszczaniu impulsu prądu dochodzącego przewodem  $p'_2$  lub  $p''_2$ . W ten sposób otrzymujemy przeniesienie  $p_2$  z pozycji drugiej do trzeciej. Sumowanie w następnych pozycjach cyfrowych przebiega analogicznie.

W podawanych schematach wykorzystywane były trzy rodzaje w różny sposób działających układów elektrycznych. Pokażemy teraz przykładowo, jak można zrealizować przy pomocy lampy elektronowej o dwóch siatkach układ drugiego rodzaju, to jest układ oznaczany literą  $I$  (impuls pojawia się na przewodzie  $z$ , jeżeli jest impuls na przewodzie  $x$  i na przewodzie  $y$ ).

Jak wiadomo, w lampie elektronowej, posiadającej tylko 2 elektrody, a mianowicie dodatnią — anodę i ujemną katodę, pod wpływem przyłożo-

nego napięcia następuje przepływ elektronów z katody do anody. Jeżeli pomiędzy anodę i katodę wstawimy dodatkowo trzecią elektrodę, tzw. siatkę, to przepływ elektronów będzie uzależniony od wartości potencjału siatki względem katody. Przy odpowiednio niskim potencjale siatki przepływ elektronów ulega przerwaniu i — jak się to mówi — lampa zostaje „zatkana”, nie przewodzi prądu. Na rysunku umieszczonym poniżej mamy lampę o dwóch siatkach, do każdej z nich doprowadzone jest na-



Rys. 15

pięcie zatykające. Napięcie jest tak dobrane, aby każda siatka przerywała prąd oddzielnie. Przy braku impulsów (dodatnich) na wejściu lampa nie przewodzi prądu, a napięcie na anodzie równe jest napięciu zasilającemu. Lampa przewodzi prąd wtedy i tylko wtedy, gdy do obu siatek przyłożymy jednocześnie impulsy dodatnie. Wówczas na wyjściu również pojawi się impuls. (Impuls ten będzie ujemny, co zazwyczaj stwarza konieczność stosowania dodatkowej lampy elektronowej zmieniającej znak impulsu).

### 1.6. Pamięć

Pamięć jest jednym z najważniejszych urządzeń maszyny automatycznej. Można ją porównać do szafy z dużą ilością ponumerowanych przegródek. W każdej przegródce może być przechowana jedna liczba lub też jeden rozkaz zapisany jako liczba. (Mając na myśli rozkaz lub liczbę mówimy często w skrócie — słowo). Numer przegródki, zwany adresem, ułatwia zarówno wprowadzanie liczb na właściwe miejsca jak też i późniejsze pobieranie ich z „szafy” — pamięci.

Charakteryzując pamięć mówi się o jej pojemności i czasie oczekiwania. Pojemnością nazywamy ilość przegródek, a więc ilość słów, jaka może być przechowana w pamięci, natomiast przez czas oczekiwania rozumie się przeciętny okres czasu, jaki jest potrzebny dla wprowadzenia lub pobrania liczby z pamięci. Maszyny wyposażone w pamięci o dużej po-

jemności mogą rozwiązywać trudniejsze, bardziej skomplikowane zadania, niedostępne dla tych maszyn, które posiadają mało pojemne pamięci. Tak więc, pojemność pamięci określa skalę zagadnień, do których maszyna może być zastosowana. Istotne znaczenie posiada również czas oczekiwania, ponieważ od wielkości tego czasu w poważnym stopniu zależy szybkość pracy maszyny.

Zbudowanie pamięci posiadającej dużą pojemność i jednocześnie krótki czas oczekiwania nastrocza wiele trudności technicznych. Z tego względu w maszynach elektronowych stosuje się zazwyczaj dwa lub trzy rodzaje wzajemnie się uzupełniających pamięci. W pamięci szybkiej o czasie oczekiwania krótszym od jednej milisekundy (tysięcznej części sekundy) umieszcza się słowa, które są często użytkowane w innych organach maszyny, przede wszystkim w urządzeniu arytmetycznym. Pamięć szybka posiada niewielką pojemność, dodatkowo stosuje się więc pamięć średnio szybką o czasie oczekiwania rzędu kilku lub kilkudziesięciu milisekund. W tej pamięci, posiadającej większą pojemność, umieszcza się słowa użytkowane rzadziej niż słowa z pamięci szybkiej. W niektórych przypadkach stosuje się jeszcze pamięci wolne o bardzo dużej pojemności i długim czasie oczekiwania, nawet rzędu kilku minut.

Istnieje wiele różnorodnych rozwiązań konstrukcyjnych pamięci. Praktyczne zastosowanie znalazły, między innymi, pamięci zbudowane przy użyciu: przekaźników elektromechanicznych, lamp elektronowych, rur rtęciowych, lamp oscyloskopowych i materiałów magnetycznych. Przykładowo opiszemy tutaj pamięci magnetyczne na taśmach i bębnach oraz pamięci magnetyczne na rdzeniach.

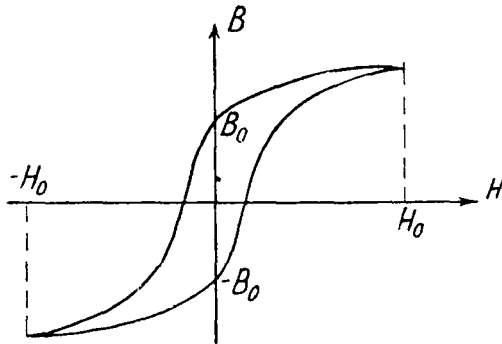
Zasada działania pamięci magnetycznych na taśmach i bębnach jest podobna do zasady działania magnetofonu. Ponad powierzchnią magnetyczną, przesuwaną się z określoną szybkością, znajdują się elektromagnesy, które zgodnie z nadsyłanymi impulsami prądu tworzą namagnesowane odcinki, zwane dipolami magnetycznymi. Odczytanie dokonanego zapisu przebiega w sposób odwrotny. Zmiany pola magnetycznego, powstałe przy przesuwaniu materiału magnetycznego z utrwalonymi dipolami, wywołują odpowiednie siły elektromotoryczne w cewkach elektromagnesów. Siły te są z kolei przetwarzane na standardowe impulsy prądu charakteryzujące odczytywane cyfry.

W przypadku pamięci taśmowej materiał magnetyczny naniesiony jest na elastyczną taśmę przewijającą się z jednej szpulki na drugą. Pamięci taśmowe posiadają wielką pojemność, dochodzącą do setek tysięcy liczb, pracują jednak bardzo wolno, ponieważ czas oczekiwania zależy od stosunkowo długiego czasu przewinięcia taśmy. Znacznie szybsze są pamięci bębnowe (należące do grupy pamięci średnio szybkich), w których ma-

teriał magnetyczny naniesiony jest na walec wirujący z szybkością rzędu 6—7 tys. obrotów na minutę. Wzdłuż tworzącej walca znajduje się szereg umieszczonych obok siebie parami elektromagnesów. Każdej parze, składającej się z elektromagnesu zapisującego i elektromagnesu odczytującego (często dla zapisu i odczytu stosuje się ten sam elektromagnes), odpowiada na materiale magnetycznym tzw. ścieżka zapisu, utworzona przez dipole, które zostały utrwalone w czasie obrotu walca. Gęstość zapisu na takiej ścieżce dochodzi do kilkudziesięciu bitów na 1 cm. Szerokość ścieżki wynosi kilka milimetrów. Pojemność pamięci bębnowych jest rzędu kilku tysięcy liczb, czas zaś oczekiwania wynosi kilka milisekund.

W ostatnim czasie coraz częściej znajdują zastosowanie pamięci zbudowane z materiałów ferromagnetycznych o tzw. prostokątnej pętli histerezy.

Jak wiadomo, pętlą histerezy nazywamy krzywą przedstawiającą zmienność indukcji magnetycznej  $B$  w materiale ferromagnetycznym (żelazo, nikiel, kobalt) w zależności od zmienności natężenia pola magnetycznego  $H$ . Wyobraźmy sobie pierścień żelazny, na który została nawinięta pewna liczba zwojów drutu. Przepuszczając przez ten drut prąd o regulowanym natężeniu  $I$  będziemy mogli wytwarzać zmienne natężenie pola magnetycznego  $H$ . Zwiększając natężenie pola przy pewnej jego wartości  $H = H_0$  osiągniemy stan nasycenia; z kolei zmniejszając natężenie pola przy  $H = 0$  otrzymamy indukcję  $B_0$  oraz przy  $H = -H_0$  osiągniemy po raz

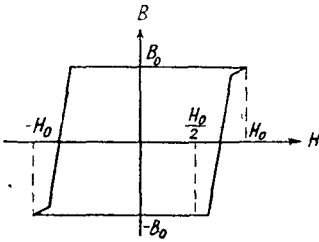


Rys. 16

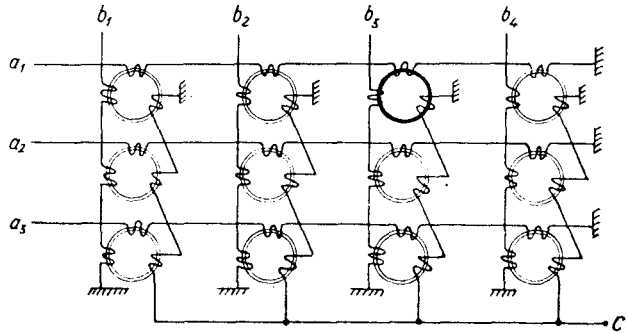
drugi stan nasycenia, tym razem dla ujemnych wartości  $B$ . Zwiększając powtórnie natężenie, przy  $H = 0$  otrzymamy indukcję  $-B_0$ . Tak więc indukcja nie zależy tylko od wartości natężenia pola, ale także od tego, jaki był poprzednio stan nasycenia materiału. Wartości indukcji  $\pm B_0$  nazywa się pozostałością magnetyczną. Dzięki tej właśnie indukcji kawałek stali pozostaje trwałym magnesem.

Niektóre materiały posiadają pętle histerezy o kształcie zbliżonym do pokazanego na rys. 17. Jest to tzw. prostokątna pętla histerezy.

