

R. MARCZYŃSKI (Warszawa)

## *ELEKTRONOWE AUTOMATYCZNE MASZYNY CYFROWE*

### 1. Wstęp

Od najdawniejszych czasów człowiek starał się uprościć trudną sztukę rachowania. Czynił to przez wynajdowanie pewnych przyrządów, które miały mu to ułatwić. Już w starożytności znane były aparaty do rachowania, jak na przykład abak.

W miarę rozwoju techniki powstawały coraz to nowe przyrządy stosunkowo proste, jak na przykład sztabki Népera czy też pierwsze arytmometry datujące się od B. Pascala, który w 1642 r. zbudował pierwszy praktyczny model. Od tego czasu arytmometry przebyły długą ewolucję, która w końcu doprowadziła do znanego typu używanego obecnie.

Obok arytmometrów rozwijały się również inne aparaty matematyczne, jak suwaki logarytmiczne, integratory itp. W tej dziedzinie Polacy położyli też pewne zasługi. Należy tu wymienić Abdanka-Abakanowicza, twórcę jednego z typów integratu. Ponadto rozwijały się w tym czasie maszyny do obliczeń statystycznych.

Wszystkie dawniejsze maszyny były urządzeniami czysto mechanicznymi i poważniejsze ulepszanie ich napotykało olbrzymie trudności. Matematyczny aparat mechaniczny jest urządzeniem bardzo skomplikowanym i precyzyjnym. Gdybyśmy go chcieli powiększyć, trudności urosłyby bardzo szybko. Dopiero zastosowanie zdobyczy elektrotechniki umożliwiło budowę aparatów zdolnych do znacznie bardziej skomplikowanych operacji. Jeżeli zbudujemy aparat, w którym element liczący będzie pracował na zasadzie elektrycznej, to na przykład zmiana długości przewodów łączących poszczególne części maszyny nie wpłynie istotnie na zmianę właściwości aparatu. Umożliwi to łatwe łączenie poszczególnych elementów w duże zespoły.

Pierwszym aparatem, w którym zastosowano technikę elektronową na wielką skalę, był ENIAC [6] (Electronic Numerical Integrator and Computer). Była to pierwsza olbrzymia elektronowa maszyna, w której podstawowym elementem liczącym była lampa elektronowa. ENIAC składał się z około 500 000 części, w tym 18 800 lamp elektronowych, podobnych do stosowanych obecnie w radioodbiornikach, 1 500 prze-

każników elektromechanicznych, analogicznych do używanych w centrach telefonicznych, 70000 oporów i wiele innych. O wielkości tego aparatu może świadczyć także potrzebna moc około 150 KW. Jego pierwotnym zastosowaniem były zagadnienia balistyki. ENIAC dodawał na sekundę 5000, mnożył około 400, a dzielił około 50 liczb dziesięciocyfrowych. Maszynę oparto na podobnych zasadach co maszyny mechaniczne i to spowodowało jej wielkie rozmiary.

Równocześnie z ENIAC-iem zbudowano kilka maszyn, w których jako elementy liczące zastosowano przekładniki elektromechaniczne. Maszyny te pracowały jednak znacznie wolniej niż ENIAC. Wykonywały one zaledwie kilka operacji na sekundę.

W dalszym rozwoju aparatów matematycznych, dzięki zastosowaniu nowych koncepcji, ilość elementów została znacznie zmniejszona przy jednoczesnym rozszerzeniu zakresu działania ([4], str. 53). Obecnie elektronowe maszyny cyfrowe mają po kilka tysięcy lamp, przy czym ilość możliwych zagadnień przez nie rozwiązywanych jest znacznie większa niż u ENIAC-a.

Maszyny rachujące dzielą się na dwie dość wyraźnie zróżnicowane grupy: aparaty pracujące na zasadzie analogii oraz aparaty pracujące numerycznie, tzw. *aparaty cyfrowe*. W pierwszych aparatach dane początkowe, przebieg i wyniki obliczeń występują w postaci funkcji ciągłych, a w aparatach cyfrowych w postaci nieciągłej, numerycznej. Maszyny pracujące na zasadzie analogii rozwiązują równania matematyczne przez wykorzystanie zależności między wielkościami fizycznymi. Odczytywanie wyników rozwiązań sprowadza się do pomiaru wielkości fizycznych, np. pomiaru napięcia lub długości. Aparaty te są na ogół proste w budowie oraz stosunkowo tanie. Dotychczas nie istnieje uniwersalna maszyna oparta na zasadzie analogii, to znaczy maszyna przystosowana do rozwiązywania różnorodnych zagadnień matematycznych, lecz każda z nich jest zazwyczaj przeznaczona do pewnej wąskiej klasy zagadnień. Dokładność otrzymywanych wyników jest zwykle ograniczona i nie przekracza w większości przypadków 0,1%, trzeci znak jest więc już niepewny.

Maszyny cyfrowe dzielimy na trzy grupy. Do pierwszej należą dobrze znane arytmetry. Z ich pomocą można dodawać, odejmować, mnożyć, dzielić, a czasami i potęgować. Istotną ich cechą jest to, że liczby nastawia się zazwyczaj po każdej operacji, oraz że po wprowadzeniu w ruch wykonują one tylko jedną operację. Do drugiej grupy należą bardzo skomplikowane aparaty, a mianowicie maszyny do celów statystycznych. Pozwalają one wprowadzać automatycznie bardzo dużo informacji (nie tylko kilka jak w arytmetrze) oraz wykonywać znacznie więcej kolejnych działań po jednorazowym nastawieniu odpowiednich przełączników. Do trzeciej grupy należą automatyczne maszyny cyfrowe. Zarówno arytmetry,

jak i maszyny statystyczne można uważać za przodków automatycznych maszyn cyfrowych.

W tym artykule będziemy się zajmowali automatycznymi maszynami cyfrowymi. Maszyny te służą do automatycznego rozwiązywania w zasadzie wszelkich zagadnień matematycznych rozwiązywalnych za pomocą metod numerycznych. Obliczanie odbywa się w takiej maszynie automatycznie. Od wprowadzenia wszelkich danych potrzebnych do obliczeń aż do chwili otrzymania wyniku maszyna nie wymaga żadnej ingerencji człowieka. Na przykład przy rozwiązywaniu układu równań algebraicznych liniowych o  $n$  niewiadomych na początku wprowadzamy do maszyny wszystkie współczynniki przy niewiadomych i wyrazy wolne oraz przepis procesu liczenia, po czym uruchamiamy maszynę. Aparat automatycznie drukuje wyniki. Od chwili uruchomienia do chwili otrzymania wyników osoba obsługująca urządzenie nie wpływa na przebieg rachowania ani na wyniki obliczeń.

## 2. Zasady działania

**2.1. Ogólna organizacja maszyn cyfrowych.** W automatycznej maszynie cyfrowej można wyróżnić pięć podstawowych elementów:

1. *Ar* — urządzenie arytmetyczne,
2. *P* — pamięć,
3. *S* — urządzenie sterujące,
4. *We* — wejście,
5. *Wy* — wyjście.

*Urządzenie arytmetyczne* jest to organ maszyny, który wykonywa operacje matematyczne, jak dodawanie, odejmowanie, mnożenie, dzielenie oraz wiele innych. Każda maszyna jest scharakteryzowana przez pewną ilość operacji, które może wykonać przy użyciu jednego rozkazu. Za rozkaz uważamy pewien sygnał zrozumiały dla maszyny, a powodujący wykonanie operacji jemu przyporządkowanej. Operację wykonaną przy użyciu jednego rozkazu będziemy nazywali *operacją podstawową*. Na ogół im bardziej skomplikowane operacje podstawowe może wykonać maszyna, tym prostsza jest jej obsługa; np. można zbudować takie urządzenie arytmetyczne, w którym jedną z podstawowych operacji będzie pierwiastkowanie. Przykładem takiej maszyny był ENIAC. To samo działanie (pierwiastkowanie) można otrzymać łącząc ze sobą wielokrotnie dodawania, dzielenia i mnożenia. To drugie postępowanie nazwiemy *operacją programowaną*. Operacja programowana składa się z operacji podstawowych. Uporządkowany zespół rozkazów, za pomocą których można wykonać operację programowaną, nazywamy *programem*.

Drugim elementem maszyny jest pamięć. *Pamięć* jest to urządzenie, w którym magazynujemy liczby i rozkazy. Liczbę lub rozkaz nazywać będziemy ogólnie *słowem*. Słowo jest trwale przechowywane w pamięci i można je w razie potrzeby odczytać. Pamięć można uważać za pewną ilość szufladek odpowiednio ponumerowanych od 0 do  $N$ , gdzie  $N+1$  jest pojemnością pamięci. Do każdej z szufladek można wprowadzić jedno słowo.

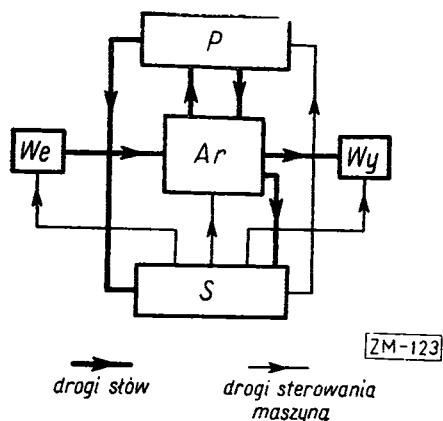
Urządzenie, które rządzi kolejnością i przebiegiem operacji maszyny, nazywamy *urządzeniem sterującym*. Urządzenie to steruje wszystkimi elementami maszyny i w zależności od przebiegu obliczeń przełącza automatycznie maszynę zgodnie z wykonywanym rozkazem.

*Wejście* jest to urządzenie, przez które wprowadzamy do maszyny liczby i rozkazy. Wszystkie liczby, parametry oraz programy, potrzebne do rozwiązania danego zagadnienia, wprowadzamy przez wejście do maszyny przed rozpoczęciem rachunku. Dane te zostają przesłane do pamięci z wejścia przez urządzenie arytmetyczne.

*Wyjście* jest to urządzenie, w którym pojawiają się gotowe wyniki obliczeń w postaci liczb.

Biorąc za podstawę schemat maszyny na rysunku 1 można przedstawić przebieg jej pracy jak następuje:

Po uruchomieniu aparatu wprowadzamy przez wejście do maszyny, a ściślej do pamięci, zespół odpowiednich rozkazów i liczb potrzebnych do rozwiązania danego zadania. Po wprowadzeniu słów do pamięci do



Rys. 1. Schemat blokowy automatycznej maszyny cyfrowej

urządzenia sterującego wchodzi pierwszy rozkaz. Rozkaz ten maszyna wykonuje. Po nim wchodzi następny (zgodnie z programem) i zostaje wykonany. Nakazane działanie zostaje wykonane w urządzeniu arytmetycznym. Z chwilą gdy skończą się obliczenia, wyniki jego są przesyłane do wyjścia, gdzie zazwyczaj zostają utrwalone, np. wydrukowane na pasku papieru. Po całkowitym rozwiązaniu zadania maszyna zatrzymuje się automatycznie.

Potrzeba automatyzacji „przełączania” maszyny powstała z postulatu szybkiego rachowania. Aby w pełni wyzyskać elektroniczne urządzenia arytmetyczne, zdolne np. do 1000 różnych operacji w ciągu 1 sekundy, trzeba je dostatecznie szybko obsługiwać. Jest to możliwe tylko przy pełnej automatyzacji. Pojedyncze dodawanie na takiej maszynie

jest zupełnie nieopłacalne, gdyż czas potrzebny do nastawienia dodajnej i dodajnika oraz odczytania sumy jest nieporównywalnie dłuższy od czasu potrzebnego na dodanie bez maszyny. Na automatycznych maszynach cyfrowych opłacają się tylko duże ciągi kolejnych operacji, tak by czas potrzebny do wprowadzenia danych i przyjęcia wyników był krótki w stosunku do czasu rachowania.

