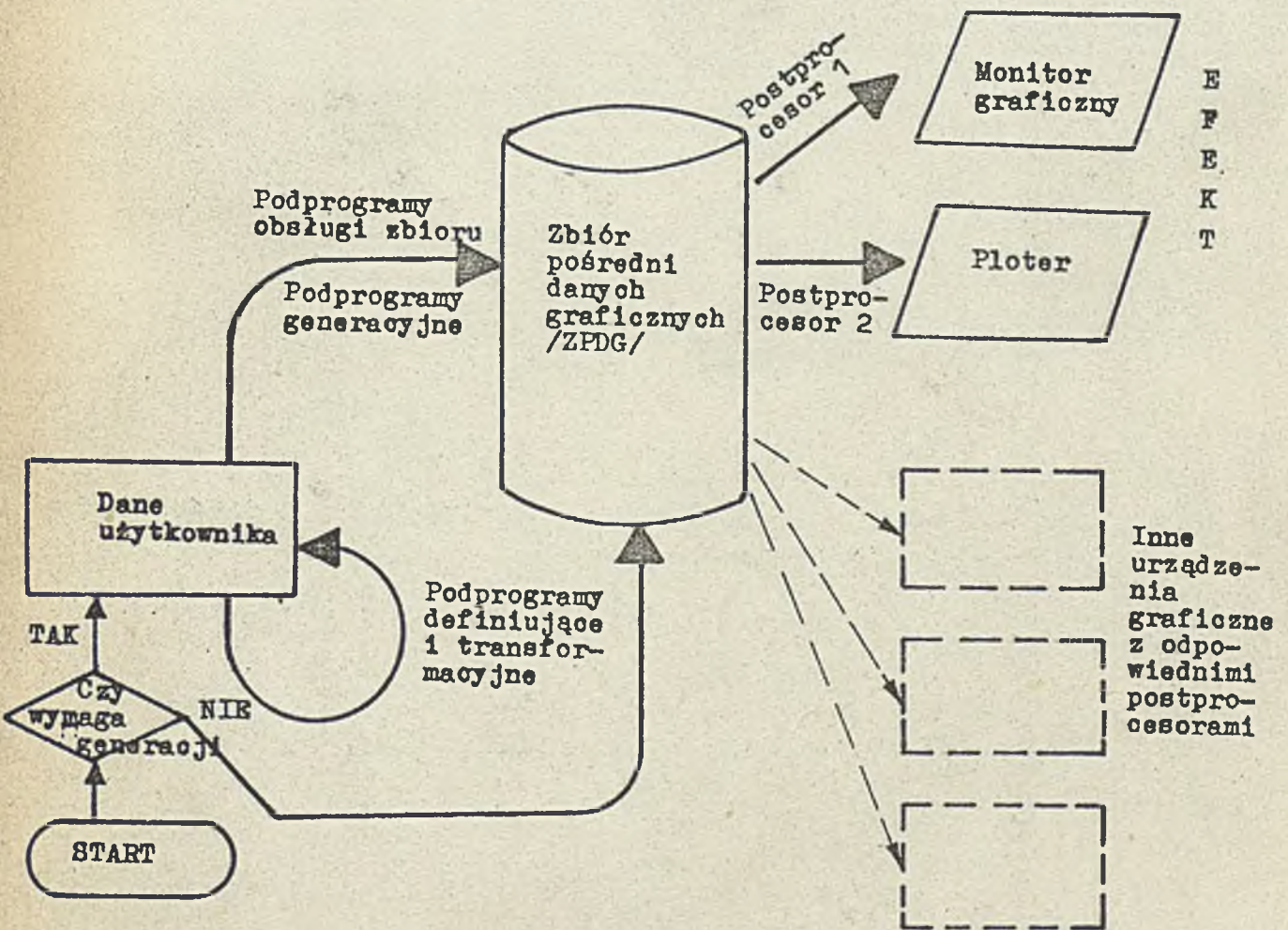


techniki komputerowe

P.3057/87
1
'87



BIULETYN INFORMACYJNY



U W A G A !

Instytut Maszyn Matematycznych przekazuje nieodpłatnie do eksploatacji sprzętowo niezależny system programowania grafiki komputerów PSG zaimplementowany na minikomputerze MERA 400.

Bliższe informacje: Pracownia Grafiki Komputerowej,
tel. 21-84-41 w. 271, 388, 428.

Rysunek na okładce: Schemat przepływu informacji w systemie graficznym: Zob.art.S.Grodzickiego i in. s.16.