Rdzenie magnetyczne, sporządzone z materiału o prostokątnej pętli histerezy, posiadają pewną, jak się okaże dalej, istotną właściwość, która umożliwi użycie ich do budowy pamięci. Właściwość ta polega na tym, że namagnesowany rdzeń, posiadający pozostałość magnetyczną np.  $-B_0$ , może być przemagnesowany pod wpływem natężenia pola  $H$  nie mniejszego od  $H_0$ , w szczególności natężenie  $H = \frac{H_0}{2}$  nie zmienia stanu magnetycznego rdzenia.



Rys. 17



Rys. 18

Pojedynczy rdzeń może być użyty do zapamiętania jednej cyfry binarnej. Cyfrze 0 przyporządkowuje się jeden stan namagnesowania, cyfrze 1 — drugi. Rdzenie wchodzące w skład pamięci tworzą układy połączone szeregowo trzema rodzajami uzwojeń (rys. 18). Uzwojenia poziome  $a$  i pionowe  $b$  są uzwojeniami pierwotnymi, służą one do zapisywania cyfr; pozostałe uzwojenia o wspólnym wyjściu  $c$  są uzwojeniami wtórnymi, służącymi do odczytywania. Jeżeli przyjmiemy, że w jednym wierszu zapisujemy jedną liczbę, to narysowany układ pozwala na zapamiętanie trzech czterocyfrowych liczb.

Pokażemy teraz na przykładzie, w jaki sposób odbywa się utrwalanie liczb w pamięci zbudowanej przy użyciu rdzeni magnetycznych. Przypuśćmy, że mamy utrwalić trzecią cyfrę pierwszej liczby równą jedności oraz przypuśćmy, że cyfrze 1 odpowiada indukcja  $B_0$ . Dla utrwalenia tej cyfry przez uzwojenie  $a_1$  i  $b_3$  przepuścimy impulsy prądu powodujące powstanie pola magnetycznego o natężeniu  $\frac{H_0}{2}$ . W rdzeniu odpowiadającym trzeciej cyfrze pierwszej liczby, przez który przechodzą obydwa uzwojenia ( $a_1$  i  $b_3$ ), będziemy mieli natężenie  $H_0$  i rdzeń zostanie namagnesowany. Stan magnetyczny pozostałych rdzeni nie ulegnie zmianie, ponieważ natę-

żenie  $\frac{H_0}{2}$  nie wystarcza dla przemagnesowania rdzenia. Przy zapisie cyfry dokonujemy jednocześnie odczytu poprzedniego zapisu. Jeżeli bowiem w omawianym przykładzie stan rdzenia charakteryzował się indukcją  $-B_0$ , to proces przemagnesowania przy zapisie wywoła pojawienie się impulsu prądu w uzwojeniu wtórnym c, w przeciwnym przypadku impulsu tego nie będzie.

Czas oczekiwania w pamięci zbudowanej na rdzeniach magnetycznych wynosi kilka  $\mu$  sek.

### 1.7. Wejście i wyjście

Powszechne zastosowanie znalazły dwa sposoby wprowadzania i pobierania danych w maszynie cyfrowej: przy pomocy kart dziurkowanych oraz przy pomocy taśm dziurkowanych (perforowanych). Na rys. 19 przedstawiono kartę dziurkowaną o 80 kolumnach. W każdej kolumnie znajduje się 12 miejsc na otwory; dziesięć spośród nich jest przeznaczonych dla wydziurkowania jednej z dziesięciu cyfr systemu dziesiętnego: 0,1...9. Dwa pozostałe miejsca, znajdujące się ponad rzędem zer, służą do dziurkowania otworów posiadających specjalne znaczenie (patrz np. str. 71).

Karty zawierające dane liczbowe niezbędne dla rozwiązania zadanego maszynie problemu dziurkuje się ręcznie przy pomocy urządzenia zwanego dziurkarką, natomiast wyniki są dziurkowane przez maszynę automatycznie.

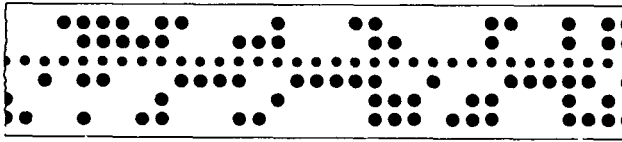
Odczytywanie kart przez maszynę odbywa się zazwyczaj na drodze elektromechanicznej: karta przechodzi pod układem szczotek, które ślizgają się wzdłuż kolumn i w przypadku natrafienia na otwór powodują zamknięcie obwodu elektrycznego. Powstały przy tym impuls elektryczny może zająć w jednym z dwunastu momentów odpowiadającym dziesięciu cyfrom 0,1,...9 oraz dwum sygnałom specjalnym. Ten system odczytywania kart dostatecznie szybki dla maszyn licząco-analitycznych okazał się zbyt wolny dla maszyn elektronowych. Dlatego też wprowadzono fotoelektryczny system odczytywania. W tym sposobie kartę przesuwają się nad układem fotokomórek dających impulsy prądu w momentach, w których bezpośrednio ponad fotokomórkami znajdują się wydziurkowane otwory przepuszczające z zewnątrz wiązki światła.

Obydwa sposoby odczytywania stosuje się również przy taśmach dziurkowanych. Na rys. 20 pokazany został fragment taśmy dalekopisowej. Cyfry przedstawia się tutaj przy pomocy kombinacji otworów dziurkowanych w pięciu równoległych ścieżkach zapisu (szósta ścieżka utworzona z mniejszych otworów przeznaczona jest do przesuwania taśmy). Oprócz wejść dostosowanych do kart lub taśm dziurkowanych maszyny





automatyczne na ogół posiadają klawiatury umożliwiające ręczne wprowadzenie danych. Wyniki często drukuje się od razu w postaci gotowych tabel liczbowych.



Rys. 20

Wprowadzanie rozkazów do maszyny nie różni się niczym od wprowadzania liczb, ponieważ rozkazy mają postać liczbową.

### 1.8. Krótki opis maszyny XYZ

Ponieważ w dalszym ciągu będzie mowa o programowaniu przede wszystkim dla polskiej maszyny XYZ, podamy najpierw krótki opis tej maszyny.

XYZ jest maszyną szeregową \*) stałoprzecinkową, dwójkową. Pojemność pamięci wynosi 1024 słowa 18 bitowe (17 cyfr oraz znak) lub 512 słów 36 bitowych (35 cyfr oraz znak). Pamięcią są 32 rury stalowe napelnione rtęcią. Uruchamiany jest obecnie (początek 1960 roku) bęben magnetyczny jako pamięć dodatkowa o pojemności około 8000 słów 36 bitowych. Średnia szybkość — około 800 operacji na sekundę (wliczając czas oczekiwania). Wejście i wyjście na 80 kolumnowych kartach dziurkowanych.

Urządzeniem czytającym i dziurkującym jest standardowy reproducer \*\*) firmy Bull, przystosowany w ZAM do współdziałania z maszyną cyfrową, o maksymalnej przepustowości około 7000 kart na godzinę. Reprodicer ten będzie zastąpiony w krótkim czasie bardzo szybkim urządzeniem na taśmę dziurkowaną, budowanym specjalnie do współdziałania z maszyną cyfrową.

Oprócz tych danych, charakteryzujących maszynę, konieczne jest — dla wyjaśnienia niektórych jej czynności — opisanie tzw. rejestrów. Rejestry są elementami pamięciowymi — różna ich rola zależy wyłącznie od miejsca w schemacie maszyny. Są to elementy technicznie wyodrębnione od pamięci; w XYZ — nie uwzględnione w numeracji miejsc pamięci.

\*) Maszyna szeregową — maszyna o szeregowym przesyłaniu informacji (patrz str. 49).

\*\*) Reprodicer — urządzenie służące do przenoszenia informacji z karty na kartę.

(W niektórych maszynach rejestry wchodzą w skład pamięci). XYZ posiada następujące rejestry: akumulator (rejestr akumulatora — **A**), mnożnik (rejestr mnożnika — **M**), rejestr rozkazów, rejestr nadmiaru oraz licznik rozkazów. Istnieje ponadto rejestr przejściowy (**P**), spełniający rolę wyłącznie techniczną.

Rejestr **A** i **M** wchodzą w skład arytmometru i służą do magazynowania danych i wyników operacji aktualnie wykonywanych przez arytmometr. A więc, dodajemy (odejmujemy) liczby w ten sposób, że jedną z nich umieszczamy za pomocą rozkazu w akumulatorze, rozkaz dodawania (odejmowania) powoduje pobranie drugiej liczby z pamięci i przeprowadzenie ich obu przez urządzenie zwane sumatorem; wynik otrzymuje się w akumulatorze.

Przy mnożeniu jeden z czynników umieszczamy w mnożniku, drugi — jest pobierany z pamięci; mnożenie arytmometr wykonuje przez kolejne sumowanie, a wynik umieszcza w **A** i **M** traktując je jako jeden rejestr o pojemności 70 bitów.

Rejestr rozkazów zawiera każdorazowo rozkaz, który ma być wykonany, w liczniku rozkazów znajduje się natomiast numer miejsca pamięci, z którego ma być pobrany następny rozkaz. Oba te rejestry wchodzą w skład urządzenia sterującego.

## 2. Programowanie

### 2.1. Wstęp

Programem nazwiemy tu opis algorytmu realizowanego przez jakieś urządzenie (w naszym wypadku przez maszynę cyfrową). Opis ten musi być przy tym zapisany w sposób umożliwiający realizację.

W przypadku maszyny cyfrowej — o której wyłącznie w dalszym ciągu będziemy mówili — program jest realizowany automatycznie, bez interwencji człowieka w trakcie działania maszyny.

Automatyczna realizacja programu wymaga, aby był on opracowany bardzo starannie i szczegółowo. Najdrobniejsza niedokładność w zapisie programu uniemożliwia poprawne wykonanie działania.

Zapisu programu dla maszyny cyfrowej dokonujemy za pomocą rozkazów.

Układ (listę) rozkazów maszyny wraz ze sposobem korzystania z nich nazywa się językiem maszyny. Programowanie możemy więc określić jako tłumaczenie z języka wyrazów i wzorów na język maszyny.