Postulat szybkiego rachowania i, co za tym idzie, szybkiego spełniania rozkazów wyłonił konieczność wprowadzenia programów, które by się wielokrotnie powtarzały. Pojemność obecnych maszyn dochodzi do kilku tysięcy słów, które dadzą się jednocześnie zapamiętać. Łatwo zauważyć, że, przy szybkości np. 1000 operacji na sekundę i przy pojemności kilku tysięcy liczb, bez programów iteracyjnych maszyna przestałaby rachować po kilku sekundach.

Dzięki wprowadzeniu operacji rozróżniania maszyny mogą wykonywać programy iteracyjne. Operacja rozróżniania dotyczy wyboru dróg w programie w zależności od wyników obliczeń. Należy tu podkreślić, że ta operacja jest szczególnie ważna i jest istotą automatycznych maszyn cyfrowych. Operacja rozróżniania jest operacją nieciągłą. Jest wiele różnych kryteriów rozróżniania, np:

- 1) wynik operacji jest  $>0$  lub  $\leq 0$ ,
- 2) „ „ „  $=0$  lub  $\neq 0$ ,
- 3) „ „ „ liczbą parzystą lub nieparzystą,
- 4) „ „ „ przekracza zakres maszyny.

Kryteria te dadzą się jedno z drugich konstruować.

Od czego zależy, czy dane zagadnienie da się rozwiązać za pomocą danej maszyny? Z grubsza biorąc — tylko od dwóch parametrów maszyny: pojemności i szybkości; na przykład do rozwiązania bardziej skomplikowanego równania różniczkowego cząstkowego o czterech zmiennych, występującego w zagadnieniach fizyki, potrzeba by pamięci o pojemności  $10^6$  słów oraz około  $10^{10}$  operacji. Na maszynie zdolnej do 1000 operacji na sekundę do rozwiązania tego zagadnienia potrzeba by około 100 dni bezustannej pracy. Przy obecnym stanie techniki takie długie działanie poszczególnych elementów maszyny bez wprowadzenia błędu jest nieprawdopodobne, gdyż wymagało by, żeby każdy element maszyny zadziałał około  $10^{13}$  razy bez błędu. W istniejących obecnie maszynach przeciętna szybkość operacji dochodzi do 30000 na sekundę przy pojemności pamięci rzędu kilkudziesięciu tysięcy słów.

**2.2. Liczby.** Podstawowymi elementami, na których maszyna wykonuje operacje, są liczby. Liczby piszemy zazwyczaj w układzie dziesiętnym, lecz w maszynach elektronowych stosujemy najczęściej układ dwójkowy, czyli binarny. Zalety układu dwójkowego zwłaszcza wyraźnie

występują przy urządzeniach rachujących elektronowo. Na to, by przedstawić jedną z cyfr układu binarnego, potrzebne jest urządzenie mające tylko dwa stany stabilne, którym odpowiednio przyporządkujemy 0 i 1. Układy rachujące elektrycznie i elektronowo są do tego szczególnie przystosowane.

Poza układami dziesiętnymi i binarnymi w niektórych maszynach stosuje się inne systemy pozycyjne a nawet specjalne systemy niepozycyjne. Stosowanie tych systemów znacznie upraszcza budowę maszyn, powodując jednak potrzebę przeliczania wyników na system dziesiętny. Najczęściej przeliczanie z jednego systemu na drugi odbywa się automatycznie.

W każdej maszynie liczby przedstawia się za pomocą ustalonej ilości cyfr charakteryzujących daną maszynę (np. liczba o 40 cyfrach w układzie binarnym). Liczby względne pisze się bądź jako liczby bezwzględne ze znakami, bądź przy użyciu dopełnień.

Poza znakiem istotnym elementem w liczbie jest także przecinek. Obecnie znane są trzy sposoby uwzględniania przecinka: stały przecinek, przesuwany przecinek i programowy przecinek. W pierwszym przypadku położenie przecinka jest stałe. Najczęściej ustalamy go tak, że liczba  $x$  spełnia nierówność  $-1 \leq x < 1$  (oczywiście można ustalić go inaczej). Zwykle arytmometry pracują przy stałym przecinku. Wymaga to ustalania położenia przecinka po każdej operacji. Stały przecinek pociąga za sobą dużo niedogodności w układaniu programu, gdyż trzeba z góry uważać, by maszyna podczas obliczeń nie przekroczyła swego zakresu liczb, co komplikuje program. Stały przecinek jest uciążliwy zwłaszcza przy tych zagadnieniach, przy których nic nie wiemy o zakresie prawdopodobnego wyniku. Natomiast zaletą stałego przecinka jest to, że bardzo upraszcza budowę urządzenia arytmetycznego.

Przy przesuwanym przecinku liczba jest zbudowana z dwóch części: części wykładniczej i części numerycznej; np. w układzie dwójkowym liczbę przedstawia się w postaci  $2^{\pm p} \pm q$ , gdzie  $\pm q$  jest częścią numeryczną,  $2^{\pm p}$  — częścią wykładniczą. Część numeryczna przedstawia liczbę przy ustalonym przecinku a część wykładnicza położenie przecinka w części numerycznej. Maszyny o przesuwanym przecinku nadają się do znacznie bardziej skomplikowanych zadań a w szczególności do zagadnień badawczych, gdyż przedział liczbowy jest znacznie większy. Maszyny pracujące z przesuwanym przecinkiem mają bardziej skomplikowane urządzenia arytmetyczne. Ta komplikacja powstaje stąd, że już przy prostych i bardzo często używanych operacjach (dodawanie, odejmowanie) przed wykonaniem operacji trzeba najpierw obie liczby zrównać, tzn. tak ustawić przecinki w części numerycznej, żeby cyfry o tej samej pozycji rozwinienia stały pod sobą.

Przy przecinku programowanym realizujemy operacje zmienno-przecinkowe za pomocą odpowiednio skonstruowanych programów elementarnych. Nie powoduje to komplikacji w budowie maszyny, ale komplikuje program oraz zmniejsza pojemność pamięci o tę ilość słów, która jest potrzebna do rozkazów programów zmienno-przecinkowych, a poza tym zmniejsza szybkość liczenia.

**2.3. Operacje arytmetyczne.** W układzie dziesiętnym bywają stosowane dwa sposoby dodawania: sumowanie przez zliczanie lub sumowanie przez użycie tablicy sumacyjnej. Jeżeli urządzenie ma działać za pomocą zliczania, to powinno być przystosowane do wykonania operacji według tablicy 1. Przeniesienie do wyższych rzędów występuje tylko wtedy, gdy do 9 dodamy 1. W przypadku tablicy sumacyjnej urządzenie sumujące musi wykonywać operacje bardziej skomplikowane, według tablicy 2, gdzie  $a$  jest dodajnikiem,  $b$  zaś dodajną.

Tablica jest wbudowana w urządzenie sumujące; po wprowadzeniu dodawanych cyfr w wyniku otrzymujemy sumę i przeniesienie według tablicy 2. W układzie dwójkowym dodawanie znacznie się upraszcza. Obie metody prowadzą do tej samej tablicy sumacyjnej (tablica 3). Odejmowanie sprowadza się do dodawania (w przypadku gdy liczby ujemne występują jako dopełnienie) lub do podobnych tabliczek jak przy dodawaniu ([9], str. 130).

Zarówno dodawanie jak i odejmowanie można przeprowadzić albo kolejno cyfra

za cyfrą, albo jednocześnie na wszystkich cyfrach. W przypadku dodawania jednoczesnego, czyli równoległego (maszyny pracujące na tej zasadzie nazywamy *równoległymi*), sam proces sprowadza się do dwóch kroków: najpierw dodajemy poszczególne cyfry, a później dodajemy przeniesienie. Należy zauważyć, że przy dodawaniu suma dwóch cyfr w układzie dziesiętnym nie może być większa niż 18. Wtedy mamy wynik 8 i przeniesienie 1. W przypadku, gdy wynik jest równy 9, przeniesienie musi być równe 0, jak widać na podanej tablicy sumacyjnej (tablica 2). Pozwala to na dodawanie w dwóch krokach.

TABLICA 1

$0+1=1$	$5+1=6$
$1+1=2$	$6+1=7$
$2+1=3$	$7+1=8$
$3+1=4$	$8+1=9$
$4+1=5$	$9+1=10$

TABLICA 2

$a \backslash b$	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10	11
3	3	4	5	6	7	8	9	10	11	12
4	4	5	6	7	8	9	10	11	12	13
5	5	6	7	8	9	10	11	12	13	14
6	6	7	8	9	10	11	12	13	14	15
7	7	8	9	10	11	12	13	14	15	16
8	8	9	10	11	12	13	14	15	16	17
9	9	10	11	12	13	14	15	16	17	18

$a \backslash b$	0	1
0	0	1
1	1	0

Mnożenie wykonujemy albo przez kolejne dodawanie poszczególnych iloczynów cyfr, przy czym iloczyny bierze się z tabliczki mnożenia wbudowanej w urządzenie, albo przez obliczenie najpierw kolejnych iloczynów mnożnej od 0 do 9, a później sumowanie iloczynów wybranych odpowiednio według mnożnika i właściwie przesuniętych. Mnożenie binarne polega tylko na przesuwaniu i dodawaniu mnożnej do częściowych iloczynów odpowiednio do mnożnika.

Dzielenie wykonujemy albo metodami bezpośrednimi, albo za pomocą wzorów iteracyjnych. W metodach bezpośrednich dzielenie przebiega podobnie jak w automatycznych arytmometrach, tzn. przez kolejne odejmowanie i dodawanie dzielnika lub przez tworzenie kolejnych iloczynów dzielnika od 1 do 9 i odpowiednie odejmowanie. Metod iteracyjnych używa się w maszynach, które nie mają urządzeń do automatycznego dzielenia i sprowadzają je do kolejnych mnożeń i odejmowań. Jeżeli mamy znaleźć iloraz  $b/a$ , znajdujemy najpierw  $1/a$ , a wynik mnożymy przez  $b$ . Odwrotność  $a$  otrzymujemy z wzoru

$$y_{n+1} = y_n(2 - ay_n).$$

Jeżeli  $y_0$  będzie zawarte między 0 a  $2/a$ , to  $y_n \rightarrow 1/a$ , gdy  $n \rightarrow \infty$ .

W niektórych maszynach poza tymi czterema działaniami arytmetycznymi wykonuje się jeszcze inne, bardziej skomplikowane operacje, np. pierwiastkowanie. Na ogół do obliczania  $\sqrt{a}$  stosuje się metody iteracyjne. Jednym z bardzo często stosowanych wzorów iteracyjnych jest wzór Newtona

$$y_{n+1} = \frac{1}{2} \left( y_n + \frac{a}{y_n} \right),$$

gdzie  $y_0$  jest dowolną liczbą. Gdy  $n \rightarrow \infty$ , to  $y_n \rightarrow \sqrt{a}$ . Operacja ta składa się tylko z dzielenia, mnożenia i dodawania.

Ponadto w maszynach wykonuje się i inne operacje w zależności od potrzeb.

**2.4. Programowanie.** Wszystkie operacje wykonywane przez maszynę są sterowane za pomocą rozkazów. Zazwyczaj rozkaz składa się z dwóch, a czasem trzech części: 1) operacyjnej, 2) adresowej, 3) sprawdzającej. W części adresowej jest zawarty (lub są zawarte) adres podający miejsce liczby w pamięci. Na tej liczbie wykonujemy operację, którą podaje część operacyjna rozkazu. Część trzecia, nie we wszystkich maszynach stosowana, dotyczy symboli sprawdzających poprawność pracy maszyny. W zależności od ilości adresów w rozkazie, rozróżniamy rozkazy jednoadresowe, dwuadresowe, ...,  $n$ -adresowe.

Jako przykład budowy rozkazów rozważymy rozkazy czteroadresowe. Zazwyczaj budowa ich przedstawia się następująco:



operacja — adres 1 — adres 2 — adres 3 — adres 4.