P.3057/87

TECHNIKI KOMPUTEROWE

Rok XXV

Nr 1

1987

Spis treści

	str.
FAUDROWICZ A.: PROGRAF1. Opis systemu i eksploatacji	3
GRODZICKI S., KWAŚNIEWSKA G., MOKRZYCKI W., PYZEL D., STEMPOSZ E.: PSG - system programowania grafiki komputerów	15
LIBIZOWA L.W., MUCHANOW W.P.: Praktyka stosowania mikrokomputerów w szkole	45
SPRAWOZDANIA	
Trzecia Jesienna Szkoła PTI	51
Komputeryzacja w dydaktyce i badaniach pedagogicznych	61
NOWOŚCI TECHNICZNE	63

DWUMIESIĘCZNIK

Wydaje:

INSTYTUT MASZYN MATEMATYCZNYCH
Branżowy Ośrodek Informacji Naukowej Technicznej i Ekonomicznej

Komitet Redakcyjny

dr inż. Stanisława BONKOWICZ-SITTNER (redaktor naczelny),
mgr Hanna PROEDOWSKA (sekretarz redakcji)
mgr inż. Zdzisław GROCHOWSKI
mgr inż. Zygmunt HAUSWIRT
mgr inż. Jan KLIMOWICZ
dr inż. Piotr PERKOWSKI
mgr inż. Romuald SYNAK

Adres redakcji

ul. Krzywickiego 34, 02-078 Warszawa
tel. 28-37-29, 21-84-41 w.244 - sekr.red., w.211 - red.nacz.

mgr inż. Adam FAUCROWICZ
Instytut Maszyn Matematycznych

PROGRAF 1. Opis systemu i eksploatacji

PROGRAF jest systemem przeznaczonym do projektowania obwodów drukowanych. Celem artykułu jest zaznajomienie czytelników ze stanem prac prowadzonych w IMM nad opracowaniem, wdrażaniem, eksploatacją i rozwojem systemu. Opisano strukturę systemu i bardzo krótko omówiono algorytmy. Dalej omówiono realizowane projekty i przeanalizowano je pod kątem przedstawienia możliwości systemu. Przedstawiono również zamierzenia rozwojowe.

Wstęp

System PROGRAF jest zespołem programów zainstalowanych na minikomputerze SM-4 (lub równoważnym), przeznaczonym do wspomaganie projektowania dwuwarstwowych obwodów drukowanych. Słowo "wspomaganie" jest tu bardzo ważne. Oznacza ono, że projektowanie nie przebiega całkowicie automatycznie, że autorem projektu jest człowiek, nie maszyna. System PROGRAF należy więc do klasy programów określanymi w literaturze jako CAD (Computer Aided Design - projektowanie wspomagane komputerem). Tak więc PROGRAF jest systemem interakcyjnym, w dużym stopniu zautomatyzowanym. Oprócz opisywania schematu zautomatyzowane - przynajmniej częściowo - jest prawie wszystko: rozmieszczenie elementów, prowadzenie połączeń, sporządzanie dokumentacji i wykonanie taśm sterujących urządzeniami technologicznymi. Jeżeli wyniki projektowania automatycznego nie są w pełni zadowalające, koryguje się je wykorzystując możliwości pracy interakcyjnej. W szczególności dotyczy to trasowania (prowadzenia połączeń), gdyż program automatycznego trasowania opracowano na podstawie algorytmu realizującego z założenia ok. 90% zadanych połączeń.

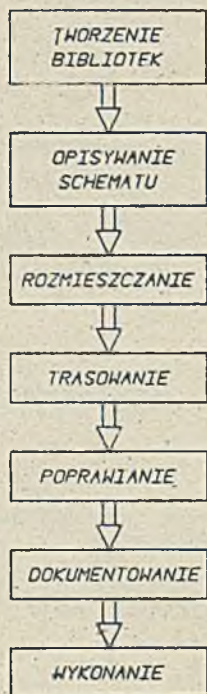
System PROGRAF jest zespołem programów użytkowych przewidzianych do wykorzystania przy użyciu systemów operacyjnych RSX 11M lub DOS-RW.

Opis systemu

Jak wspomniano we wstępie, system projektowania obwodów drukowanych PROGRAF składa się z szeregu funkcjonalnie powiązanych ze sobą programów, które można podzielić na siedem klas tak, jak to pokazano na rys. 1.

Oprócz programów, integralną częścią systemu PROGRAF jest również baza danych, w której zawarte są opisy elementów. Elementy opisane są w językach zewnętrznych i mogą być w miarę potrzeb sukcesywnie uzupełniane przez użytkowników. Jest to tzw. baza danych źródłowych. Z bazy tej wybiera się elementy potrzebne w danym projekcie, transluuje ich opisy za pomocą translatorów języków wejściowych tworząc nową bazę danych. Ta baza, zwana dalej bazą danych projektu, zawiera opisy tylko wybranych elementów. Są to opisy w kodzie wewnętrznym.