Ze względu na to, że poszczególne rozkazy odpowiadają czynnościom bardzo elementarnym, układanie programu trwa dosyć długo i wymaga dużej uwagi.

Czas przygotowywania programu wielokrotnie przewyższa czas wykonywania go przez maszynę, a ułożony już program z reguły wymaga wielu poprawek.

Aby ułatwić i przyspieszyć układanie programów, niemal w każdym ośrodku używającym maszyn cyfrowych opracowuje się mniej lub więcej rozwinięty system pomocniczy, przerzucający część pracy programującego na maszynę.

Systemy takie używają programu lub zespołu programów, które zwiększają „słownik” języka maszyny oraz zbliżają ten język do języka formuł matematycznych.

Ze względu na używanie lub nie używanie systemów pomocniczych podzielimy programowanie na:

- a) programowanie bezpośrednie w języku maszyny,
- b) programowanie z programem pierwotnym,
- c) programowanie automatyczne.

W dalszym ciągu omówimy krótko powyższe systemy programowania.

Maszyna posiada ustaloną listę rozkazów, z których każdy składa się zasadniczo z dwóch części: z symbolu (numeru) operacji oraz z jednego lub większej ilości numerów miejsc pamięci, do których odnosi się dany rozkaz. Numery miejsc pamięci w rozkazach zwane są adresami. W zależności od tego czy w każdym rozkazie danej maszyny jest jedna liczba adresowa, czy więcej — dzielimy maszyny na jednoadresowe i wieloadresowe. Dla wyjaśnienia tych nazw podamy przykłady organizacji rozkazów \*) w maszynach jedno i wieloadresowych.

a) Rozkaz maszyny jednoadresowej składa się z symbolu operacji i adresu liczby, na której ma być wykonana operacja. Przy operacjach dwuelementowych, jak np. dodawanie (jeżeli operacja obejmuje dwie liczby, mówimy, że jest dwuelementowa), jeden z elementów musi znajdować się w miejscu ustalonym, którego adresu nie potrzeba umieszczać w rozkazie. Wynik operacji znajduje się w ustalonym miejscu pamięci lub w rejestrze.

Dalej przytoczymy przykład jednoadresowej listy rozkazów.

b) Rozkaz maszyny dwuadresowej zawiera symbol operacji oraz adresy dwu elementów operacji lub adresy elementu operacji i wyniku.

c) Trójadresowy rozkaz zawiera symbol operacji oraz adresy obu elementów operacji i wyniku. Jeżeli operacja jest jednoelementowa — element ten jest określany przez ustalony adres, np. przez pierwszy; jeden z adresów nie ma wówczas żadnego znaczenia.

---

\*) Organizacja rozkazu — wzajemny układ i przeznaczenie poszczególnych części rozkazu.

Oczywiście organizacja rozkazów wieloadresowych może być różna od podanych wyżej (patrz [8]). Maszyny z większą od trzech ilością adresów w rozkazie budowane są rzadko. (W Czechosłowacji zbudowano maszynę pięcioadresową, zwaną SAPO).

W dalszym ciągu będziemy zajmowali się wyłącznie listami rozkazów jednoadresowymi jako najprostszymi. W Polsce budowane są dotychczas wyłącznie maszyny jednoadresowe.

Dla ułatwienia będziemy operowali konkretną listą rozkazów maszyny XYZ, zbudowanej w Zakładzie Aparatów Matematycznych PAN, a eksploatowanej przez Biuro Obliczeń i Programów.

| Symbol liczbowy | Symbol literowy*)<br>związany z systemem zewnętrznym zapisu | Nazwa<br>symbol opisowy | Rozkaz powoduje   |
|-----------------|---|-------------------------|---|
| 1               | 2   | 3                       | 4   |
| 0               | G   | Stop $n$                | Zatrzymanie się maszyny (maszyna przestaje wykonywać rozkazy) i przesłanie liczby $n$ do licznika rozkazów, w którym każdorazowo znajduje się adres następnego rozkazu. Po uruchomieniu maszyny, jako pierwszy rozkaz zostanie pobrane słowo**) z miejsca pamięci o numerze $n$ . |
| 1               | I   | Wykonaj $n$             | Pobranie słowa z miejsca $n$ ***) i przesłanie go do rejestru rozkazów; zawartość licznika rozkazów wzrasta o 1. Rozkaz znajdujący się w rejestrze rozkazów zostaje natychmiast wykonany.   |
| 2               | S   | Skocz $n$               | Przesłanie liczby $n$ do licznika rozkazów. (Jako następny rozkaz zostanie pobrane słowo z miejsca $n$ ).   |
| 3               | T   | Skocz $+ n$             | Jeżeli w rejestrze akumulatora znajduje się liczba ze znakiem $+$ , działanie jak 2. W przeciwnym wypadku rozkaz nieefektywny.  |
| 4               | Z   | Skocz $0 n$             | Jeżeli w rejestrze akumulatora znajduje się liczba 0 — działanie jak 2. W przeciwnym wypadku rozkaz nieefektywny.   |

\*) Symbol literowy przyjęto dla wygody, jest on tłumaczony przez maszynę na symbol liczbowy (rubryka 1).

\*\*) Ponieważ rozkazy mają w maszynie postać liczb, dla oznaczenia jednych i drugich używa się nazwy „słowo”.

\*\*\*) Miejsce  $n$  — miejsce o numerze  $n$ .

| 1  | 2 | 3                     | 4  |
|----|---|-----------------------|--|
| 5  | V | Skocz $V n$           | Jeżeli w rejestrze nadmiaru znajduje się jedynka — działanie jak 2 i zerowanie rejestru nadmiaru. W przeciwnym wypadku rozkaz nieefektywny.<br><br>(Rejestr nadmiaru jest rejestrem jednocyfrowym, w którym pojawia się jedynka, jeżeli moduł liczby w akumulatorze jest większy od jedności. Zerowanie tego rejestru powoduje rozkaz 5).          |
| 6  | D | Dodaj $(d)^* n$       | Do liczby znajdującej się w akumulatorze dodanie liczby z miejsca $n$ .  |
| 7  | R | Cdejmij $(d) n$       | Od liczby znajdującej się w akumulatorze odjęcie liczby z miejsca $n$ .  |
| 8  | B | Cdejmij bezw. $(d) n$ | Od wartości bezwzględnej liczby znajdującej się w akumulatorze odjęcie wartości bezwzględnej liczby z miejsca $n^{**}$ .   |
| 9  | X | Mnóż $(d) n$          | Pomnożenie liczby znajdującej się w rejestrze mnożnika ( <b>M</b> ) przez liczbę z miejsca $n$ . Wynik jako liczba 70 bitowa znajduje się w <b>A</b> (część więcej znacząca) i w <b>M</b> (część mniej znacząca). Poprzednia zawartość <b>A</b> zostaje wymazana.  |
| 10 | Q | Dziel $(d) n$         | Podzielenie liczby 70 bitowej znajdującej się w <b>A</b> i <b>M</b> (tak jak w wyniku mnożenia) przez liczbę z miejsca $n$ . Wynik znajduje się w <b>M</b> ; w <b>A</b> — reszta. Liczba w $n$ musi być większa od liczby w <b>A</b> i <b>M</b> w przeciwnym wypadku pojawia się jedynka w rejestrze nadmiaru, a wynik dzielenia jest bezsensowny. |
| 11 | U | Umieść $(d) n$        | Umieszczenie w <b>A</b> liczby z miejsca $n$ ; poprzednia zawartość <b>A</b> zostaje wymazana.   |
| 12 | N | Naładuj $(d) n$       | Umieszczenie w <b>M</b> liczby z miejsca $n$ ; poprzednia zawartość zostaje wymazana.  |
| 13 | A | Zapamiętaj $A (d) n$  | Przesłanie liczby zawartej w <b>A</b> do miejsca $n$ , którego poprzednia zawartość zostaje wymazana. Zawartość <b>A</b> bez zmian.  |
| 14 | M | Zapamiętaj $M (d) n$  | Podobnie jak 13, lecz w odniesieniu do <b>M</b>  |
| 15 | C | Zaokrąglij            | Dodanie do liczby w <b>A</b> i <b>M</b> liczby $2^{-36}$ (tzn. jedynki na najwięcej znaczącej pozycji <b>M</b> ), na najmniej znaczącej pozycji <b>M</b> pojawia się jedynka. Adres w tym rozkazie jest obojętny.  |

\*) Znaczenie symbolu  $(d)$  będzie wyjaśnione niżej.\*\*) Wyniki po wykonaniu rozkazów 6, 7 i 8 znajdują się w **A** (patrz 1.8).

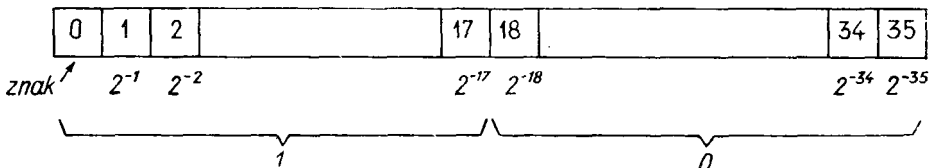
| 1  | 2 | 3                      | 4  |
|----|---|------------------------|--|
| 16 | L | Posuń w lewo $n$       | Pomnożenie liczby w <b>A</b> i <b>M</b> przez $2^n$ , znak <b>A</b> taki sam jak <b>M</b> *).  |
| 17 | P | Przesuń w prawo $n$    | Pomnożenie liczby w <b>A</b> i <b>M</b> przez $2^{-n}$ , znak <b>M</b> taki sam jak <b>A</b> .   |
| 20 | J | Czytaj $n$             | Uruchomienie urządzenia czytającego (wejścia) i przesłanie pierwszego słowa pojawiającego się na wejściu do miejsca $n$ .  |
| 21 | W | Pisz $n$               | Uruchomienie urządzenia dziurkującego (wyjścia) i przesłanie do niego liczby z miejsca $n$ .   |
| 30 | K | Czytaj klucze $n$      | Przeniesienie do miejsca $n$ liczby, która została zakodowana przy pomocy kluczy na stoliku operatora.   |
| 31 | H | Koniunkcja ( $d$ ) $n$ | Logiczne wymnożenie liczby w <b>M</b> przez liczbę z $n$ . Wynik w <b>A</b> . (Koniunkcja poszczególnych cyfr — jeżeli na określonej pozycji czynników znajdowały się cyfry 1 i 1 w wyniku otrzymujemy 1. Przy pozostałych kombinacjach otrzymujemy zero). |

Działanie rozkazów 6, 7, 10, 15 i 16 może powodować przekroczenie zakresu, a więc i pojawienie się jedyńki w rejestrze nadmiaru. Rozkazy z symbolem ( $d$ ) mogą dotyczyć zarówno słów krótkich (17 bitów oraz znak), jak i długich (35 bitów oraz znak). Decyduje o tym określona pozycja w rozkazie: jeżeli znajduje się na nim jedyńka — rozkaz odnosi się do słowa długiego, jeżeli zero — do krótkiego. W pozostałych rozkazach zawartość tej pozycji jest obojętna.