Adres 1 — dotyczy pierwszej liczby, na której wykonujemy operację,  
 adres 2 — dotyczy drugiej liczby, na której wykonujemy operację,  
 adres 3 — jest miejscem wyniku,  
 adres 4 — jest adresem następującego rozkazu.

Przy takiej budowie rozkazów kolejność ich wykonywania wynika z samych rozkazów. Wieloadresowe maszyny ułatwiają programowanie.

Jako drugi przykład rozpatrzmy budowę rozkazów jednoadresowych. Mają one kształt

operacja — adres.

Typowym rozkazem jednoadresowym jest na przykład taki: „Do zawartości akumulatora dodaj liczbę, która się znajduje w pamięci pod adresem podanym w rozkazie”. *Akumulator* jest to rejestr w urządzeniu arytmetycznym, w którym gromadzi się wyniki obliczeń.

O kolejności wykonywanych rozkazów w maszynie jednoadresowej decyduje tzw. *licznik rozkazów*. Licznik rozkazów określa adres rozkazu, który ma zostać wykonany. Po każdym wykonanym rozkazie zawartość jego wzrasta o 1. Przy rozkazach jednoadresowych maszyna pracuje w ten sposób, że po rozkazie  $n$  wykonuje rozkaz  $n+1$ , potem  $n+2$  itd., aż do chwili, gdy dojdzie do rozkazu przerywającego proces kolejnego wykonywania programu. Następuje wtedy zmiana zawartości licznika rozkazów zgodnie z zawartością tzw. *rozkazu warunkowego*, w którym znajduje się adres następnego rozkazu. Adres ten zostaje umieszczony w liczniku rozkazów i maszyna rachuje dalej kolejno, począwszy od tego adresu. Rozkazy jednoadresowe komplikują budowę programu, ale są bardziej ekonomiczne, jeśli chodzi o wykorzystanie pamięci.

Porównajmy dla przykładu najprostszy program dla maszyny trójadresowej i jednoadresowej. Na przykład mamy dodać 5 liczb, umieszczonych w pamięci pod adresami  $a, b, c, d, e$ , a wynik umieścić w pamięci o adresie  $p$ . Niech  $S$  oznacza operację dodawania a  $P$  operację przeniesienia. W maszynie trójadresowej program rozwiązujący zagadnienie przedstawia się następująco:

$S a b p,$

$S p c p,$

$S p d p,$

$S p e p.$

Dla maszyny jednoadresowej otrzymujemy program

$S a, S b, S c, S d, S e, P p.$

Zakładając, że ilość cyfr w części adresowej i operacyjnej dla obu maszyn jest taka sama, stwierdzamy, że ilość zajętego miejsca w pamięci w systemie trójadresowym musi być większa niż w jednoadresowym; w tym przykładzie stosunek ilości jest 16 do 12.

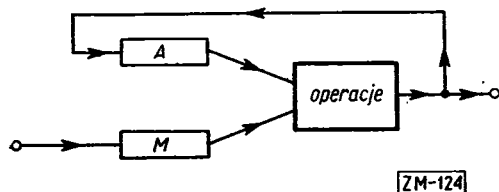
Przebieg przygotowania programu dla danego zagadnienia jest następujący. Najpierw opracowujemy zagadnienie pod względem matematycznym, wybieramy metodę rozwiązania, szacujemy błąd tej metody itp. Następnie zapisujemy przebieg rozwiązania w postaci dyrektyw matematycznych, a potem transformujemy je na „język” zrozumiały dla maszyny, tj. na rozkazy.

Sterowanie wykonywaniem rozkazów może być *liniowe*, tzn. że zaczynamy wykonywać rozkazy od rozkazu oznaczonego numerem 0 oraz że po rozkazie  $n$ -tym następuje rozkaz  $n+1$  itd., albo *nieliniowe*, a mianowicie w programie po rozkazie  $p$  następuje rozkaz  $q$ . Ta nieliniowość może wystąpić bezwarunkowo lub dopiero po spełnieniu pewnego warunku — kryterium rozróżniania (tzw. *rozkazy warunkowe*).

Program można wykonywać liniowo lub skokami w przód albo wstecz; można też wykorzystywać powtórnie rozkazy już użyte. Poza tym jest możliwa zmiana w trakcie rachowania pewnej części programu i wielokrotne wykonywanie części programu, przy czym za każdym kolejnym powtórzeniem może się ta część programu zmieniać. Wszystkie

te przypadki i ich odmiany mogą być uwarunkowane przez wyniki poprzednich operacji.

Relacja między wykonywanym programem a matematyczną metodą rozwiązywania problemu nie jest stała, ale może zmieniać się w zależności od warunków początkowych oraz wyników obliczeń.



Rys. 2. Schemat blokowy urządzenia arytmetycznego

W celu lepszego wykorzystania pamięci należy budować programy możliwie krótkie. Jest to możliwe dzięki metodom iteracyjnym.

W celu bliższego zapoznania się z programowaniem oraz pracą maszyny automatycznie rachującej podamy pewien bardzo prosty schemat organizacji maszyny. Używając tego schematu za przykład będziemy mogli pokazać, w jaki sposób się programuje i wykonuje działania na maszynach.

Za schemat blokowy rozpatrywanej maszyny będziemy uważali rysunek 1. Schemat blokowy urządzenia arytmetycznego podaje rysunek 2; przedstawia on bardzo uproszczone urządzenie, składające się z dwóch rejestrów  $M$  i  $A$ , gdzie  $M$  jest rejestrem, w którym umieszczamy

mnożną lub dzielną,  $A$  zaś jest akumulatorem. Trzecim elementem jest urządzenie wykonujące operację. Przykładowa maszyna jest jedno-adresowa, rozkazy są wykonywane kolejno, jeden po drugim, tzn. że po rozkazie  $n$ -tym maszyna wykonuje rozkaz  $n+1$ . Pamięć jej ma następującą własność: gdy wprowadzamy do niej słowo, to poprzednie zostaje wymazane; przy pobieraniu słów zawartość jej nie ulega zmianie. Spis rozkazów podstawowych wykonywanych przez maszynę i ich symboli podajemy w tablicy 4.

TABLICA 4  
( $P$  — pamięć,  $M$  — rejestr)

Znak rozkazu w języku symbolicznym	Rozkaz
1. $A + P_n$	do zawartości akumulatora dodaj zawartość $P_n$ !
2. $A - P_n$	od zawartości akumulatora odejmij zawartość $P_n$ !
3. $M \times P_n$	zawartość $M$ pomnóż przez zawartość $P_n$ , wynik dodaj do akumulatora!
4. $M : P_n$	zawartość $M$ podziel przez zawartość $P_n$ a wynik dodaj do akumulatora!
5. $A \times 2^n$	zawartość akumulatora pomnóż przez $2^n$ !
6. $A \times 2^{-n}$	„ „ „ „ $2^{-n}$ !
7. $P_n \rightarrow M$	zawartość $P_n$ prześlij do $M_0$ !
8. $A \rightarrow P_n$	zawartość akumulatora prześlij do $P_n$ nie sprowadzając akumulatora do zera!
9. $A \overset{0}{\rightarrow} P_n$	zawartość akumulatora prześlij do $P_n$ i sprowadź akumulator do zera!
10. $A > 0 \rightarrow P_n$	jeżeli zawartość akumulatora jest większa od 0, to następny rozkaz weź z $P_n$ , w przeciwnym przypadku weź kolejny rozkaz!
11. $A < 0 \rightarrow P_n$	jeżeli zawartość akumulatora jest mniejsza od 0, to następny rozkaz weź z $P_n$ , w przeciwnym przypadku weź kolejny rozkaz!
12. $We \rightarrow P_n$	zawartość wejścia prześlij do $P_n$ !
13. $A \rightarrow D$	„ akumulatora wydrukuj na wyjściu!
14. stop	zatrzymaj maszynę!

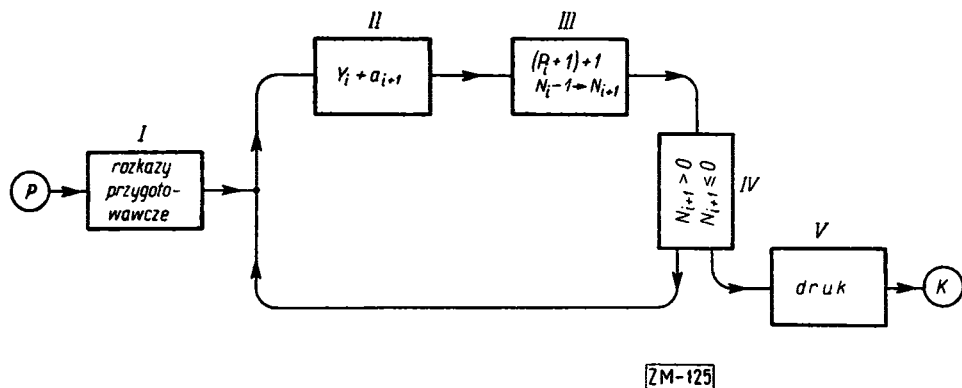
Rozkazy 1, 2, 3, 4, 5, 6 są rozkazami arytmetycznymi obejmującymi cztery działania. Rozkazy 7, 8, 9 są rozkazami przesyłowymi, służącymi do przesyłania liczb z jednej części maszyny do drugiej. Rozkazy

10 i 11 są to tzw. *rozказы warunkowe*; służą one do powiązania wyników rachunku z programem, umożliwiając tworzenie iteracji i powodując zmianę kolejności wykonywanych rozkazów. Rozkazy 12 i 13 służą do przesyłania liczb i rozkazów poza maszyny i odwrotnie. Rozkaz 14 jest oczywisty. W maszynie rozkazy są zakodowane w postaci liczb.

Rozpatrzmy teraz bardzo prosty przykład. Chcemy obliczyć

$$Y = \sum_{n=1}^{200} a_n,$$

gdzie  $a_n$  są pewnymi liczbami uprzednio obliczonymi i umieszczonymi w pamięci. Przy układaniu programu zazwyczaj postępujemy w ten sposób, że najpierw tworzymy schemat blokowy (graficzny) danego problemu. Rysunek ten powinien obrazować wszystkie istotne momenty programu, jego przekształcenia, rozgałęzienia i operacje, które wykonuje urządzenie arytmetyczne w związku z zadaniem problemattem. „Schemat operacji” ([5], t. I, str. 69; t. II, str. 68; t. III, str. 23) dla omówionego problemu jest pokazany na rysunku 3.



Rys. 3. Schemat operacji.  $Y_i = \sum_{n=1}^n a_n$ ,  $N$  — liczba operacji

Poszczególne prostokąty mają w takim schemacie własne znaczenie. Każdy prostokąt odpowiada pewnej grupie operacji. Zazwyczaj rozróżniamy następujące grupy operacji:

- 1) operacje arytmetyczne na liczbach i operacje przesyłowe,
- 2) operacje arytmetyczne i inne na rozkazach,
- 3) rozkazy warunkowe.

Każdej z tych grup przyporządkowujemy w schemacie operacji osobne znaczenie graficzne w postaci prostokąta; dla grupy 1 i 2 o jednym wejściu i wyjściu, dla grupy 3 o jednym wejściu i dwóch wyjściach. Poza

tym rozróżniamy jeszcze początek i koniec schematu, które oznaczamy odpowiednio literami *P* i *K*. Łącząc odpowiednio koniec jednego programu z początkiem innego tworzymy programy stopniowo dłuższe, służące do rozwiązywania coraz to bardziej skomplikowanych zagadnień. Prostokąty podane na rysunku 3 odpowiadają właściwym grupom operacji podanych dalej w szczegółowym programie. Prostokąty *I*, *II* i *V* należą do grupy pierwszej, prostokąt *III* do grupy drugiej, prostokąt *IV* dotyczy rozkazu warunkowego.

Na początku, w pamięci mamy zmagazynowane liczby i rozkazy jak w tabelicy 5.