System PROGRAF ma trzy wejściowe języki zewnętrzne: JOG, JOK i JOS. Język JOG jest językiem opisu geometrycznego i służy do opisywania kształtów zarówno elementów, jak i płytek. Język JOK jest językiem opisu konstrukcyjnego i służy do wprowadzania danych konstrukcyjnych, jak np. obszarów zabronionych na płytce i innych. Język JOS jest językiem opisu schematu. Służy on do opisu własności elektrycznych zarówno projektowanego obwodu drukowanego, jak poszczególnych jego składników.

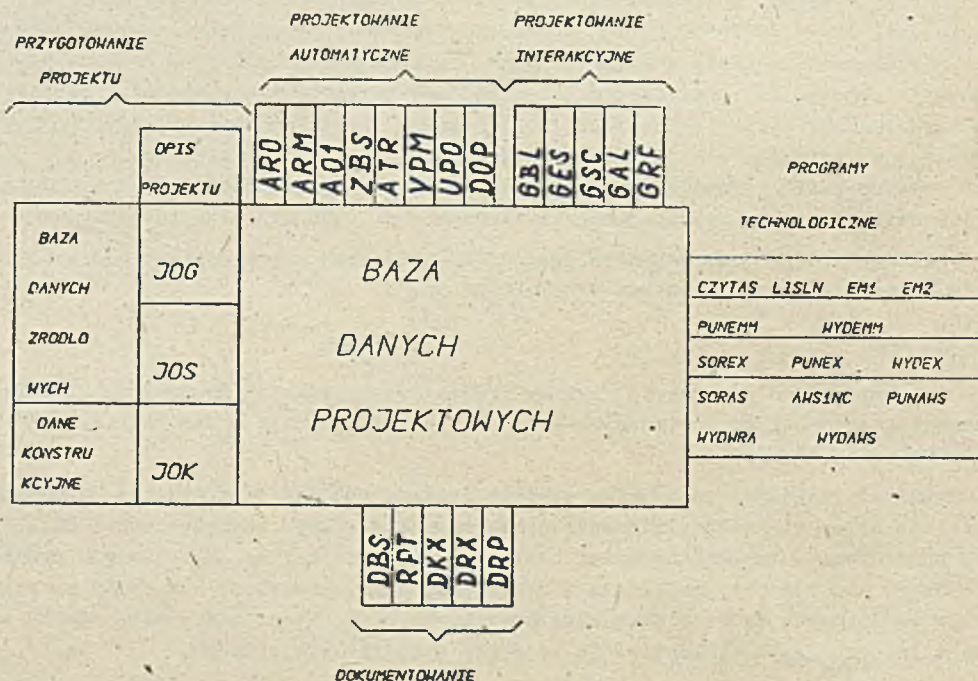


Rys. 1. Struktura procesu projektowania

Po opisanu schematu i generacji bazy danych projektowych można przystąpić do właściwego projektowania, a więc do rozmieszczenia elementów i trasowania ścieżek metalizacji. Obie te czynności można wykonywać automatycznie, bezpośrednio i w sposób mieszany. Wszystkie opisane dotąd czynności są wysoce zautomatyzowane. Przeważnie nie dają one w pełni zadowalających projektanta wyników, dlatego przewidziano możliwość interaktywnej korekty rezultatów projektowania automatycznego.

Końcowym etapem projektowania jest sporządzenie taśm (papierowych i magnetycznych) przeznaczonych do sterowania urządzeniami technologicznymi. Unika się więc etapu digitalizacji, tak uciążliwej przy projektowaniu bez użycia komputera.

Na każdym etapie projektowania można sporządzić dokumenty - wydruki i rysunki relacjonujące stan projektu. Strukturę systemu pokazano na rys. 2.