Wszystkie działania arytmetyczne maszyna wykonuje uwzględniając znak.

Słowo długie obejmuje po prostu dwa słowa krótkie umieszczone obok siebie w pamięci; słowa długie są numerowane liczbami parzystymi.

Organizacja słowa długiego (liczby):



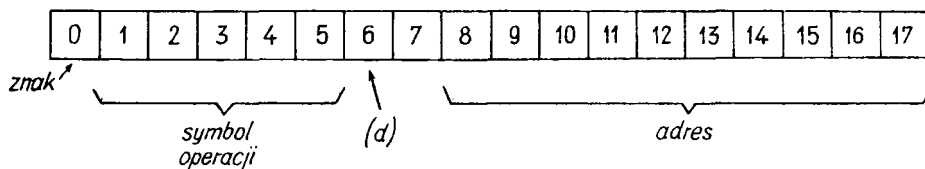
Rys. 21

W liczbie długiej część więcej znacząca znajduje się w miejscu o numerze nieparzystym, część mniej znacząca w miejscu o numerze parzystym. (Numeracja miejsc w pamięci rozpoczyna się od zera).

\*) Tzn. znak **M** nie ulega zmianie, znak **A** zostaje utożsamiony ze znakiem **M**.

Organizacja słowa krótkiego w maszynie:

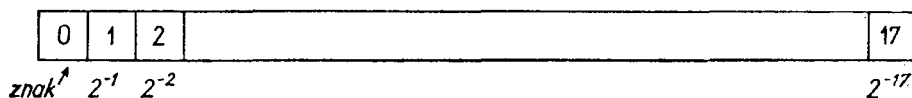
a) interpretowanego jako rozkaz:



Rys. 22

W rozkazie pozycja 7 nie ma wpływu na działanie maszyny.

b) interpretowanego jako liczba:



Rys. 23

Rozkaz w maszynie możemy interpretować jako liczbę:

$$\pm (N \cdot 2^{-5} + d \cdot 2^{-6} + a \cdot 2^{-17})$$

$N$  jest symbolem liczbowym operacji,  $d$  — zerem lub jedyneką,  $a$  — adresem ( $0 \leq a \leq 1023$ ).

Znak rozkazu nie ma wpływu na jego działanie.

Obecnie możemy przejść już do opisu samego programowania.

## 2.2. Programowanie bezpośrednie

*a. Programy liniowe.* (Należy odróżnić układanie programów liniowych od tzw. programowania liniowego, które jest określeniem powstałym na gruncie ekonometrii, gdzie nazwa programowania ma inny sens). Programem liniowym nazwiemy program, w którym są kolejno zapisane dokładnie wszystkie działania, których wymaga rozwiązanie postawionego zagadnienia. Najlepiej wyjaśni to przykład.

Należy wykonać sumomnożenie o stu składnikach:

$$\sum_{i=1}^{100} a_i b_i; \text{ liczby (krótkie) } a_i, b_i \text{ są umieszczone odpowiednio w miejscach}$$

600 ÷ 699, i 500 ÷ 599. Wynik umieścić w miejscu 700 — (długim).

Program będzie wyglądał następująco:

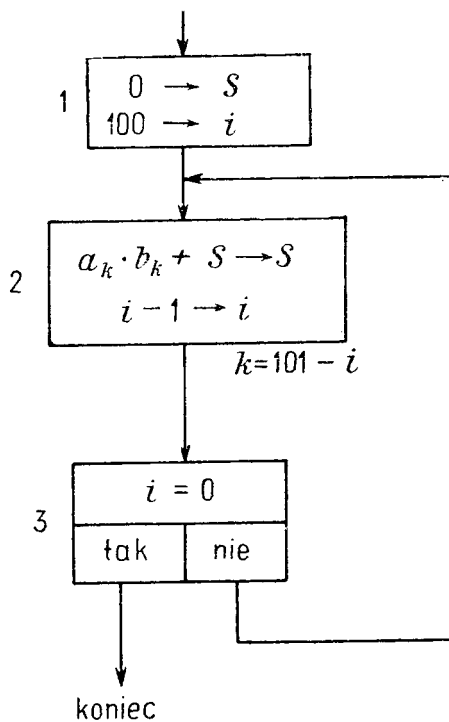
- |   |     |     |     |  |
|---|-----|-----|-----|--|
| 0 | $N$ | $k$ | 600 | Do rejestru mnożnika zostaje przesłana liczba (krótka) z miejsca 600 pamięci.                |
| 1 | $X$ | $k$ | 500 | Liczba znajdująca się w rejestrze $M$ zostaje pomnożona przez liczbę z miejsca 500 (krótka). |



|     |   |   |     |   |
|-----|---|---|-----|---|
| 2   | A | d | 700 | Wynik mnożenia znajdujący się w akumulatorze (jest to liczba długa) zostaje przesłany do miejsca 700. |
| 3   | N | k | 601 | Do M zostaje przesłana następująca liczba.  |
| 4   | X | k | 501 | Mnożenie.   |
| 5   | D | d | 700 | Dodanie wyniku pierwszego mnożenia.   |
| 6   | A | d | 700 | Zapamiętanie sumy częściowej.   |
| ⋮   | ⋮ | ⋮ | ⋮   |   |
| 395 | N | k | 699 |   |
| 396 | X | k | 599 |   |
| 397 | D | d | 700 |   |
| 398 | A | d | 700 |   |

Jak widać program tego prostego zagadnienia jest bardzo długi, nie wykorzystuje on jednak tak istotnej cechy automatycznych maszyn cyfrowych jak rozkazy skokowe i warunkowe.

Zauważmy, że zagadnienie to można zapisać w postaci schematu blokowego:



Rys. 24

W „skrzynkach” schematu umieszczamy krótki, symboliczny opis czynności, które ma wykonać program. Rozmieszczenie skrzynek i połączenia

między nimi uzależnia organizację programu. To znaczy maszyna wykonuje czynności opisane w danej skrzynce i następnie przechodzi do wykonywania czynności tej skrzynki, do której prowadzi strzałka.

W naszym przypadku schemat należy rozumieć w sposób następujący:

Skrzynka 1. W miejscu  $S$  umieszczamy zero; w miejscu  $i$  umieszczamy 100 (będzie to licznik ilości iloczynów, które mają być obliczone).

Skrzynka 2. Obliczamy iloczyn  $a_k \cdot b_k$ , dodajemy do niego  $S$  i sumę przesyłamy na miejsce  $S$ .

Skrzynka 3. Sprawdzamy czy spełniona jest równość  $i = 0$ . Jeżeli nie — wracamy do skrzynki 2. Jeżeli tak — koniec obliczeń, tzn.  $S = \sum_{k=1}^{100} a_k \cdot b_k$

(Pomijamy tu zagadnienie wprowadzania danych i programów, będzie ono omówione dalej). W ten sposób przechodzimy do następnego typu programów.

*b. Programy z pętlami.* Termin „pętla” wywodzi się z graficznego przedstawienia programu w postaci schematu blokowego (zamiast „schemat blokowy” używa się czasami angielskiej nazwy „flowdiagram”).

Na rys. 24 skrzynki 2 i 3 są zamknięte w pętli, co oznacza, że operacje w nich opisane są wykonywane wielokrotnie.

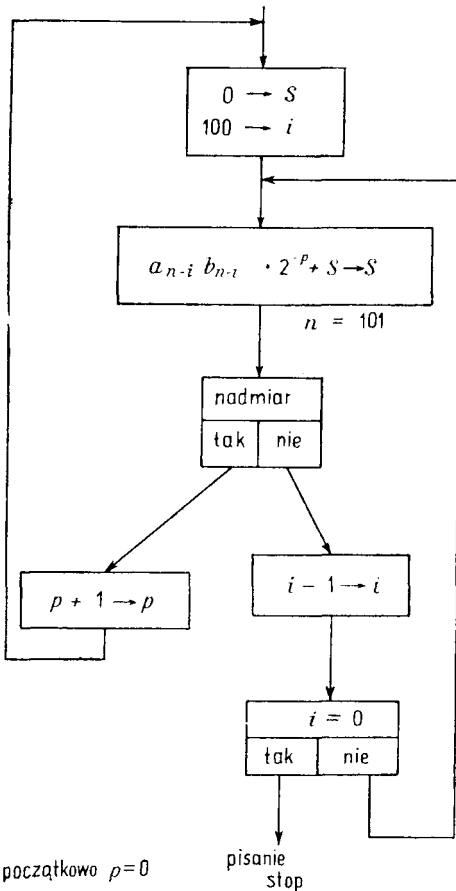
Aby przepisać w języku rozkazów podany wyżej schemat, musimy mieć w pamięci oprócz liczb  $a_i$  oraz  $b_i$  także liczbę 100; 1; 0. Załóżmy, że liczby te znajdują się odpowiednio w miejscach 992, 993, 994.