TABLICA 5

Nr pamięci (adres)	Zawartość pamięci		Objaśnienie
	liczby	rozkazy	
0	1		parametr pomocniczy
1	199		parametr pomocniczy określający ilość sumowań
2	0		magazyn sum częściowych
3		$A + P_1$	rozkazy wiążące pętlę iteracyjną
4		$A \xrightarrow{0} P_1$	
5		$A + P_{301}$	sumowanie kolejnych składników
6		$A + P_2$	
7		$A \xrightarrow{0} P_2$	
8		$A + P_5$	operacja na rozkazie; rozkaz $A + P_{301}$ przez dodanie 1 przechodzi na $A + P_{302}$
9		$A + P_0$	
10		$A \xrightarrow{0} P_5$	
11		$A + P_1$	liczenie ilości sumowań
12		$A - P_0$	
13		$A > 0 \rightarrow P_4$	rozkaz warunkowy
14		$A + P_2$	rozkazy drukujące wynik
15		$A \rightarrow D$	
...	...	...	
301	$a_1$		liczby sumowane
302	$a_2$		
...	...	...	
500	$a_{200}$		

Pamięć  $P_2$  służy za magazyn  $\sum a_n$ ; gromadzimy tam kolejno sumy częściowe. W celu zaoszczędzenia miejsc w pamięci, rozkazy, które wykonują zadanie, przekształca się w miarę postępu liczenia. Przez dodanie 1 do adresu rozkazu umieszczonego w  $P_5$  w każdym kolejnym cyklu iteracji

rozkaz ten zmieniamy w ten sposób, że za jego pomocą wszystkie kolejne liczby zostaną wybrane z pamięci i zsumowane. Za pomocą rozkazów umieszczonych w  $P_{12}$  i  $P_{13}$  oraz uprzednio umieszczonej w pamięci liczby 199 liczymy kroki iteracji w ten sposób, że po każdej iteracji od 199 odejmujemy 1 i badamy, czy reszta jest większa, czy równa 0. Dopóki wynik jest większy od 0, dopóty maszyna wykonuje iterację, gdy zaś wynik zrówna się z zerem, maszyna przechodzi do rozkazu 14, a następnie za pomocą rozkazu 15 drukuje wynik zadania na wyjściu.

Przytoczony przykład jest przykładem tzw. *programu elementarnego*. Każdy program elementarny przed wprowadzeniem do maszyny zostaje (np. przy użyciu dziurkarki dalekopisowej) odpowiednio przygotowany do maszyny. Zostaje on w tym przypadku przetłumaczony na odpowiednią kombinację dziurek w pasku papieru. Przy tłumaczeniu na dziurki każdy rozkaz, składający się zazwyczaj z kilku symboli cyfrowych i literowych, tłumaczymy i dziurkujemy oddzielnie. Taśma papieru odpowiednio podziurkowana przedstawia zakodowany program. W tej formie można już dany program wprowadzić do maszyny. Łącząc programy elementarne, np. przez zlepianie pasków papieru, otrzymujemy program główny. Program główny wraz z danymi początkowymi i parametrami potrzebnymi do obliczeń wprowadza się do maszyny. Służy on do sterowania maszyną. Maszyna wykonuje rozkazy zawarte w programie i tym samym rozwiązuje postawione zagadnienie.

Zbiór programów elementarnych tworzy tzw. *bibliotekę programów*. Im bogatsza jest ta biblioteka, tym więcej zadań na danej maszynie można rozwiązać. Programowanie jest w obecnej chwili jednym z najzwyklejszych zajęć przy obsłudze maszyn. Budując programy staramy się, aby były one możliwie uniwersalne i nadawały się bez przeróbek do rozwiązywania różnych zagadnień. Na przykład program rozwiązywania układu równań liniowych o  $n$  niewiadomych budujemy dla największego  $n$  (zwykle ograniczeniem jest pojemność pamięci). Jeżeli chcemy rozwiązać układ o mniejszej ilości niewiadomych, stosujemy ten sam program zmieniając go przez wprowadzenie odpowiednich parametrów.

Ostatnio rozwój idzie w kierunku tzw. *automatycznego programowania*. Uzyskuje się je przez odpowiednio skonstruowane programy elementarne lub odpowiednią organizację maszyny. Programowanie automatyczne jest tak skonstruowane, by skrócić i uprościć proces programowania. Przy rozwiązywaniu zagadnienia podaje się tylko program ramowy a szczegóły programowania maszyna wykonuje sama. Zmniejsza to możliwość pomyłek spowodowanych przez programującego.

**2.5. Kontrola wyników.** Przy wykonywaniu rachunków jest istotne, żeby otrzymane wyniki były wolne od błędów. Źródła błędów w obliczeniach mogą być bardzo różnorodne. W zwykłych arytmometrach

przyjmuje się, że jeśli arytmometr nie jest uszkodzony, to wyniki są poprawne. Często identyfikuje się uszkodzenie arytmometru z zacięciem. Pogląd ten jest niesłuszny, gdyż bywają w arytmometrach uszkodzenia chwilowe nie powodujące zacięcia, ale dające co pewien czas fałszywe wyniki. Ponieważ maszyny elektronowe nie mają ruchomych części, uszkodzenie nie może powodować zacięcia. Ponadto, ze względu na wielką ilość automatycznie wykonywanych operacji, musimy dysponować urządzeniami i sposobami pozwalającymi na kontrolę wyników. W celu usunięcia lub tylko zmniejszenia ilości niewykrytych błędów w obliczeniach jest pożądana automatyczna kontrola za pomocą odpowiedniego urządzenia kontrolującego. Mechanizm wykrywający błędy powinien poza ich stwierdzeniem również określić, w którym miejscu obliczenia i w jakiej części maszyny powstał błąd.

Jest kilka sposobów umożliwiających kontrolę pracy maszyny; dadzą się one podzielić na matematyczne i techniczne.

W tym artykule zajmiemy się tylko niektórymi sposobami automatycznego wykrywania błędów. Należy zwrócić uwagę, że urządzenia kontrolujące albo zwiększają wielkość maszyny, albo zwiększają czas trwania obliczeń. Należy podkreślić, że możliwość wystąpienia błędu w rachunkach ogranicza zakres używalności maszyny. Przy bardzo długich obliczeniach prawdopodobieństwo powstania błędu jest bliskie 1, a to ze względu na niedoskonałość techniczną elementów, z których jest zbudowana maszyna.

Jednym ze sposobów wykrywania błędów jest zwielokrotnianie elementów maszyny. Na przykład budujemy dwa niezależne jednakowe aparaty. Aparaty rachują jednocześnie, a odpowiednie urządzenie porównujące sprawdza wszystkie wyniki krok za krokiem. W razie powstania różnicy w obliczeniach między maszynami maszyny zatrzymują się a operator usuwa błąd. Przez wprowadzenie trzech jednakowych maszyn można w razie powstania błędu kontynuować obliczenia, gdyż mając jednocześnie trzy wyniki możemy w razie błędu jednej maszyny prawie na pewno stwierdzić, w której z nich wystąpił błąd. Uszkodzoną maszynę można albo zatrzymać (dwie pozostałe pracują dalej) nie przerywając rachowania, albo za pomocą dwóch dobrze pracujących skorygować wynik trzeciej.

Inna metoda, nie zwiększająca wielkości maszyn, polega albo na wielokrotnym wykonaniu przez tę samą maszynę tego samego obliczenia za pomocą różnych programów, albo na sprawdzeniu wyników za pomocą specjalnych programów sprawdzających. Metody te przedłużają jednak czas rozwiązywania zagadnień.

W niektórych maszynach stosuje się też sprawdzanie aktualnego stanu technicznego maszyny w pewnych określonych odstępach czasu. Na przy-

kład przed przystąpieniem do obliczeń nadajemy pewien program tak skonstruowany, by na podstawie otrzymanych wyników można było nie tylko stwierdzić uszkodzenie, ale i określić jego miejsce. Powyższą metodę można połączyć z tzw. *kontrolą marginesową*. Metoda ta służy do badania stanu zużycia lamp elektronowych. Znaczenie jej polega na tym, że przed rozpoczęciem obliczeń wykrywamy te lampy elektronowe, które są już tak zużyte, że podczas działania maszyny przestałyby prawidłowo działać.

Są jeszcze dwie ciekawe metody kontrolujące wyniki. Pierwsza z nich jest szczególnie przydatna do sprawdzania urządzeń arytmetycznych. Każdej liczbie w maszynie przyporządkowujemy pewien sygnał (liczbę). Liczba ta jest elementem sprawdzającym, tzw. *wagą liczby*. Metoda ta pochodzi od dobrze znanej „próby przez dziewiątkę”. Urządzenie sumuje osobno liczby i osobno ich wagi oraz oblicza wagę sumy i porównuje wyniki. Suma wag poszczególnych składników powinna być taka, jak waga sumy.

Druga metoda sprawdzająca nadaje się szczególnie do kontroli skorygowania zawartości pamięci. Polega ona na wprowadzeniu specjalnego sposobu kodowania informacji. Jest ona przystosowana szczególnie do urządzeń pracujących w układach binarnych. Podamy najprostszy przykład ilustrujący tę metodę. W tym najprostszym przypadku będziemy mogli wykryć jeden i tylko jeden błąd w liczbie binarnej, tzn. że wykryjemy błąd tylko wtedy, gdy jedna cyfra zmieni swą wartość z 0 na 1 lub odwrotnie. Przypuśćmy, że mamy  $n$ -cyfrową liczbę binarną. Podzielmy wszystkie liczby, które można przedstawić za pomocą  $n$ -cyfrowej liczby binarnej, na dwie klasy. Do pierwszej niech należą te, których suma cyfr jest parzysta, do drugiej zaś te, których suma cyfr jest nieparzysta. Załóżmy, że tylko liczby parzyste są poprawne, a nieparzyste — błędne. Gdy jedna z cyfr którejkolwiek liczby przypadkowo zmieni swą wartość, liczba stanie się nieparzysta, co pozwala wykryć błąd. Zazwyczaj w tym prostym przypadku przedstawiamy liczbę w sposób następujący:

$xxxx...xx \ ?$

liczba                      miejsce na znak sprawdzający

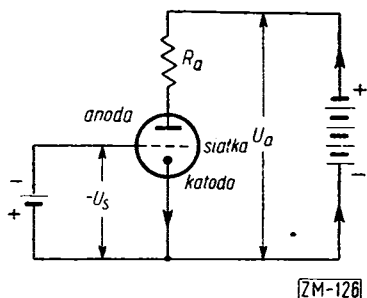
Rozwijając ostatnią metodę można zbudować kody do przedstawiania liczb, które nie tylko umożliwiają znalezienie więcej niż jednego błędu, ale pozwalają określić, w którym miejscu powstał ten błąd. Przez łączenie wymienionych metod otrzymujemy praktyczne rozwiązania stosowane w automatycznych maszynach cyfrowych [8].



### 3. Realizacja techniczna

**3.1. Wiadomości wstępne.** W celu przypomnienia czytelnikowi pojęć z radiotechniki, które mogą się okazać potrzebne do zrozumienia dalszej części artykułu, podaje kilka informacji w tej sprawie.

Podstawowym elementem w elektronowych maszynach cyfrowych jest lampa elektronowa. Najprostszą lampą elektronową jest dioda; składa się ona tylko z dwóch elektrod, anody i katody. Istotną cechą diody jest własność prostowania, tzn. przewodzenia prądu tylko w jednym kierunku: od anody do katody. Dla prądu w kierunku odwrotnym opór, jaki przedstawia dioda, jest bardzo duży. Gdy między katodę a anodę wprowadzimy jeszcze jedną elektrodę, tzw. *siatkę*, otrzymamy lampę trójelektrodową — triodę. Działanie siatki możemy przyrównywać do zaworu. Gdy do anody przyłożymy dodatnie napięcie  $U_a$ , jak pokazano na rysunku 4, wtedy przez lampę popłynie prąd. Natężenie jego jest zależne od napięcia  $U_s$  przyłożonego do siatki i mającego zmieniać się od 0 do  $-U_{s,max}$ . Przy napięciu  $-U_{s,max}$  prąd płynący przez lampę wynosi 0. Przy napięciach  $|U_s|$  większych od  $U_{s,max}$  stan ten się nie zmienia. Stan ten nazywamy *zatkanie lampy*. Stan przewodzenia nazywamy *odetkaniem lampy*. Stosując więcej siatek niż jedną otrzymujemy lampę wielosiatkową. Poszczególne siatki działają niezależnie od siebie, podobnie jak zawory połączone w szereg.