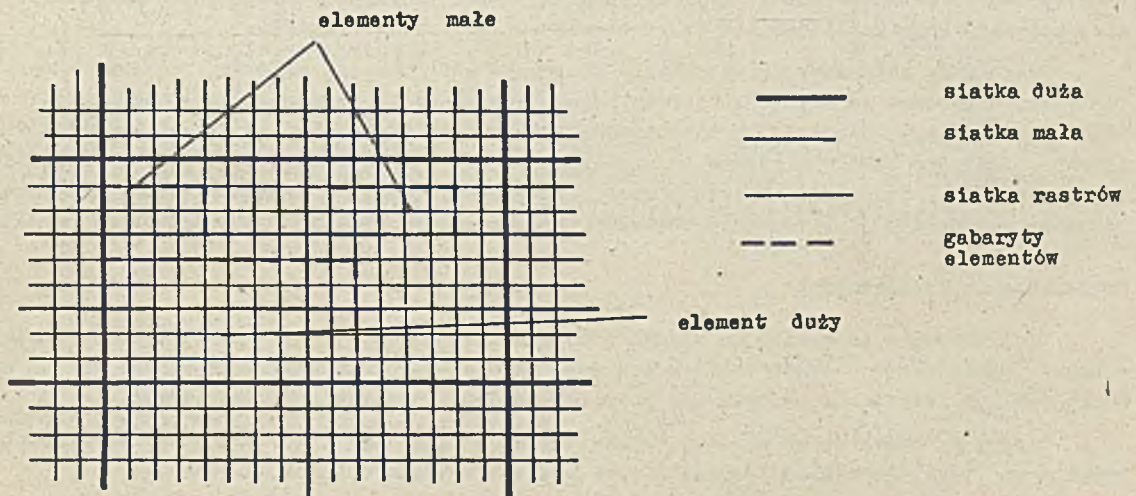


Rys. 2. Struktura systemu PROGRAF

Rozmieszczanie elementów

Rozmieszczanie automatyczne przebiega według tzw. algorytmu konstrukcyjnego [1]. Znaczący to, że elementy rozmieszczane są kolejno, przy czym kolejność wyznaczana jest przez program rozmieszczający ARO na podstawie współczynnika powiązań względem elementów już rozmieszczonych. Tak więc przed przystąpieniem do rozmieszczania automatycznego, trzeba w sposób bezpośredni umieścić przynajmniej jeden element. Rolę takich wstępnie bezpośrednio umieszczonych elementów spełniają zwykle łączówki - ich pozycje są z góry zadane.

Elementy podlegające rozmieszczeniu automatycznemu i mieszanemu mogą być umieszczane w określonych miejscach. Program rozmieszczający (ARO) przydziela danemu elementowi określone miejsce. Miejsca definiowane są jako oczka umownych siatek konstrukcyjnych. Używa się trzech siatek (rys. 3): a/ siatki rastrów, b/ siatki dużej, c/ siatki małej.



Rys. 3. Siatki konstrukcyjne

Siatka rastrów składa się z poziomych i pionowych linii odległych od siebie o odległość wielokrotność rastra podstawowego r_0 :

$$r = n \cdot r_0 ; \quad n = 1, 2, 3 \dots$$

Raster podstawowy:

$$r_0 = \begin{matrix} 0.3125 \text{ mm} & \text{w standardzie matrycznym} \\ 1/80 \text{ cala} / = 0.3175 \text{ mm} / & \text{w standardzie calowym} \end{matrix} ;$$

Siatka rastrów pokrywa płytkę automatycznie. Odległość między liniami rastrów jest jednostką odległości na płycie. Wszystkie otwory - zarówno kontakty, jak i przejścia międzywarstwowe muszą znajdować się na przecięciach linii rastrów. Siatki dużą i małą trzeba zdefiniować. Nie muszą być one kwadratowe. Przy rozmieszczeniu automatycznym pozycję elementu wyznacza lewy dolny róg oka siatki. Tak więc oczek siatki dużej powinno być co najmniej tyle ile jest dużych elementów. Oczek siatki małej musi być znacznie więcej, gdyż:

a/ duże elementy zasłaniają małe oczka (rys. 3)

b/ po wyocerpaniu się oczek dużej siatki program może umieszczać duże elementy w oczkach małej siatki.

Jak już wspomniano, elementy są rozmieszczane w kolejności wyznaczonej przez program rozmieszczający. Przed przystąpieniem do umieszczenia kolejnego elementu tworzy się listę wolnych miejsc, w których element może być umieszczony (uwzględnia się przy tym wymiary). Dalej następuje

wybór miejsca dającego minimalną wartość iloczynu długości połączeń i sumy średniego wykorzystania przejść poziomych i pionowych. I tak aż do wyzerpania się wszystkich elementów, lub wszystkich miejsc.

Rozmieszczanie mieszane polega na tym, że kolejność elementów ustala projektant.

Przy rozmieszczeniu bezpośrednim projektant może umieścić dowolny element w dowolnym położeniu.