Program teraz zapiszemy następująco:

|   |    |   |   |     |
|---|----|---|---|-----|
|   | 0  | U | k | 994 |
|   | 1  | A | d | 700 |
|   | 2  | U | k | 992 |
|   | 3  | A | k | 995 |
|   | 4  | N | k | 600 |
|   | 5  | X | k | 500 |
|   | 6  | D | d | 700 |
|   | 7  | A | d | 700 |
|   | 8  | U | k | 4   |
| 2 | 9  | D | k | 993 |
|   | 10 | A | k | 4   |
|   | 11 | U | k | 5   |
|   | 12 | D | k | 993 |
|   | 13 | A | k | 5   |
|   | 14 | U | k | 995 |
|   | 15 | R | k | 993 |
| 3 | 16 | T | k | 3   |

W rozkazach 8÷13 traktowaliśmy rozkazy 4 i 5 jako liczby. Oczywiście liczby 100 i 1 zapisane są w maszynie jako  $100 \cdot 2^{-17}$  i  $1 \cdot 2^{-17}$ . Dodając więc do rozkazu *N k* 600 liczbę  $2^{-17}$  otrzymujemy rozkaz *N k* 601.

Zapis programu w postaci schematu blokowego jest bardzo wygodny i przejrzysty, a przejście do języka rozkazów jest formalnym przepisywaniem zawartości kolejnych „skrzynek” na ten język. Można również dla oszczędności miejsca korzystać w następnej skrzynce z rozkazów skrzynki poprzedniej (w naszym przypadku rozkaz nr 3).



Rys. 25

W przypadku programu z pętlą program składa się zaledwie z 17 rozkazów oraz z 4 miejsc pomocniczych, w przypadku programu liniowego otrzymaliśmy 399 rozkazów. Jednakże wykonanie programu liniowego wymaga wykonania dokładnie 399 rozkazów, a więc w maszynie XYZ trwa to około pół sekundy, natomiast wykonanie programu z pętlą wymaga wykonania 1304 rozkazów, a więc zajmuje trzykrotnie więcej czasu. Czasami może więc opłacać się stosować (fragmentami) programy liniowe ze względu na czas pracy maszyny. Większość programów nie daje się oczywiście w całości zapisać w postaci liniowej ze względu na ograniczoną pojemność pamięci i prędkość wypisywania wielkiej ilości rozkazów.

Stosowanie programów z pętlami jest możliwe dzięki temu, że rozkazy mają postać liczb i istnieją rozkazy warunkowe. Te cechy w sposób istotny wyróżniają ma-

szyny cyfrowe spośród innych przyrządów matematycznych. Na przykładzie pokazanym wyżej można zauważyć, że program może zawierać więcej niż jedną pętlę, mogą one również łączyć się ze sobą w różny sposób.

Zauważmy, że w naszym przypadku w trakcie liczenia może nastąpić przekroczenie zakresu, o ile nie jest zagwarantowane spełnienie nierów-

ności  $|a_i| < \frac{1}{10}$ ,  $|b_i| < \frac{1}{10}$ ;  $i = 1, 2, \dots, 100$ . W takim wypadku schemat

Program ulega więc niewielkim zmianom, po każdorazowym przekro-  
programu będzie wyglądał jak na rys. 25.

czeniu zakresu nastąpi podwyższenie  $p$  o jeden, a więc otrzymamy zamiast  
prawdziwej wartości sumoiloczynu, wartość pomnożoną przez  $2^{-p}$ . Liczbę  
 $p$  możemy odczytać z odpowiedniego rozkazu (każąc go np. wypisać).

|    |   |   |     |   |
|----|---|---|-----|---|
| 0  | U | k | 994 |   |
| 1  | A | d | 700 |   |
| 2  | U | k | 992 |   |
| 3  | A | k | 995 |   |
| 4  | N | k | 600 |   |
| 5  | X | k | 500 |   |
| 6  | P | k | 0   | „Licznik” przesunąć                               |
| 7  | D | d | 700 |   |
| 8  | A | d | 700 |   |
| 9  | V | k | 22  |   |
| 10 | U | k | 4   |   |
| 11 | D | k | 993 |   |
| 12 | A | k | 4   |   |
| 13 | U | k | 5   |   |
| 14 | D | k | 993 |   |
| 15 | A | k | 5   |   |
| 16 | U | k | 995 |   |
| 17 | R | k | 993 |   |
| 18 | T | k | 3   |   |
| 19 | W | k | 6   | } Wypisanie sumy i licznika przesunąć             |
| 20 | W | d | 700 |   |
| 21 | G | k | 0   |   |
| 22 | U | k | 6   |   |
| 23 | D | k | 993 |   |
| 24 | A | k | 6   |   |
| 25 | U | k | 4   | } Regeneracja adresu początkowego<br>w rozkazie 4 |
| 26 | D | k | 995 |   |
| 27 | R | k | 992 |   |
| 28 | A | k | 4   |   |
| 29 | U | k | 5   | } Regeneracja adresu początkowego<br>w rozkazie 5 |
| 30 | D | k | 995 |   |
| 31 | R | k | 992 |   |
| 32 | A | k | 5   |   |
| 33 | S | k | 0   |   |

Podany powyżej przykład może być częścią składową większego programu, np. programu podstawiającego rozwiązanie do układu równań liniowych lub innego i traktowany jako podprogram (po wprowadzeniu pewnych zmian).

Podprogramem nazywamy program, który może być wykorzystywany w ramach jakiegoś większego programu, możliwie bez żadnych zmian. Aby stosować podprogramy nie dokonując w nich żadnych zmian, a więc przede wszystkim zmian adresów, musimy zastosować pewien system, który sprowadza wstawienie podprogramu do mechanicznego dołączenia go, np. na kartach dziurkowanych, do kart pozostałej części.

Stosowanie takiego systemu umożliwia tzw. program pierwotny.

### 2.3. Program pierwotny

Nie będziemy tu opisywać budowy ani działania wewnętrznego programu pierwotnego. Zostaną omówione pokrótce jedynie zasady działania takiego programu i korzyści stąd płynące.

Od programu pierwotnego żądamy, aby wprowadzał do pamięci programy i dane liczbowe, pozwalał na wprowadzenie dowolnego słowa lub grupy słów w dowolne miejsca pamięci oraz na numerowanie słów w każdym programie i podprogramie niezależnie. Ostatnie żądanie oznacza, aby np. podany powyżej program przykładowy można było bez żadnych zmian w adresach zapisać nie tylko od miejsca zerowego w pamięci, ale od miejsca 200 (lub dowolnego innego), a liczby  $a_i$  oraz  $b_i$  odpowiednio od 700 i 800 bez żadnych zmian w zapisie.

Musi istnieć ponadto możliwość rozpoczęcia wykonywania programu od dowolnego miejsca.

Zadania te są realizowane przez wprowadzenie „rozkazów zewnętrznych” oraz adresów względnych.

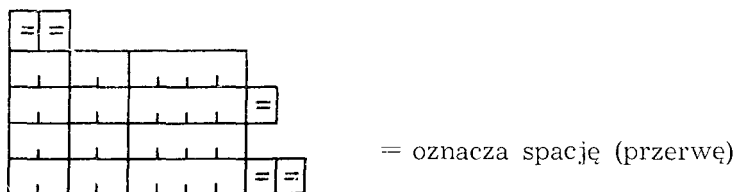
Rozkazy zewnętrzne nie istnieją w liście rozkazów maszyny, lecz służą do tego, aby program pierwotny, w zależności od tego jaki rozkaz zewnętrzny został wprowadzony wybrał taki lub inny sposób postępowania. Każdy rozkaz wczytywanego programu musi być zaopatrzony w symbol mówiący czy jego adres ma ulec zmianom (przeadresowaniu) i w jaki sposób. Program pierwotny jest uprzedzany za pomocą odpowiednich rozkazów zewnętrznych, mówiących, że rozkazy oznaczone danym symbolem zmienności należy przeadresować w określony sposób.

Po krótko opiszemy system pracy dotychczas stosowany na maszynie XYZ.\*)

---

\*) System ten jest obecnie zastępowany innym, można go jednak zawsze używać, o ile nie ulegnie zasadniczym zmianom maszyna. W szczególności będzie on używany przy wykorzystywaniu programów już opracowanych przy użyciu tego systemu.

Jako urządzenie wejściowe i wyjściowe maszyna posiada standardowy reproducer na karty dziurkowane firmy Bull; fakt ten zmusza do tego, aby na karcie były wykorzystane nie wszystkie 80 kolumn, lecz 36 lub 72 (ze względu na długość słowa maszyny i system czytania reproducera). To z kolei ogranicza ilość symboli dodatkowych w rozkazie. Rozkazy i liczby są zapisywane na formularzu w tzw. kodzie alfanumerycznym. Formularz składa się z poszczególnych klatek (rys. 26).



Rys. 26

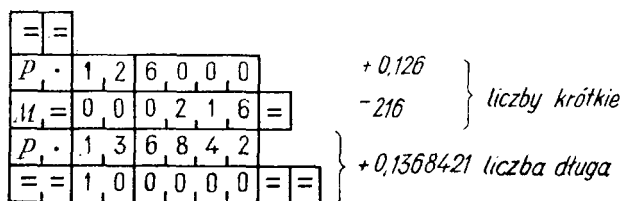
Treść klatki jest przepisywana na kartę dziurkowaną kolejno od góry poczynając od kolumny 37 do 1. Spacje służą jako informacja dla programu pierwotnego. Jednej klatce odpowiada jedna karta dziurkowana.

W ten sposób zapisujemy 4 rozkazy po jednym w każdym wierszu klatki. Każdy rozkaz składa się z 8 symboli: znak *P* lub *M* (+, -), *R* (symbol stały rozkazu), symbol literowy operacji, *K* lub *D* (symbol słowa krótkiego lub długiego) oraz 4 symbole adresowe (mogą to być 4 cyfry dziesiętne lub litera, kropka i 2 cyfry).

Literom na karcie odpowiadają 2 dziurki w jednej kolumnie, kropce — dziurka w wierszu 11 karty, cyfrze — dziurka w odpowiednim wierszu.

Ten zapis jest tłumaczony przez maszynę przy pomocy odpowiedniego programu na układ dwójkowy i znów wypisany na kartach (4 lub 12 słów długich na karcie). Litery posiadają tu odpowiedniki liczbowe. W układzie dwójkowym rozkaz lub liczba są zapisywane na karcie w jednym wierszu — dysponujemy bowiem tylko dwoma symbolami 0 i 1, którym odpowiada brak dziurki i dziurka.