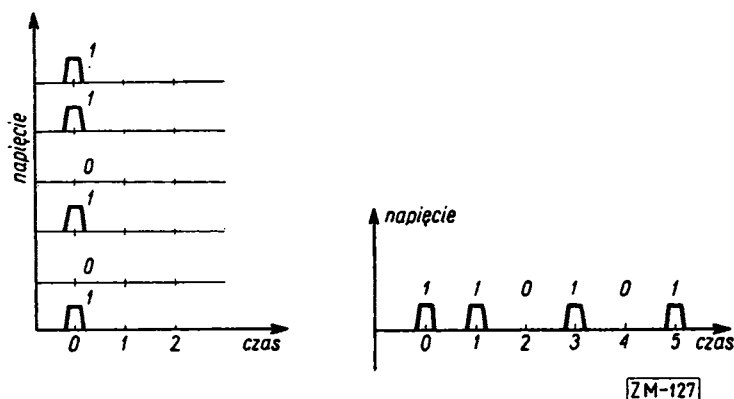
Rys. 4. Schemat połączeń triody

W maszynach cyfrowych lampy pracują przeważnie tylko w dwóch stanach: przewodzenia i nieprzewodzenia. Przez włączenie w obwód anodowy oporu  $R_a$  otrzymujemy urządzenie wzmacniające. Z anody lampy możemy otrzymywać napięcie w postaci impulsów. Impuls elektryczny używany w maszynach cyfrowych jest to funkcja jednostkowa o określonym czasie trwania. Impuls może być dodatni lub ujemny; czyli — jak mówimy — może mieć polaryzację dodatnią lub ujemną.

Stosuje się dwie zasadnicze metody realizacji maszyn cyfrowych: realizacja równoległa i szeregową. W realizacji równoległej każdej cyfrze w liczbie przyporządkowujemy jeden przewód. Cała liczba jest wtedy przesyłana lub przekształcana równocześnie.

W realizacji szeregowej (rys. 5) przesyła się liczbę za pomocą jednego przewodu cyfra za cyfrą, a położenie impulsu w czasie jest ściśle określone; różnym cyfrom odpowiadają różne położenia impulsów w czasie. Przy realizacji szeregowej konieczna jest synchronizacja poszczególnych

operacji i synchronizacja działania całej maszyny. W tym celu maszyny szeregowo mają osobne urządzenia, tzw. *zegary*, które dzielą czas na pewne ściśle określone odcinki. Każdemu takiemu odcinkowi czasu jest przyporządkowana inna wartość rozwinięcia cyfrowego liczby.



Rys. 5. Równoległe i szeregowe przedstawienie liczby 101011. Równoległe przesyłanie na sześciu drutach, szeregowe na jednym

Maszyny równoległe są bardziej skomplikowane i wymagają większej ilości elementów, lecz rachują szybciej. Maszyny szeregowe są prostsze, mają mniej elementów, lecz rachują wolniej i wymagają ścisłej synchronizacji.

**3.2. Układy przełączające.** Istotą pracy maszyn cyfrowych jest brak ciągłości: w obwodach rachujących oraz w przewodach, które wchodzi w obwody rachujące czy przesyłowe, rozróżnia się tylko skończoną ilość stanów. W obecnie budowanych maszynach elektronowych jest tendencja do rozróżniania tylko dwóch stanów: obecności napięcia w przewodzie i braku napięcia, którym odpowiednio przyporządkujemy cyfry 1 i 0.

Układy przełączające można porównać z zaworami, które zatrzymują lub przepuszczają liczby znajdujące się w jednej części maszyny do drugiej. Na drogach łączących poszczególne urządzenia znajdują się tzw. *bramki*, które pod wpływem sygnałów z urządzenia sterującego rządzą pracą maszyny.

Zarówno układy rachujące, jak i sterujące są zbudowane z pewnej ilości podstawowych elementów. Przez łączenie tych elementów w większe zespoły otrzymujemy poszczególne części maszyny.

W tabelicy 6 podano symbole graficzne elementów przełączających, występujących w maszynach szeregowych pracujących w układzie bi-

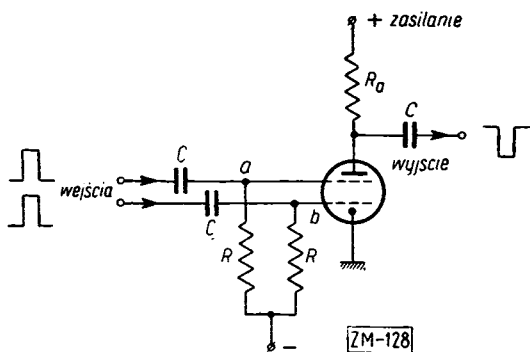
TABLICA 6

Nr	Symbol graficzny	Operacja	Objaśnienie	Symbol algebraiczny
1		rozwidlenie	sygnał pojawia się na $y$ i $z$ , gdy przyłożony jest do $x$	$y=x$ i $z=x$
2		sumowanie logiczne	sygnał pojawia się na $z$ , gdy jest na $x$ lub $y$	$z=x+y$
3		koincydencja	sygnał pojawia się na $z$ , gdy jest na $x$ i $y$	$z=x \cdot y$
4		negacja (dopełnienie)	sygnał na wyjściu, gdy brak sygnału na wejściu i odwrotnie	$z=\bar{x}$
5		koincydencja z negacją	sygnał pojawia się na $z$ , gdy jest na $x$ , a brak go na $y$	$z=y \cdot \bar{x}$
6		opóźnienie	sygnał pojawia się na wyjściu po upływie $k$ przedziałów czasu	

narnym, tak że słowa i rozkazy są przedstawione w postaci ciągu cyfr 0 i 1. Cyfra 1 jest identyczna z impulsem a cyfra 0 z pauzą. (*Rozwidlenie* jest połączeniem przewodów; elementy sumowania, koincydencji i negacji nazywamy *bramkami*; element opóźniający nazywa się *linią opóźniającą*.) Traktując te elementy, z wyjątkiem opóźniania, jako realizację funktorów rachunku zdań, możemy tworzyć *sieci liczące* [7], będące realizacją funkcji logicznych a opisujące poszczególne fragmenty maszyny.

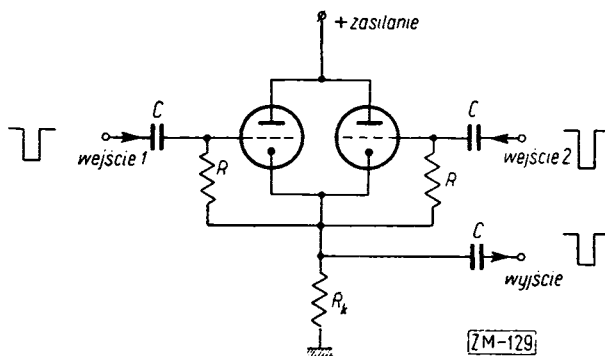
Omówimy teraz realizację techniczną ([1], [10]) poszczególnych elementów w maszynach matematycznych. Ograniczymy się tu do podania tylko kilku prostych przykładów rozwiązań.

Rysunek 6 przedstawia bramkę typu „i” dla impulsów dodatnich. Bramka jest lampą o dwóch siatkach sterujących; do obu siatek jest



Rys. 6. Bramka „i” dla impulsów dodatnich

doprowadzone z zewnątrz napięcie zatykające. Napięcie jest tak dobrane, by każda siatka przerywała prąd oddzielnie. Ponieważ przy braku impulsów lampa nie przewodzi prądu, napięcie na anodzie jest równe napięciu zasilającemu. Lampa przewodzi prąd wtedy i tylko wtedy, gdy do obu siatek przyłożymy jednocześnie impulsy dodatnie. Wtedy na wyjściu pojawi się impuls ujemny. Zmiana biegunowości impulsów nie pozwala na bezpośrednie łączenie wymienionych układów w szereg. Zwykle, gdy chcemy stosować wielokrotne łączenie szeregowo bramek tego typu,



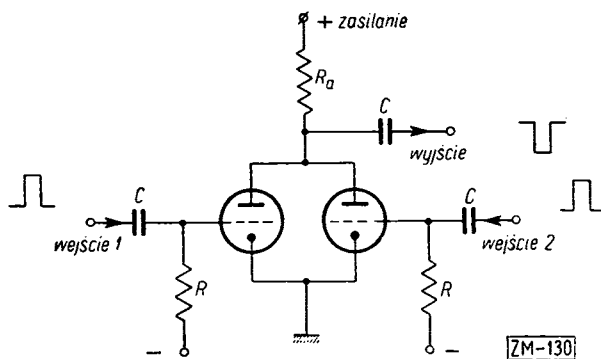
Rys. 7. Bramka „i” dla impulsów ujemnych

wtrącamy między nie jeszcze jedną lampę elektronową, pracującą jako zwykły wzmacniacz, w celu odwrócenia biegunowości impulsów. Kondensatory  $C$  stosowane w tym układzie służą do oddzielenia dalszych układów od stałych napięć zasilania tej bramki.

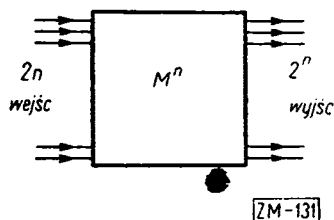
Drugim przykładem bramki „i” jest układ podany na rysunku 7. Bramka ta pracuje impulsami ujemnymi. Impulsy wyjściowe z bramek tego typu są nieodwrócone i pozostają ujemne. Obie lampy w chwili spoczynku przewodzą prąd. Prąd ten na oporze katodowym  $R_k$  wytwarza spadek napięcia, tak że napięcie na katodach jest dodatnie. Gdy do jednej lampy do siatki doprowadzimy impuls ujemny, wtedy przerwiemy płynący przez nią prąd. Spowoduje to spadek napięcia na katodach. Wielkość spadku jest jednak nieznaczna, gdyż prąd w drugiej lampie wzrasta i powoduje częściową kompensację spadku. Przy doprowadzeniu do obu siatek impulsów ujemnych obie lampy przestają przewodzić prąd; powoduje to spadek napięcia na  $R_k$  i wytworzenie impulsu na wyjściu.

Przykładem bramki „lub” jest schemat na rysunku 8. Obie lampy przy braku impulsów są zatkane i napięcie na anodzie jest równe napięciu zasilania. Gdy do jednej lampy doprowadzimy impuls dodatni, spowoduje on pojawienie się sygnału na wyjściu. Impuls wyjściowy jest ujemny.

Realizacją operacji negacji może być schemat z rysunku 6 po następujących zmianach: Jedną siatkę, np.  $a$ , zamiast z napięciem zatykającym łączymy z napięciem zerowym. Drugą, tj.  $b$ , zostawiamy bez zmiany. Do wyjścia  $b$  doprowadzamy ze źródła prądu ciąg impulsów dodatnich. Otrzymujemy wtedy dla wyjścia  $a$  dla impulsów ujemnych urządzenie realizujące negację.



Rys. 8. Bramka „lub” dla impulsów dodatnich



Rys. 9. Symbol graficzny matrycy

Jeżeli do wejścia  $b$  będziemy przykładali impulsy dodatnie nie ze źródła, ale z jakiejś innej bramki czy urządzenia, a do wejścia  $a$  impulsy ujemne, to otrzymamy realizację operacji 5 z tablicy 6 ( $x$  odpowiada wejściu  $a$ ,  $y$  zaś wejściu  $b$ ).