Liczby są zapisywane na formularzu następująco:



Rys. 27

podobnie jak rozkazy, w kolumnach od 37 do 1. Znak + lub - w rozkazie jest w tym systemie istotny, gdyż służy do informowania programu pierwotnego czy adres w danym rozkazie ma ulec zmianie, czy nie, natomiast sposób zapisu adresu informuje o sposobie dokonania zmian w adresie (czyli o względnym punkcie odniesienia adresu). Sposób adresowania wiąże się z podziałem programu na bloki. Program może składać się z najwyżej 30 bloków, z których każdy może zawierać najwyżej 64 rozkazy. Numeracja w każdym bloku jest prowadzona oddzielnie (tzn. od zera). Bloki w kodzie zewnętrznym (tzn. w zapisie dziesiętnym i literowym) oznaczane są zarówno cyframi jak i literami. W adresie symbol bloku jest oddzielany od numeru kropką.

Rozkazy z adresami względnymi mają zawsze znak *M* (minus); adres względny może być zarówno blokowy jak i nieblokowy\*). W kodzie zewnętrznym adres blokowy sposobem zapisu różni się od nieblokowego, natomiast w kodzie wewnętrznym (dwójkowym) — nie. W związku z tym program może zawierać adresy względne wyłącznie blokowe albo nieblokowe, co jest sygnalizowane programowi pierwotnemu przez odpowiednie rozkazy zewnętrzne. Rozkazy z adresami bezwzględnymi mają zawsze znak *P* (plus) i mogą być w każdym programie.

W dalszym ciągu omówimy sposób działania programu pierwotnego.

1. Na pierwszej karcie czytanej przez program pierwotny zapisany jest rozkaz zewnętrzny (z reguły każdy rozkaz zewnętrzny znajduje się na osobnej karcie zwanej kartą tytułową), podający numer miejsca pamięci, od którego będą wprowadzane rozkazy lub liczby. Karta ta nosi nazwę karty *P*.

2. Druga karta tytułowa informuje, jakim symbolem będzie oznaczany we wszystkich częściach programu dany blok (jeżeli program zawiera więcej niż jeden blok, za tą kartą umieszcza się karty informujące ile bloków zawiera program). Jest to karta *B*.

3. Po kartach informujących o nazwach bloków programów umieszcza się kartę informującą czy blok składa się z liczb (karta *L*), z rozkazów z adresami nieblokowymi (karta *R*), czy z rozkazów z adresami blokowymi (karta *G*).

Następnie zostają wczytane rozkazy lub liczby aż do odczytania przez maszynę symbolu, tzw. końca pliku. Wówczas następuje ewentualne przedadresowanie, a program pierwotny jest przygotowany do czytania

---

\*) Adresy blokowe wprowadzono dlatego, że program z adresami względnymi nie blokowymi mógłby korzystać z podprogramów wyłącznie stałomiejscowych (umieszczanych zawsze w tym samym miejscu pamięci), co byłoby bardzo niewygodne.

karty tytułowej. W dalszym ciągu mogą być wczytywane dalsze bloki, lub karta tytułowa z rozkazem zewnętrznym żądającym wykonania natychmiastowego rozkazu na niej zapisanego (rozkażu maszynowego) — najczęściej jest to rozkaz skokowy do początku programu.

Przy stosowaniu tego systemu można maksymalnie wykorzystać pojemność maszyny przez odpowiednie rozmieszczenie programu i podprogramów oraz liczb w pamięci, gdyż wszystkie części programu nie muszą następować kolejno po sobie. Ponadto, jeżeli w programie wielokrotnie występuje ta sama funkcja, nie potrzeba wielokrotnie wstawiać podprogramu tej funkcji, wystarczy wywołanie tego podprogramu za pomocą tzw. łącznika. Podamy przykład standardowego łącznika dla podprogramów funkcji jednej zmiennej (łącznik — to grupa rozkazów spełniająca czynność „łączenia”). W programie głównym (czyli nadrzędnym w stosunku do podprogramu), w miejscu, w którym żądamy wartości określonej funkcji, umieszczamy rozkazy:

|         |     |         |   |
|---------|-----|---------|---|
| $m$     | $N$ | $x$     | umieszczenie argumentu w $\mathbf{M}$ ,                       |
| $m + 1$ | $U$ | $m + 1$ | rozkaz ten umieszcza w $\mathbf{A}$ samego siebie,            |
| $m + 2$ | $S$ | $p$     | $p$ jest adresem pierwszego rozkazu podprogramu wywoływanego. |

Jeżeli jako wartość danej funkcji otrzymujemy jedną liczbę, podprogram z reguły zostawia ją w akumulatorze, a więc rozkaz o numerze  $m + 3$  „zastaje” tę wartość już w  $\mathbf{A}$ . Ponieważ w miejscu  $m + 2$  znajduje się rozkaz skokowy, a więc sterowanie przenosi się z wykonywanego dotychczas programu do podprogramu; konieczne jest istnienie w podprogramie informacji umożliwiającej powrót do programu głównego. Informacją tą jest rozkaz  $S\ m + 3$ , który nie jest umieszczony w podprogramie na stałe, lecz powstaje już po wywołaniu podprogramu w sposób następujący: od liczby reprezentującej rozkaz  $U\ m + 1$ , którą program główny zostawił jako ślad w  $\mathbf{A}$ , zostaje odjęta liczba  $a$ , taka, że otrzymujemy rozkaz  $S\ m + 3$ , który zostanie umieszczony w miejscu zarezerwowanym dla niego na końcu podprogramu. Każdy więc podprogram tego typu rozpoczyna się od rozkazów:

|         |     |     |         |   |
|---------|-----|-----|---------|---|
| $p$     | $B$ | $k$ | 41      | w miejscu 41 znajduje się liczba $a$ ,    |
| $p + 1$ | $A$ | $k$ | $p + c$ | $c$ jest ilością rozkazów w podprogramie. |

Użycie rozkazu  $B$  pozwala wywoływać podprogram przez rozkazy zarówno dodatnie (inwariantne) jak i ujemne (wariantne). Oczywiście podprogram może być programem głównym dla innych podprogramów, tzn. może z nich korzystać.



Istnieje pewna ilość podprogramów typowych, a więc obliczających wartości funkcji elementarnych i częściej spotykanych, np. sinus, cosinus, exponent, mnożenie macierzy, transponowanie macierzy itp. Ponadto dla pewnych zagadnień istnieją podprogramy specjalne.

#### Przykład korzystania z podprogramu:

Zadanie polega na obliczeniu odległości punktów sieci  $A$  od punktów sieci  $B$ . Ilość punktów w sieci  $A$  wynosi  $m$ , w sieci  $B$  —  $n$ . Współrzędne tych punktów umieszczamy odpowiednio parami w blokach  $A$  i  $B$ . Staramy się skonstruować program w ten sposób, aby można go było stosować wielokrotnie, dla dowolnego układu punktów  $A$  i  $B$ . Program główny (w kodzie zewnętrznym) nazwiemy blokiem  $M$ .

Bloki liczb  $A$  i  $B$  zorganizujemy następująco: na krótkim miejscu zerowym znajdują się odpowiednio liczby  $m$  i  $n$ ; na pierwszym miejscu w bloku  $A$  znajduje się rozkaz  $PRL\ K\ p$ , gdzie  $p$  jest miejscem przecinka dwójkowego liczb bloków  $A$  i  $B$  \*). Następnie kolejno parami liczby  $x_i^A, y_i^A$  w bloku  $A$ , a w bloku  $B$  —  $x_i^B, y_i^B$ .

#### Program główny \*\*)

0\*\*\*)  $MRUK\ A \cdot 01$

Rozkaz zerowy powoduje umieszczenie w  $A$  pierwszej (krótkiej) liczby bloku  $A$ . W trakcie wprowadzania zostanie dodana do adresu w tym rozkazie (ponieważ rozkaz ma znak  $M$ ) liczba będąca numerem miejsca w pamięci, od którego jest umieszczony blok  $A$ . Jeżeli np. blok  $A$  jest umieszczany od miejsca 400, rozkaz zerowy przyjmie w maszynie ostateczną postać —  $(11 \cdot 2^{-5} + 401 \cdot 2^{-17})$  czyli —  $U(k)401$ .

1  $PRAK\ 0000$

Rozkaz pierwszy jest rozkazem z adresem bezwzględny (znak  $P$ ), powoduje on umieszczenie liczby krótkiej z  $A$  w zerowym miejscu pamięci. Do adresów rozkazów ze znakiem  $M$ , zawierających litery „blokowe”  $B, M, N$ , zostaną dodane odpowiednio adresy początkowe tych bloków.

\*) Jeżeli chcemy liczby w maszynie traktować jako większe od jedności, umawiamy się, że przecinek znajduje się na miejscu 5 i odpowiednio uwzględniamy to w programie.

\*\*\*) Patrz opis kodu zewnętrznego, str. 62.