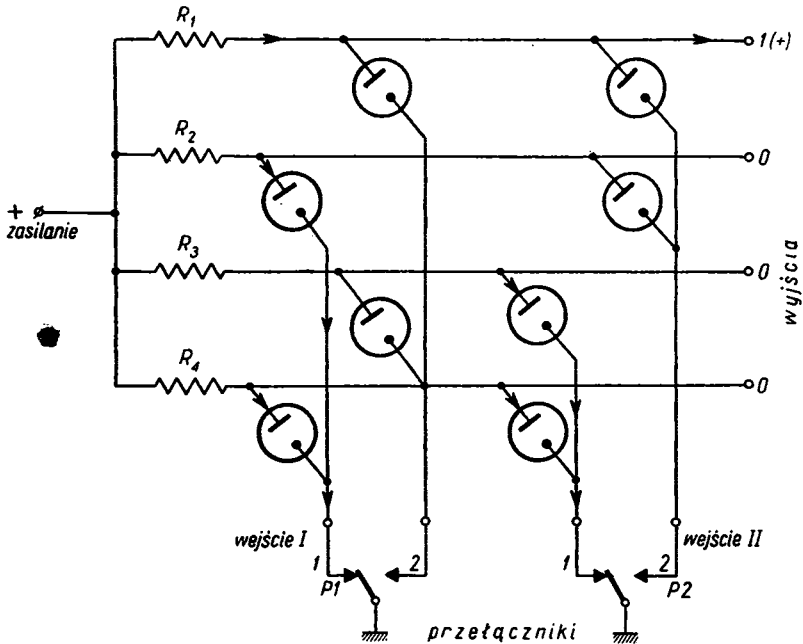
Opóźnienie realizujemy za pomocą elektrycznych linii opóźniających; przy większych opóźnieniach stosuje się również ultradźwiękowe linie opóźniające. Bliższe dane o nich podamy dalej.

Kończąc przykłady elementów przełączających opiszemy jeszcze jedno urządzenie, tzw. *matrycę* [3]. Matryca, którą oznaczamy jak na rysunku 9, jest urządzeniem o  $2n$  wejściach i  $2n$  wyjściach. Ma ona następujące własności:  $2n$  wejść tworzy  $n$  par wejść, przy czym w każdej parze tych wejść jeden przewód jest w stanie 0, a drugi w stanie 1. Dla każdej kombinacji jedno i tylko jedno wyjście ma stan 1, a pozostałe stany 0. Schemat matrycy  $M^2$ , tj. mającej cztery wyjścia i dwa podwójne wejścia, podano na rysunku 10. Urządzenie to pracuje następująco:

Niech przełączniki  $P$  będą jak w pozycji podanej na rysunku. Oznacza to, że w wejściach  $I$  i  $II$  przewody  $1$  są zwarte z ziemią. W tym położeniu przełączników napięcie pojawi się tylko na wyjściu oznaczonym  $1$ . Wszystkie inne wyjścia przez diody są zwarte z ziemią, a ponieważ opór diod jest bardzo mały w stosunku do oporów zasilających  $R$ , więc napięcia na pozostałych wejściach są w przybliżeniu równe 0. Łatwo sprawdzić, że dla każdej kombinacji przełączników tylko jeden przewód

wyjściowy jest pod napięciem. Urządzenia tego typu służą do zmiany rozkazów na sygnały sterujące.

**3.3. Pamięć.** Jak już mówiliśmy, pamięć (magazyn informacji) jest jednym z najważniejszych elementów automatycznych maszyn cyfrowych. Najprostszym przykładem pamięci jest kartka papieru, na



ZM-132

Rys. 10. Schemat matrycy diodowej

której możemy utrwalić piórem pewne informacje. Informacje są odtwarzalne dopóty, dopóki znaki nie zetrą się lub papier nie zniszczy. Pamięć stosowana w maszynach jest raczej podobna do tablicy, na której zapisujemy pewne symbole. Symbole są dopóty czytelne, dopóki tablicy nie zetrzemy. Po starciu możemy na niej umieścić inne informacje.

Jest wiele rodzajów pamięci używanych w maszynach cyfrowych a badania i postęp w tej dziedzinie przynoszą wciąż coraz to doskonalsze rozwiązania. Rozwój pamięci idzie w kierunku zwiększania jej pojemności przy jednoczesnym zmniejszaniu wymiarów i zużycia energii elektrycznej oraz zwiększaniu szybkości odtwarzania i utrwalania informacji.

Pamięć możemy podzielić na statyczną i dynamiczną. Pobranie informacji z pamięci *statycznej* polega na przyłączeniu do niej odpowiednich obwodów za pomocą układów przełączających. Pamięć *dynamiczna* jest to pamięć, której informacje krążą w pewnych obwodach; informacje te

można wtedy pobrać, gdy przechodzą one przez specjalne urządzenie czytające.

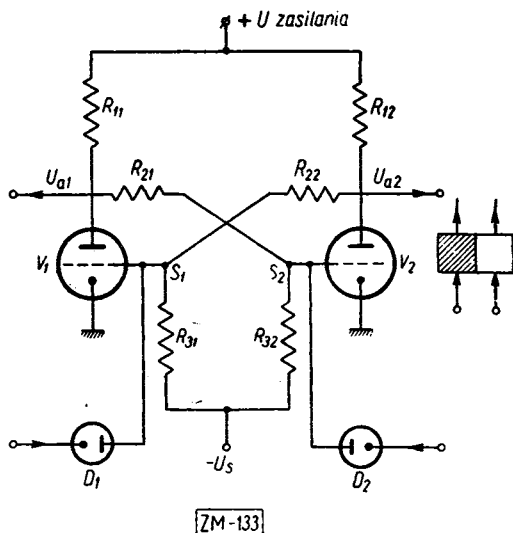
Bardzo istotną cechą charakteryzującą pamięć jest *czas oczekiwania*. Jest to okres czasu, który upływa od chwili zażądania informacji do chwili jej otrzymania. Czas oczekiwania istotnie wpływa na szybkość rachowania maszyny. Zwiększenie szybkości elektronowego urządzenia arytmetycznego niewiele przyspieszy rachunek, jeżeli czas oczekiwania wynosi np. 95% czasu trwania całego rachunku.

Najdawniejszym elementem pamięciowym w maszynach elektronowych był przełącznik elektronowy, tzw. *przerzutnik*. Schemat przerzutnika oraz jego symbolu graficznego podano na rysunku 11. Przerzutnik ma dwa stany równowagi. Urządzenie składa się z dwóch jednakowych elementów połączonych symetrycznie. Aby wyjaśnić pracę tego urządzenia, przypuśćmy, że lampa  $V_1$  przewodzi prąd. Prąd płynąc przez opór  $R_{11}$  powoduje

spadek napięcia na anodzie lampy  $V_1$ . Napięcie  $U_{a1}$  jest niskie. Napięcie  $U_{a1} - (-U_s)$  dzieli się w stosunku do oporów  $R_{21}, R_{32}$ . Opory te są tak dobrane, żeby napięcie na siatce lampy  $V_2$  było mniejsze od  $-U_s \max$ . Powoduje to zatkanie  $V_2$ . Napięcie  $U_{a2}$  jest równe napięciu  $U$  zasilania. Napięcie to przez opory  $R_{22}$  i  $R_{31}$  przenosi się na siatkę  $V_1$ . Napięcia zasilania są tak dobrane, żeby lampa  $V_1$  przewodziła. W tym stanie układ pozostaje dopóty, dopóki jakiś bodziec zewnętrzny nie przerzuci go do drugiego stanu równowagi. Stan ten charakteryzuje się tym, że  $U_{a1}$  jest niskie, a  $U_{a2}$  wysokie. Stan ten oznaczamy tak, jak na symbolu graficznym (rys. 11).

Gdy do siatki  $V_2$  przez diodę  $D_2$  przyłożymy impuls ujemny, nic się w układzie nie zmieni i przerzutnik pozostanie w stanie poprzednim. Gdy zaś impuls ujemny przyłożymy do siatki  $V_1$  przez diodę  $D_1$ , układ ten przejdzie do drugiego stanu równowagi. Proces przebiega w następujący sposób:

Impuls ujemny obniży napięcie na siatce lampy  $V_1$ , co w konsekwencji podwyższy napięcie na anodzie  $V_1$ , a dalej przez opór  $R_{21}$  podnie-



Rys. 11. Schemat przerzutnika oraz jego symbolu graficznego

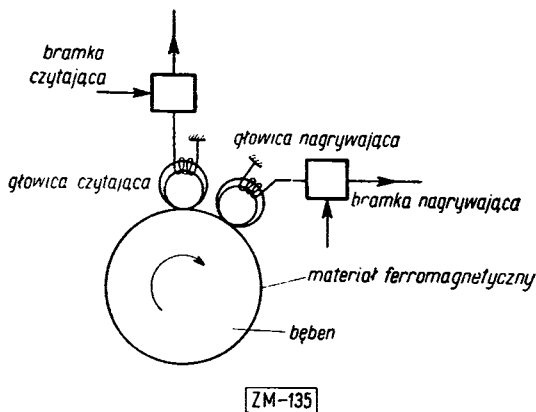




Ponieważ po każdym obiegu obwodu informacje zacierająby się coraz bardziej, tak że w końcu stałyby się nieczytelne, jest konieczne regenerowanie impulsów impulsami wzorcowymi z zegara. Sterując odpowiednio bramkami *We*, *K*, *Wy* możemy zapamiętywać, wymazywać i otrzymywać informacje.

Ilość informacji (impulsów) umieszczonych w takim urządzeniu jest zależna od długości rury i częstości powtarzania impulsów. Im więcej impulsów umieścimy w rurze, tym dłuższy jest czas oczekiwania (przy stałej częstości repetycji). Średni czas oczekiwania dla tego typu pamięci wynosi połowę czasu obiegu impulsu przez obwód.

Ilość lamp potrzebna do uruchomienia jednego zespołu składającego się z rury i generatora, wzmacniacza i bramek wynosi kilkanaście. Zazwyczaj ilość słów umieszczonych w jednej rurze jest potęgą 2; przeważnie używa się 8, 16 lub 32 słów. Przeciętna ilość impulsów w słowie wynosi około 30, przy częstości repetycji impulsów 1 MHz i długości około  $0,5\mu s$ . Z tych danych widzimy, że na zamagazynowanie 1 informacji binarnej potrzeba około 0,02 lampy.



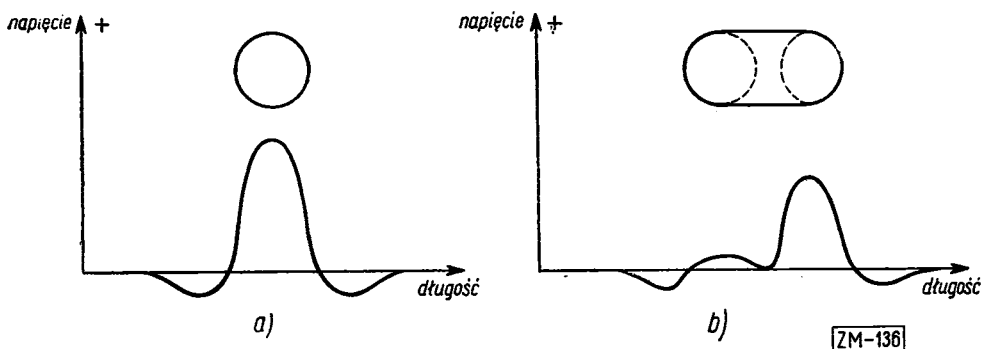
Rys. 13. Schemat pamięci magnetycznej

Pamięć magnetyczna opiera się na zjawisku histerezy magnetycznej występującej w niektórych materiałach ferromagnetycznych. Jeden z kilku znanych typów pamięci magnetycznej jest zupełnie analogiczny do metody używanej przy rejestracji dźwięków w magnetofonach.