\*\*\*\*) Początek bloku  $M$ .

|    |                |               |   |
|----|----------------|---------------|---|
| 2  | <i>M R U K</i> | <i>A · 00</i> |   |
| 3  | <i>P R A K</i> | 0002          |   |
| 4  | <i>M R U K</i> | <i>N · 01</i> | Regeneracja rozkazów umieszczania<br>$x_1^A, y_1^A$ ;<br>W miejscu 34 znajduje się liczba $2^{-16}$ ,<br>wprowadzana tam przez program pierwotny. |
| 5  | <i>M R A K</i> | <i>M · 16</i> |   |
| 6  | <i>P R R K</i> | 0034          |   |
| 7  | <i>M R A K</i> | <i>M · 24</i> |   |
| 8  | <i>M R U K</i> | <i>N · 02</i> | Regeneracja rozkazów odejmowania<br>$x_1^B, y_1^B$ .<br>Regeneracja rozkazu zapamiętywania<br>wyniku w bloku C.                                   |
| 9  | <i>M R A K</i> | <i>M · 17</i> |   |
| 10 | <i>P R R K</i> | 0034          |   |
| 11 | <i>M R A K</i> | <i>M · 25</i> |   |
| 12 | <i>M R U K</i> | <i>N · 06</i> |   |
| 13 | <i>M R A K</i> | <i>M · 35</i> |   |
| 14 | <i>M R U K</i> | <i>B · 00</i> |   |
| 15 | <i>P R A K</i> | 0001          |   |
| 16 | <i>M R U D</i> | <i>A · 02</i> |   |
| 17 | <i>M R R D</i> | <i>B · 02</i> | $x_1^A - x_1^B$   |
| 18 | <i>P R A D</i> | 1008          |   |
| 19 | <i>P R P K</i> | 0035          |   |
| 20 | <i>P R X D</i> | 1008          |   |
| 21 | <i>P R I K</i> | 0000          | $(x_1^A - x_1^B)$   |
| 22 | <i>P R C K</i> | 0000          |   |
| 23 | <i>P R A D</i> | 1010          |   |
| 24 | <i>M R U D</i> | <i>A · 04</i> |   |
| 25 | <i>M R R D</i> | <i>B · 04</i> | $(y_1^A - y_1^B)$   |
| 26 | <i>P R A D</i> | 1008          |   |
| 27 | <i>P R P K</i> | 0035          |   |
| 28 | <i>P R X D</i> | 1008          |   |
| 29 | <i>P R I K</i> | 0000          |   |
| 30 | <i>P R C K</i> | 0000          |   |
| 31 | <i>P R D D</i> | 1010          | $(x_1^A - x_1^B)^2 + (y_1^A - y_1^B)^2$   |
| 32 | <i>P R P K</i> | 0035          |   |
| 33 | <i>M R U K</i> | <i>M · 33</i> |   |
| 34 | <i>M R S K</i> | <i>R · 00</i> | Wywołanie programu pierwiastka, za-<br>pamiętanie $\sqrt{(x_1^A - x_1^B)^2 + (y_1^A - y_1^B)^2}$  |
| 35 | <i>M R A D</i> | <i>C · 00</i> |   |
| 36 | <i>M R U K</i> | <i>M · 17</i> |   |
| 37 | <i>M R R K</i> | <i>N · 03</i> |   |
| 38 | <i>M R A K</i> | <i>M · 17</i> |   |
| 39 | <i>M R U K</i> | <i>M · 25</i> |   |

|     |         |        |   |
|-----|---------|--------|---|
| 40  | M R R K | N · 03 | Przeadresowania   |
| 41  | M R A K | M · 25 |   |
| 42  | M R U K | M · 35 |   |
| 43  | P R R K | 0034   |   |
| 44  | M R A K | M · 35 |   |
| 45  | P R U K | 0001   |   |
| 46  | P R R K | 0032   | Licznik   |
| 47  | M R T K | M · 15 |   |
| 48  | M R U D | N · 04 | Przesłanie do miejsca 1018 adresu poczatkowego wyniku, tzn. C · 00, tak jak |
| 49  | P R A D | 1018   | wymaga tego standardowy program wypisujący                                  |
| 50  | M R U K | B · 00 | wyniki w układzie dziesiętnym („Wyprod.”),                                  |
| 51  | P R A K | 1020   | podanie do programu wyprawdzającego   |
| 52  | M R U K | M · 52 | ilości wyników.   |
| 53  | M R S K | X · 00 | Program wyprawdzający „Wyprod”  |
| 54  | M R U K | M · 16 | zawsze zajmie bloki X : Y.  |
| 55  | M R R K | N · 03 |   |
| 56  | M R A K | M · 16 | Przeadresowania w bloku A.  |
| 57  | M R U K | M · 24 |   |
| 58  | M R R K | N · 03 |   |
| 59  | M R A K | M · 24 |   |
| 60  | P R U K | 0002   |   |
| 61  | P R R K | 0032   | Licznik.  |
| 62  | P R A K | 0002   |   |
| 63  | M R T K | M · 08 |   |
| 0*) | P R G K | 0042   | Skok (z zatrzymaniem) do programu   |
| 1   | M R U D | A · 02 | pierwotnego — czytanie nowego programu                                      |
| 2   | M R R D | B · 02 | lub nowych liczb.   |
| 3   | P R G K | 0004   | Pseudorozkazy służące do regeneracji,                                       |
| 4   | P R U D | 0000   | przeadresowań i łączności z „Wyprodem”.                                     |
| 5   | P R U K | C · 00 |   |
| 6   | M R A D | C · 00 |   |
| 7   | M R H D | S · 63 |   |

Program pierwiastka znajduje pierwiastek kwadratowy danej liczby wg wzoru iteracyjnego Newtona:

$$x_n = \frac{1}{2} \left( x_{n-1} + \frac{N}{x_{n-1}} \right); \quad x_n \xrightarrow{n \rightarrow \infty} \sqrt{N}$$

\*) Początek bloku N.

## Program pierwiastka.

|    |      |        |
|----|------|--------|
| 0  | PRBK | 0041   |
| 1  | MRAK | 0023   |
| 2  | PRUK | 0031   |
| 3  | PRAD | 1008   |
| 4  | PRMD | 1010   |
| 5  | PRUK | 0000   |
| 6  | MRDK | 0026   |
| 7  | MRAK | 0009   |
| 8  | PRUD | 1010   |
| 9  | PRPK | p      |
| 10 | PRQD | 1008   |
| 11 | PRUK | 0033   |
| 12 | PRLK | 0035   |
| 13 | PRDD | 1008   |
| 14 | PRPK | 0001   |
| 15 | PRAD | 1008   |
| 16 | PRND | 1008   |
| 17 | PRXD | 1008   |
| 18 | PRIK | 0000   |
| 19 | PRRD | 1010   |
| 20 | MRBD | 0024   |
| 21 | MRTK | 0008   |
| 22 | PRUD | 1008   |
| 23 | PRGK | 0000   |
| 24 | PRGK | 0000   |
| 25 | PRGK | 0008   |
| 26 | PRIK | 0000   |
| 27 | MRHD | S · 63 |

Utworzenie rozkazu powrotu do programu głównego, jako zerowe przybliżenie bierzemy  $\frac{1}{2}$

Utworzenie rozkazu PKp.

Miejsce rozkazu powrotu.

$\varepsilon$  (Jeżeli:  $|x^2 - N| < \varepsilon$  koniec liczenia).

Ten pseudorozkaz sygnalizuje programowi pierwotnemu koniec wprowadzania rozkazów.

W ten sposób otrzymujemy odległości wszystkich punktów sieci  $B$  kolejno od punktu  $(x_1^A, y_1^A), (x_2^A, y_2^A), \dots, (x_m^A, y_m^A)$ . Jeżeli nie mieszczą się w pamięci współrzędne wszystkich punktów któreś z sieci, wprowadzamy je częściami (program jest do tego przystosowany).

Po przetłumaczeniu programu na układ dwójkowy, układamy program wraz z podprogramami następująco (tzn. składamy karty programów).

=====

Program pierwotny (karty programu).

|            |  |
|------------|--|
| <i>P a</i> | <i>a</i> jest adresem rzeczywistym miejsca <i>A</i> ·00.               |
| <i>B A</i> | Sygnalizuje, że od miejsca <i>a</i> będzie umieszczony blok <i>A</i> . |
| <i>P b</i> | Sygnalizacja bloku <i>B</i> .  |
| <i>B B</i> |  |
| <i>P c</i> |  |
| <i>B C</i> | Sygnalizacja bloku <i>C</i> .  |
| <i>P w</i> | } „Wyprod”.<br>(Pozostałe karty tytułowe „Wyprod” posiada na stałe).   |
| _____      |  |
| _____      |  |
| _____      |  |
| <i>P r</i> | } Podprogram pierwiastka wraz z kartami tytułowymi.                    |
| <i>B R</i> |  |
| <i>R</i>   |  |
| _____      |  |
| _____      |  |
| <i>P g</i> | } Program główny.  |
| <i>B M</i> |  |
| <i>N</i>   |  |
| <i>G</i>   |  |
| _____      |  |
| <i>P a</i> | } Blok <i>A</i> .  |
| <i>L</i>   |  |
| _____      |  |
| _____      |  |

P b

L

Blok B.

S

Na karcie S umieszczamy rozkaz SKM · 00.

Jeżeli za tymi kartami umieścimy znów plik z blokiem A i B (lub tylko B) i kartę S, to po obliczeniu, wypisaniu bloku C i uruchomieniu maszyny — maszyna wczyta nowe dane i obliczy nowy blok C. Zauważmy, że po ewentualnych niewielkich zmianach program ten może służyć jako podprogram, np. przy rachunkowym opracowaniu sieci triangulacyjnych, nawiązanych do punktów stałych, nie podlegających przesunięciu [5].

Stosowanie programu pierwotnego ułatwia znacznie programowanie, każdy rozkaz musi jednak człowiek sam napisać. (Niektóre ośrodki obliczeniowe, szczególnie w ZSRR, nie korzystają z programów pierwotnych; w takim systemie np. podprogramy muszą zawsze znajdować się w tych samych miejscach pamięci, są więc z reguły zapamiętane na stałe).

#### 2.4. Programowanie automatyczne

Programowanie automatyczne polega na tym, że tzw. program programujący tłumaczy otrzymywane informacje, zapisane w odpowiednim kodzie na język maszyny (na miejsce wzorów i wyrazów wstawia całe podprogramy). Ponieważ program programujący może być dowolnie duży (ograniczany tylko wielkością pamięci), zewnętrzna lista rozkazów, czyli wyrazów i znaków, „znanych” przez ten program może również być bardzo obszerna. Programowanie sprowadza się wówczas niejako do wypisywania nazw żądanych podprogramów, z których nie wszystkie oczywiście mają charakter arytmetyczny, oraz parametrów. Praca jest jeszcze o tyle łatwiejsza, że większość programu pisze się po prostu słowami, a formuły matematyczne — wzorami. Zapis programu jest więc o wiele krótszy niż w języku rozkazów elementarnych i ze względu na sposób zapisu znacznie łatwiejszy do sprawdzenia. Program programujący na podstawie odczytywanych rozkazów symbolicznych układa program już bezbłędnie (o ile maszyna działała poprawnie) i odpowiednio adresuje. Poza tym, można pewne części programu pisać w języku rozkazów elementarnych tak, jak przy użyciu programu pierwotnego. Przy zastosowaniu odpowiedniego urządzenia wyjściowego (podobnego do drukarki

dalekopisowej) można bezpośrednio programem opisywać słownie wyniki wypisywane w dowolnym układzie graficznym.