Najbardziej rozpowszechnioną pamięcią typu magnetycznego [2] jest pamięć z wirującym bębniem. Na bębnie jest umieszczony materiał ferromagnetyczny, a dokoła niego — jak widać na rysunku 13 — znajdują się głowice nagrywające i odtwarzające. Bęben ten wiruje ze stałą szybkością rzędu 100 obrotów na sekundę. Częstość repetycji jest rzędu 100 000 impulsów na sekundę. Na ogół ilość głowic wzdłuż osi bębna wynosi kilkadziesiąt i odpowiada ilości cyfr w słowie danej maszyny. Poszczególne słowa są zapamiętywane kolejno wzdłuż obwodu. Za pomocą specjalnych urządzeń synchronizujących, pozwalających określić położenie kątowe bębna, możemy otworzyć odpowiednie bramki, czytające lub nagrywające, w żądanej chwili, a ponieważ bramki te są połączone z głowicami w wyjściu, więc otrzymujemy żądane słowo. Średni czas ocze-

kiwania tej pamięci wynosi połowę czasu potrzebnego na jeden obrót bębna. Ilość lamp potrzebnych do zapamiętania 1 informacji binarnej jest mniejsza niż w pamięci ultradźwiękowej i wynosi, w zależności od rozwiązania, 0,01-0,0005 lampy na binarną informację. Pamięć tego typu jest bardzo rozpowszechniona i nadaje się szczególnie do maszyn równoległych.

Trzecim rodzajem pamięci używanej w obecnych maszynach cyfrowych jest pamięć elektrostatyczna [12]. Pamięć ta opiera się na zjawisku gromadzenia ładunku elektrycznego na pewnych powierzchniach lub urządzeniach odpowiednio do tego przystosowanych.

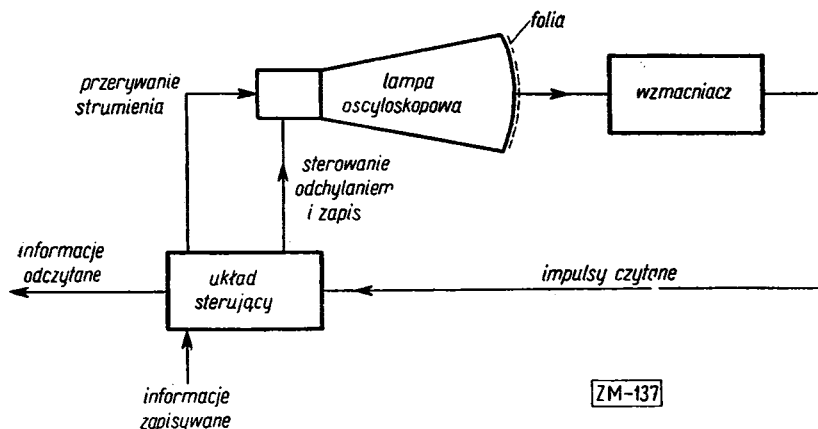


Rys. 14. Rozkład ładunków na ekranie lampy oscyloskopowej wzdłuż powierzchni ekranu: a) dla strumienia dającego kropkę; b) dla strumienia dającego kreskę (przerwane kółka oznaczają położenia kropek)

Ograniczymy się tu do omówienia tylko powszechnie używanego typu pamięci elektrostatycznej. Pamięć ta jest oparta na zjawisku emisji wtórnej powstającej na ekranie lampy oscyloskopowej pod wpływem strumienia elektronów. Jeżeli strumień elektronów padnie na ekran lampy oscyloskopowej, to elektrony te wybiją z tego ekranu elektrony wtórne. W miejscu, gdzie pada strumień elektronów, powstanie potencjał dodatni, a dookoła niego potencjał ujemny, jak widać na rysunku 14.

Schemat układu pamięciowego podano na rysunku 15. Układ składa się z lampy oscyloskopowej, na której ekranie jest umieszczona metalowa folia, tzw. *sonda*, połączona z czułym wzmacniaczem. Układ sterujący służy do przerywania strumienia elektronów oraz odchyłania jego wiązki według danych otrzymanych z maszyny. W tym rozwiązaniu istnieje wiele sposobów przyporządkowania obrazu na lampie zarejestrowanym informacjom. Opiszemy tu system „kropka — kreska”, przy czym kropka odpowiada 0, a kreska 1. Mechanizm działania jest następujący: Za pomocą systemu sterującego można skierować strumień elektronów w żądane miejsce na ekranie. Ekran dzielimy na  $n^2$  kwadracików i na

każdym z nich możemy zapamiętać jedną informację. Rozróżniamy dwa cykle pracy: cykl zapisujący (lub czytający) i regeneracyjny. Rozpatrzmy stan, gdy wszystkie informacje są równe 0: strumień elektronów na lampie oscyloskopowej skacze z kwadracika na kwadracik, odpowiednio przerywany impulsami, przez wszystkie  $n^2$  punktów i wszędzie regeneruje kropkę (co oznacza 0). Gdy strumień uderza w miejsce, gdzie była zanotowana kropka, wtedy w sondzie umieszczonej na ekranie



Rys. 15. Schemat blokowy układu pamięci elektrostatycznej

lampy powstaje mały impuls ujemny. Gdy w miejscu, w które ma następnie uderzyć strumień elektronów, była zanotowana kreska, wtedy znajdujący się tam potencjał ma wartość niższą i sonda wykryje impuls dodatni. Pojawienie się impulsu dodatniego wywołuje zadziałanie układu sterującego: strumień, który w normalnym przypadku pisał kropkę, napisze kreskę. Regeneracja impulsów jest konieczna ze względu na to, że potencjał na ekranie lampy oscyloskopowej wyrównuje się po paru dziesiątych sekundy.

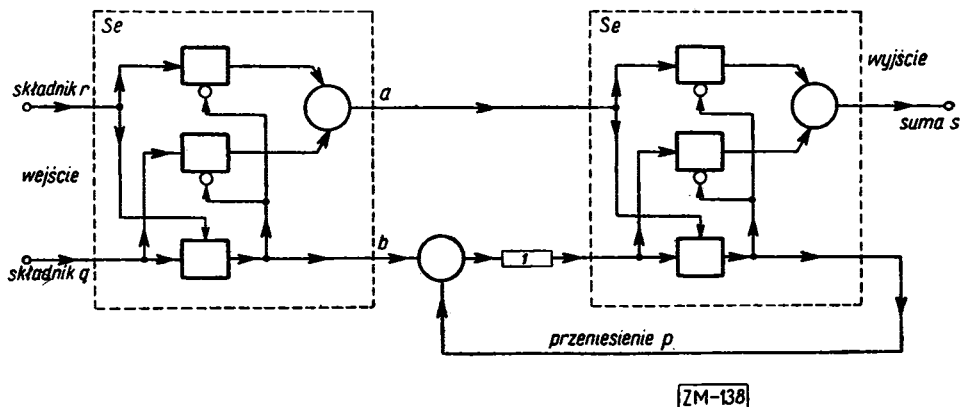
Pamięć budowana za pomocą tego typu lamp oscyloskopowych składa się zazwyczaj z tylu poszczególnych zespołów, ile cyfr ma słowo. Jednakowe współrzędne punktów na wszystkich lampach oscyloskopowych odpowiadają jednemu słowu. Słowa są czytane i zapisywane równolegle na wszystkich lampach. Przeciętna długość impulsów w tym układzie wynosi około kilku mikrosekund, czas oczekiwania jest niewiele dłuższy. Pamięć ta nadaje się zwłaszcza do maszyn równoległych.

**3.4. Przykłady rozwiązań.** Omówimy teraz realizację kilku prostych układów rachujących. Jednym z prostszych układów jest *szeregowy sumator binarny*, tzn. dodający poszczególne cyfry kolejno jedna po drugiej. Podamy dwie realizacje sumatora binarnego. Pierwsze rozwią-

zanie otrzymamy przez dwukrotne użycie sumatora częściowego. *Sumator częściowy* jest to urządzenie sumujące tylko dwie cyfry binarne. Tablica 7 podaje wszystkie możliwe stany wejść i odpowiadające im stany wyjść dla tego urządzenia. Realizację tego urządzenia przy użyciu elementów przełączających z tablicy 6 podano na rysunku 16 i oznaczono przez *Se*. Jest to tzw. *sumator częściowy*. Łącząc dwa sumatory częściowe otrzymujemy dwuwejściowy szeregowy sumator binarny (rys. 16). Linia opóźniająca na tym schemacie służy do przesunięcia cyfr o jedno miejsce binarne w lewo, czyli mnoży liczbę przez 2.

TABLICA 7

wejście <i>r</i>	0	1	0	1
wejście <i>q</i>	0	0	1	1
wyjście <i>a</i>	0	1	1	0
wyjście <i>b</i>	0	0	0	1



Rys. 16. Schemat szeregowego sumatora binarnego dwuskładnikowego

Tablica 8 przedstawia wszystkie możliwe stany wejść (wraz z przeniesieniem) i odpowiadające im stany wyjść dla sumatora dwuskładnikowego i trójskładnikowego.

Drugą realizacją jest tzw. *szeregowy sumator trójskładnikowy*. Tablicę 8 dla sumatora trójskładnikowego możemy napisać w języku ra-

TABLICA 8

Sumator dwuwejściowy	Sumator trójswejściowy	
składnik 1	składnik 1	<i>r</i> 0 1 0 1 0 1 0 1
składnik 2	składnik 2	<i>q</i> 0 0 1 1 0 0 1 1
przeniesienie	składnik 3	<i>p</i> 0 0 0 0 1 1 1 1
suma	suma	<i>s</i> 0 1 1 0 1 0 0 1
	przeniesienie	0 0 0 1 0 1 1 1

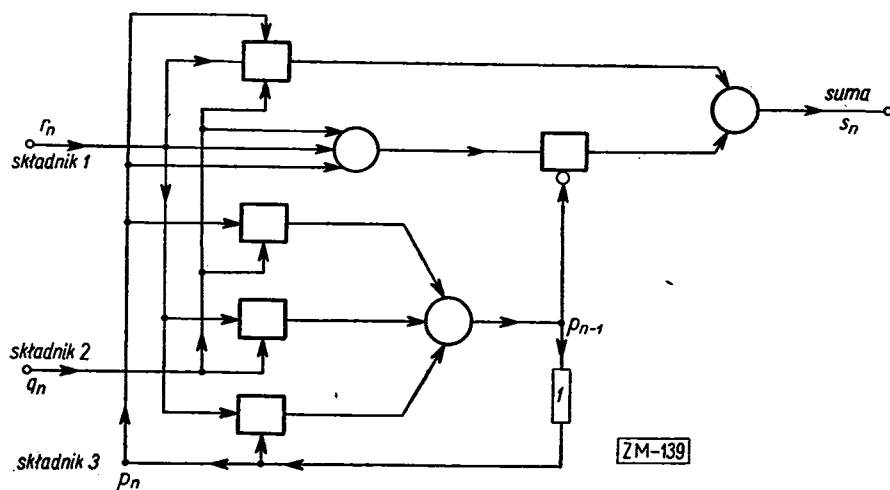
chunku zdań dla sumy w postaci

$$s_n = (r_n \cdot q_n \cdot p_n) + (r_n + q_n + p_n) \cdot ((r_n \cdot q_n) + (r_n \cdot p_n) + (q_n \cdot p_n))',$$

a dla przeniesienia w postaci

$$p_n = (r_n \cdot q_n) + (r_n \cdot p_n) + (q_n \cdot p_n),$$

gdzie  $s_n$  jest  $n$ -tą cyfrą sumy liczb  $r$  i  $q$ , a  $p_n$   $n$ -tym przeniesieniem. Schemat blokowy tego urządzenia jest podany na rysunku 17. Wstawiając zamiast symboli graficznych odpowiednie elementy (bramki lampowe) otrzymujemy urządzenia praktycznie stosowane w maszynach cyfrowych.