Oczywiście te ułatwienia można uzyskać również bez programu programującego — mianowicie trzeba by stworzyć tę samą bibliotekę podprogramów, z której korzysta program kompilujący \*) i — po stworzeniu programu w języku symbolicznym — przepisywać je na kartkach papieru w odpowiedniej kolejności. Zasługą samej maszyny jest więc niewielka — jest nią tylko szybkość. Istotą automatycznego programowania jest system, który musi być dostatecznie krótki, aby można go było stosować oraz zupełnie ogólny i pełny, aby obejmował wszystkie możliwe programy. Stosowanie takiego systemu wymaga więc dużo pracy zarówno matematyka jak i programisty, wymaga również odpowiednio dużej pamięci, gdyż programy programujące zawierają z reguły po kilka tysięcy rozkazów.

Systemów automatycznego programowania jest tyle, ile ośrodków je stosuje. Jednym z najogólniejszych jest Fortran amerykański i zbliżony do niego Autokod — przygotowywany obecnie do eksploatacji przez ZAM. Ponadto jest opracowywany uniwersalny system, który mógłby być stosowany na wszystkich maszynach — struktura programu programującego pozostawałaby ta sama, zmianie ulegałby tylko język. Natomiast kod zewnętrzny byłby tylko jeden. W ten sposób każdy program bez żadnych zmian mógłby być wykorzystywany w każdym ośrodku korzystającym z tego systemu. System ten nosi nazwę ALGOL (ALGORithmic Language), ośrodek jego opracowania znajduje się w Szwajcarii.

Ponieważ przez mechaniczną kompilację programu z części standardowych otrzymujemy program dłuższy niż przez napisanie bezpośrednie — odpowiednie przygotowanie systemu jest konieczne. Przy użyciu niektórych systemów automatycznego programowania otrzymuje się czasami programy tak duże, że ze względu na ich wielkość jak i czas trwania obliczeń, są niemożliwe do zastosowania.

### 3. Zastosowania maszyn cyfrowych w geodezji

Maszyny cyfrowe mogą z powodzeniem wykonywać wszelkie obliczenia spotykane w geodezji, trudności mogą zaistnieć tylko w wypadku rozwiązywania bardzo dużych układów równań liniowych (ze względu na pojemność pamięci). Aby jednak maszyna była wykorzystana w pełni, a obliczenia wykonywane rzeczywiście z właściwą dla maszyny prędkością — konieczny jest odpowiedni system komunikacji z maszyną.

---

\*) Program kompilujący — składający program z gotowych podprogramów.

Na przykład przy użyciu wejścia i wyjścia na kartach dziurkowanych dokonanie transformacji helmertowskiej (samo liczenie) około 1000 punktów będzie trwało około 5 minut, natomiast wprowadzenie i wyprowadzanie około godziny. W takim więc przypadku maszyna jest wykorzystywana najwyżej w 20%, gdyż urządzenie dziurkujące wypuszcza przeciętnie jedną kartę na sekundę. Doliczając czas pisania na kartach, tłumaczenia na układ dwójkowy, oraz tabulowania wyników — otrzymamy czas znacznie dłuższy. Dopiero stosowanie urządzeń wejścia i wyjścia, budowanych specjalnie dla maszyn cyfrowych (np. marki Creed), pozwala sprowadzić wprowadzanie i wyprowadzanie do wymiarów czynności pomocniczych.

Maszyny bez bardzo szybkiej komunikacji z otoczeniem można z powodzeniem stosować do zagadnień o niezbyt wielkiej ilości wyników, a o dużej ilości działań.

Ponieważ w większości rachunków geodezyjnych stosunek ilości danych wejściowych i wyjściowych do ilości działań jest duży, a są to z kolei rachunki masowe — sposób eksploatacji maszyny cyfrowej gra tu rolę bardzo istotną. Dlatego też budowane są maszyny specjalne, uwzględniające przede wszystkim potrzeby np. geodezji.

Maszyna taka posiada pewną ilość programów umieszczonych na stałe w pamięci (lub w postaci tablicy połączeń z możliwością wymiany). Wówczas sposób wczytywania danych jak i wypisywania wyników jest tak dobrany, aby geodeta żądający wyników mógł bezpośrednio korzystać z maszyny. Taką maszyną jest na przykład Z-11 firmy Zuse (NRF). Tego typu maszyny jednakże są mało przydatne przy rozwiązywaniu układów równań. Zagadnienia te — w wypadku dużych układów — wymagają maszyn o dużej pamięci i dosyć szybkich, a maszyny specjalne są na ogół małe, chociaż szybkie. Dysponowanie maszyną dużą i uniwersalną pozwala nie tylko na rozwiązywanie układów równań normalnych, ale również na ich układanie, a więc rachunki wykonywane przez człowieka mogą być ograniczone do minimum. Zadania te z powodzeniem może spełniać każda maszyna o odpowiednio dużej pamięci — i przede wszystkim — przy odpowiednim systemie podawania informacji.

W Biurze Obliczeń i Programów w Warszawie prowadzone są prace nad automatyzacją rachunków geodezyjnych. Między innymi istnieją np. programy rozwiązywania układów liniowych. Obecnie można rozwiązywać układy symetryczne do 24 równań (ze względu na niewielką pojemność pamięci), przy czym czas rozwiązania, bez wprowadzenia danych i wyprowadzenia wyników, wynosi zaledwie około 2 minut. Oczywiście przygotowanie danych, tzn. umieszczenie na kartach dziurkowanych, sprawdzenie itp. trwa znacznie dłużej. Państwowe Przedsiębiorstwo



Geodezyjne systematycznie dostarcza układy równań, które są rozwiązywane za pomocą tego programu. Koszt rozwiązania wraz z wyliczeniem macierzy odwrotnej i sprawdzeniem wynosi od 100 do 250 zł — w zależności od rzędu macierzy współczynników. Robione są próby masowego obliczania długości boków i azymutów, również na zamówienie PPG. Ze względu na inne prace prowadzone na maszynie XYZ „cykl produkcyjny” trwa kilka dni (zamawiający dostarcza dane i odbiera rozwiązanie w ustalone dni tygodnia).

W trakcie opracowania znajdują się programy przeliczania współrzędnych, wcięć oraz dla innych rachunków geodezyjnych.

Największą trudność stanowi tu nie sama praca obliczeniowa, lecz organizacja dostarczania danych i wyprowadzanie wyników, gdyż zajmuje to bardzo dużo czasu, od czego z kolei są uzależnione koszty. Z chwilą uruchomienia wejścia i wyjścia na taśmę dalekopisową — bardzo pewnego i szybkiego — trudności te znacznie się zmniejszą.

Współpraca z PPG rozwija się tak, że w chwili ukazania się niniejszego zeszytu doświadczenia będą już znacznie bogatsze.

*Rękopis dostarczono Redakcji w styczniu 1960 r.*

#### LITERATURA

- [1] *Alt Fr.* Electronic digital computers, their use in science and engineering. Academic Press, New York — London, 1958.
- [2] *Booth A. D., Booth K. H. V.* Automatic digital calculators. Butterworths Scientific Publications, London, 1956.
- [3] *Booth K. H. V.* Programming for an automatic digital calculators. wyd. j.w. London, 1958.
- [4] *Erszow A. P.* Programmirujuszczaja programma dla bystrodiejstwujuszczej szczotnoj masziny. Izd. A. N. S. S. R. Moskwa, 1958.
- [5] *Hausbrandt S.* W jaki sposób można by stosować transformację Heimerta... Biuletyn W. A. T. Rok VIII, Warszawa, 1959.
- [6] *Jeanel J.* Programming for digital computer. McGraw Hill Book, New York, 1959.
- [7] *Kitow A.* Elektroniczne maszyny cyfrowe. Wyd. M O. N., 1959.
- [8] *Wilkes M. V., Wheeler D. J., Gill S.* The preparation of programs for an electronic digital computer. Addison-Wesley Publ. Co. Mass. 1957.
- [9] Autokod — system programowania automatycznego (skrypt przygotowany w Zakładzie Aparatów Matematycznych).

*ВЛОДЗИМЕЖ ОСТАЛЬСКИ  
ЕЖЫ ГАЗЬДЗИЦКИ*

## АВТОМАТИЧЕСКИЕ ЦИФРОВЫЕ МАШИНЫ И ИХ ПРИМЕНЕНИЕ В ГЕОДЕЗИИ.

### Р е з ю м е

В первой части доклада описана в популярной, доступной форме схема и принципы действия автоматических цифровых машин. Дана классификация математических машин, затем представлена блок-схема автоматической машины, в которой выделено пять основных механизмов: ввод, арифмометр, запоминающее устройство, управляющий механизм и вывод. Описание арифмометра дополнено примером технической реализации многоцифрового сумматора в машине с параллельным пересыланием чисел. В форме примера даны также принципы действия магнитных запоминающих устройств на барабане и на сердечниках.

Во второй части описаны вкратце общие принципы составления программ для цифровых машин. Дан подробный реестр команд польской машины XYZ, построенной Отделением Математических Аппаратов Польской Академии Наук, а также и примеры программ для этой машины. При примерах описано способ использования подпрограмм, а также систему программы ввода.

Доклад содержит кроме того сведения о конкретных применениях цифровых машин в геодезических вычислениях.

WŁODZIMIERZ OSTALSKI  
JERZY GAŹDZICKI

## AUTOMATIC DIGITAL COMPUTERS AND THEIR USE IN GEODESY

### S u m m a r y

In the first part of this paper the construction and operation principles of automatic digital computers have been described in a simple way.

Classification of mathematical machines having been done, a scheme of an automatic compound computer has been presented of which five fundamental arrangements are segregated: input, arithmometer, memory, control unit and output. The description of arithmometer is supplemented by an instance of technical realisation of a multifigure adder in a machine with a parallel conveying of digits. There have also been given the principles of operating the magnetic drum and core memories.

In the second part general principles of programm making for digital computers have been given. A detailed list of commands for Polish computer XYZ, constructed at the Department for Mathematical Machines of Polish Academy of Science, as well as some instances of programmes for the said calculator have been annexed. The instances are supplied with a description of use both subroutines and initial orders.

The paper contains, moreover, information about real application of digital calculators in geodetic computations.