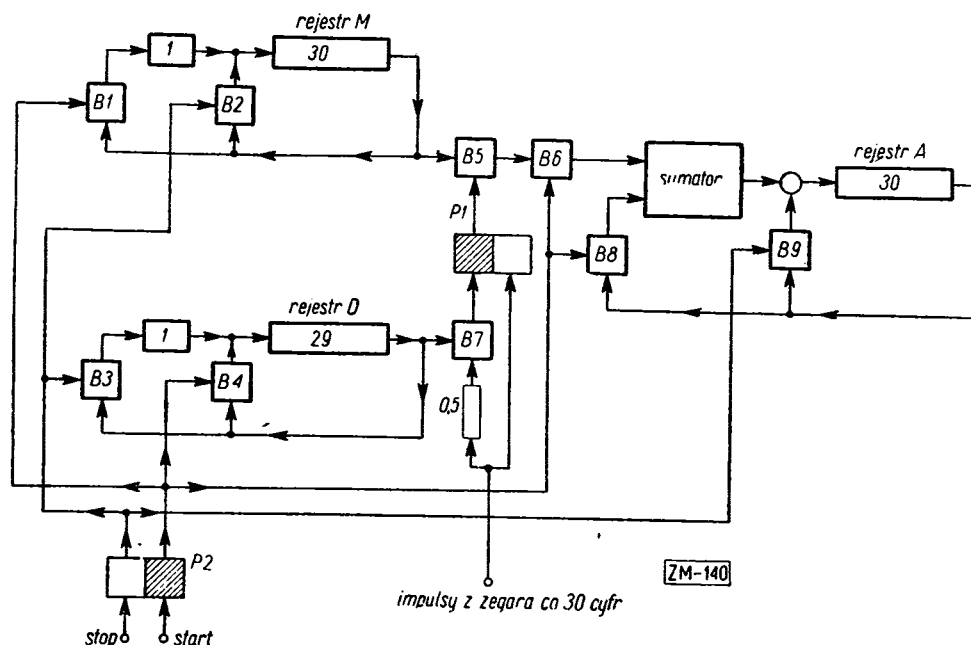


Rys. 17. Schemat szeregowego sumatora binarnego trójskładnikowego

Na zakończenie podamy jeszcze jeden przykład. Będzie to urządzenie bardziej skomplikowane, służące do mnożenia. Schemat jego jest podany na rysunku 18. Przedstawia on szeregowe urządzenia mnożące dla liczb binarnych. Mnożna i mnożnik są odpowiednio zmagazynowane w rejestrach dynamicznych  $M$  i  $D$  (mogą to być np. rejestry ultradźwiękowe). W schematach rejestrów opuszczono nieistotne elementy. Ilooczyn mamy umieścić w rejestrze  $A$ . Przerzutnik  $P2$  steruje urządzeniem; gdy znajduje się on w takim stanie jak na rysunku, urządzenie nie rachuje, a gdy jest w stanie przeciwnym, odpowiednie bramki zostają otwarte i urządzenie mnoży.

Proces mnożenia dokonuje się przez wielokrotne sumowanie i przesuwanie mnożnej. Liczba w rejestrze mnożnika również przesuwana jest za każdym sumowaniem o jedno miejsce w czasie, czyli mnożona jest przez  $2^{-1}$ . Rozpatrzmy maszynę, w której słowa są 15-cyfrowe. Cykl

jednego sumowania jest następujący: W stanie początkowym bramka  $B2$  jest zamknięta przez przerzutnik  $P1$ . Najmniej znacząca cyfra mnożnika po koincydencji z impulsem z zegara działa na przerzutnik  $P1$ . Gdy równa się ona 0, przerzutnik nie zmienia swego stanu, gdy równa się 1, przerzutnik otwiera  $B5$  i mnożna dodaje się do zawartości rejestru  $A$ . Po 15 impulsach przerzutnik  $P1$  przerzucony impulsem z zegara wraca do



Rys. 18. Blokowy schemat urządzenia mnożącego

stanu pierwotnego. Następną cyfrą mnożnika działa na przerzutnik i w zależności od tego, czy równa się ona 0, czy 1, odpowiednio go uruchamia. W ten sposób po 15 sumowaniach otrzymujemy iloczyn w rejestrze  $A$ .

### Zastosowania

Maszyny cyfrowe już obecnie mają olbrzymie zastosowanie w różnych dziedzinach nauki i życia, a w szczególności w matematyce, fizyce, chemii, mechanice, statystyce, meteorologii, astronomii, geodezji, w planowaniu gospodarczym itp.

Maszyny cyfrowe nadają się zwłaszcza do tablicowania funkcji, rozwiązywania układów równań liniowych o dużej (rzędu setek) ilości niewiadomych, rozwiązywania równań i układów równań algebraicznych

wyższych stopni, numerycznego rozwiązywania układów równań różniczkowych zwyczajnych, liniowych i nieliniowych (obliczanie wartości własnych), równań różniczkowych cząstkowych, równań całkowych, do numerycznego obliczania całek wielokrotnych, do modelowania zjawisk przypadkowych itd.

Bądź ze względu na brak odpowiedniej metody, bądź ze względów natury konstrukcyjnej, nie wszystkie przypadki podanych problemów nadają się do rozwiązania za pomocą dzisiejszych maszyn cyfrowych; np. rozwiązanie układu 10000 równań liniowych jest dziś jeszcze niemożliwe.

Obecnie podam kilka przykładów zagadnień rozwiązanych zagranicą przy użyciu takich maszyn. Oczywiście przegląd ten nie jest kompletny; podane przykłady mają tylko ilustrować niektóre możliwości zastosowań maszyn cyfrowych.

Za pomocą maszyn cyfrowych (Mark) oblicza się znane tablice funkcji Bessla wydawane przez Harvard University Press.

Maszyna UNIVAC rozwiązywała układy 300 równań liniowych w ciągu pół godziny, a równanie różniczkowe Poissona na siatce  $22 \times 22$  około godziny. W obu przypadkach dokładność wynosiła  $10^{-8}$ .

Za pomocą maszyny SWAC wykazano, że  $2^{216} + 1$  nie jest liczbą pierwszą.

W fizyce za pomocą maszyn cyfrowych badano strukturę kryształów i molekuł, rozkład pola magnetycznego we wnętrzu ziemi, zagadnienia dyfuzji neutronów, funkcje falowe atomów litu i helu itd.

Maszyny cyfrowe oddają również bardzo duże usługi przy projektowaniu różnych urządzeń technicznych. Obliczano na nich skomplikowane systemy optyczne, telekomunikacyjne układy nieliniowe z uwzględnieniem szumów, przepływy naddźwiękowe w dyszach itd.

Buduje się również specjalne maszyny dla meteorologii. Obecnie najszybsze maszyny mogą w ciągu mniej więcej 15 minut ustalić prognozę 24-godzinną dla całego kontynentu.

Maszyny cyfrowe stosuje się także w statystyce i planowaniu. Służą one między innymi do obliczania ilości surowców i półfabrykatów oraz ich rozdziału w czasie na poszczególne fabryki produkujące różne przedmioty, a także do planowania miesięcznego w budownictwie całego kraju, z dokładnością do jednego kilograma. Czas takich obliczeń jest rzędu kilku godzin.

Równoległe z rozwojem uniwersalnych maszyn cyfrowych rozwijają się metody matematyczne dotyczące zarówno struktury maszyn, jak i zagadnienia błędów, powstają nowe metody numeryczne, np. odpowiednio przystosowana metoda Monte Carlo lub metoda nadrelaksacji.

Ciekawe wydaje się scharakteryzowanie rozwoju maszyn w latach 1944-1954. W tablicy 9 są zawarte dane dotyczące tylko trzech parametrów.

TABLICA 9

	1944 r.	1954 r.
Średnia ilość arytmetycznych operacji na sekundę	0,2	30 000
Pojemność pamięci	72	2 000
Szybkość wprowadzania i przyjmowania informacji (ilość 10-cyfrowych liczb na sekundę)	1	300

Rozwój ten, jak widzimy, idzie w kierunku wzrostu szybkości rachowania i zwiększania pojemności pamięci.

Ostatnio powstają nowe dziedziny i możliwości zastosowań maszyn cyfrowych, np. są próby tłumaczenia z jednego języka na drugi (rosyjski — angielski) i zaczyna się używać maszyn jako automatycznych kartotek towarów.

Zaczęto również przystosowywać maszyny cyfrowe do automatycznej segregacji danych bibliograficznych. Wymaga to, oczywiście, odpowiednio spreparowanych kodów.

Wydaje się prawdopodobne, że za kilka lub kilkanaście lat przez zwiększenie pewności działania, zmniejszenie kosztów i wymiarów, zmianę elementów, np. przez zastąpienie lamp i przekaźników przez inne elementy, uniwersalne maszyny matematyczne staną się zwykłym wyposażeniem zarówno pracowni badawczych instytutów i uczelni, jak i laboratoriów przemysłowych.

Na zakończenie pragnę wyrazić doc. H. Greniewskiemu i mgr Z. Pawlakowi podziękowanie za cenne rady i uwagi.

#### Prace cytowane

[1] A. D. Booth, *A magnetic digital storage system*, Electronic Eng. 21 (1949), str. 234.

[2] — *The physical realization of an electronic digital computer*, Electronic Eng. 22 (1950), str. 492-498.

[3] D. R. Brown, *Rectifier networks for multiposition switching*, Proceedings of the I. R. E. 37 (1949), str. 139-147.

[4] A. W. Burks, H. H. Goldstine and J. von Neuman, *Preliminary discussion of an electronic computing instrument*, Princeton 1946.



- [5] H. H. Goldstine and J. von Neuman, *Planning and coding for an electronic computing instrument*, tom I, II, III, str. 23, 69, 68.
- [6] H. H. Goldstine and A. Goldstine, *The electronic numerical integrator and computer ENIAC*, Mathematical tables and other aids to computation 2 (1946), str. 97-110.
- [7] H. Greniewski, K. Bochenek, R. Marczyński, *Application of bi-elemental Boolean algebra to electronic circuits*, Studia Logica 2 (1955), str. 7-76.
- [8] R. W. Hamming, *Error detecting and error correcting codes*, The Bell Telephone System Tech. Journal 26 (1950), str. 147-160.
- [9] D. R. Hartree, *Computing machines and instruments*, Urbana 1949.
- [10] C. H. Page, *Digital computer switching circuits*, Electronics 21 (1948), str. 110-118.
- [11] M. V. Wilkes and W. Renwick, *An ultrasonic memory unit for the EDSAC*, Electronic Eng. 20 (1948), str. 213-280.
- [12] F. C. Williams and T. Kilburn, *A storage system for use with binary digital computing machines*, Proceedings Ins. Elec. Engrs., Part III, 96 (1952), str. 81-100.

INSTYTUT MATEMATYCZNY POLSKIEJ AKADEMII NAUK

*Praca wpłynęła dnia 21. I. 1954 r.*

Р. МАРЧИНСКИЙ (Варшава)

## АВТОМАТИЧЕСКИЕ ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ

### РЕЗЮМЕ

Статья посвящена общим проблемам вычислительных машин. Рассмотрена классификация вычислительных приборов и схематически описана работа автоматической вычислительной машины. Далее рассмотрены арифметические действия и команды, даны простые примеры программы и описано программирование. Далее объяснены: контроль точности работы машины и применяемые для этого метода.

Во второй части, посвященной реализации вычислительных машин, рассмотрены переключительные элементы и их построение, затем несколько родов памяти; даны простые примеры частей машин. В заключении приведено несколько примеров применения вычислительных машин к разным отраслям науки и техники.

R. MARCZYŃSKI (Warszawa)

## ELECTRONIC AUTOMATIC DIGITAL COMPUTERS

### SUMMARY

The paper deals with the general problems of digital computers. It gives the classification of mathematical machines and describes in a simplified manner the action of an automatic digital computer. The author discusses arithmetical opera-

tions and commands, gives simple examples of programs for the machines and describes programming. He then explains the accuracy control of the machine and the methods used in the control.

In the second part, devoted to the realization of digital computers, switching circuits and their realization, as well as a few kinds of memory, are discussed. The paper contains simple examples of elements of the machines. Finally, the author gives several examples concerning the application of digital computers in various branches of science and engineering.

---