Trasowanie

Przez trasowanie rozumie się tu realizację połączeń za pomocą ścieżek metalizowanych (tras). Trasowanie automatyczne przebiega według tzw. algorytmu promieniowego [2]. Ścieżki mogą się składać z wielu odcinków znajdujących się na różnych warstwach: pierwszej - po stronie lutowania i drugiej - po stronie elementów. Odcinki te, zwane segmentami muszą leżeć na liniach siatki rastrów. Widać, że trasy nie mogą zawierać skosów. Wykonanie skosów jest możliwe jedynie przy projektowaniu interakcyjnym. Leżące na różnych warstwach segmenty jednej trasy są łączone za pomocą otworów metalizowanych zwanych też przejściami międzywarstwowymi. Przejścia te, jak również kontakty mogą leżeć jedynie na skrzyżowaniu linii sieci rastrów.

Trasowanie automatyczne wykonywane programem ARO prowadzi segmenty poziome na warstwie pierwszej a pionowe na drugiej. Jedyne segmenty o długości jednego rastra mogą znajdować się nie na swojej warstwie. Segmenty mogą być prowadzone na sąsiednich liniach sieci rastrów. Sieci trasowane są kolejno zgodnie z nadaną im wcześniej numeracją. Porządek ten można zmienić wskazując kolejność sieci, które mają być trasowane. Postępowanie takie nazywa się trasowaniem mieszanym. Ostatnią czynnością jest usuwanie zbędnych przejść międzywarstwowych. Wykonuje to program VPM.

Projektowanie interakcyjne

Projektowanie interakcyjne służy do ostatecznej korekty projektu. Opracowane są dwie wersje: z użyciem grafoskopu - program GFR i bez jego użycia - program GAL. Obie wersje są pod względem funkcjonalnym identyczne, różnią się jedynie sposobem komunikacji z komputerem.

Operacje, które można wykonać podzielone są na dziewięć grup. Ich nazwy są prezentowane na ekranie monitora. Po wyborze jednej z grup można wykonywać wybraną czynność.

Grupy są następujące:

T - trasowanie	P - pamiętanie
M - modyfikacje	U - usuwanie
E - elementy	C - sprawdzanie
K - koniec	D - czynności pomocnicze
S - makrooperacje	

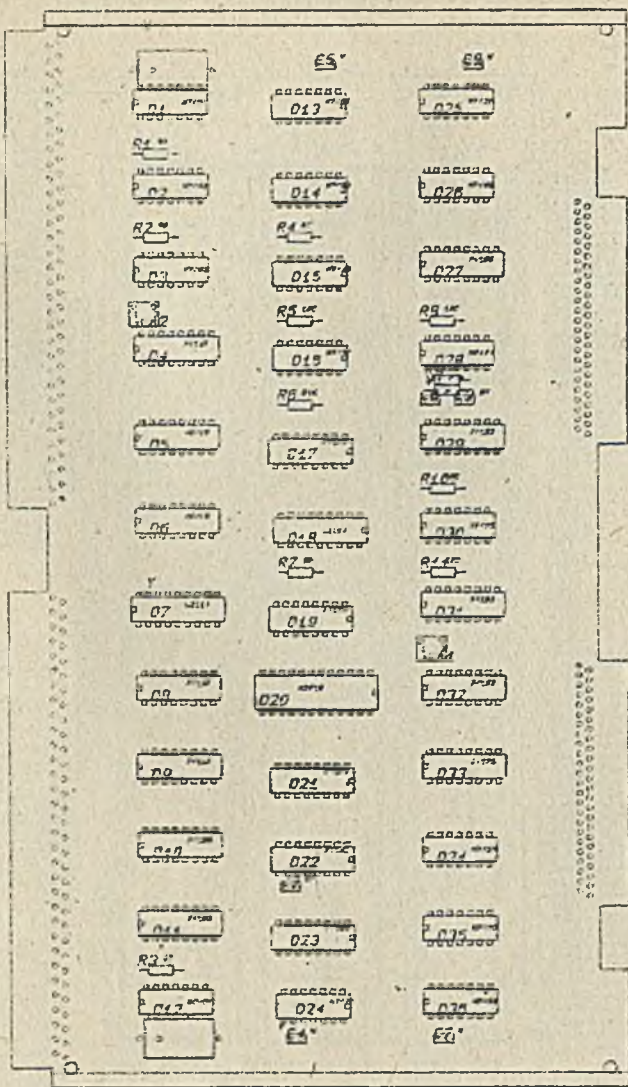
Trasowanie obejmuje: trasowanie niewytrasowanego połączenia i trasowanie ścieżki swobodnej. Ścieżka taka nie musi łączyć konkretnych punktów wymienionych w opisie schematu.

Modyfikacja obejmuje:

- retrasowanie czyli zmianę fragmentu ścieżki,
- przesuwanie segmentów i narożników,
- zmianę kodu szerokości ścieżki,
- zmianę warstwy.

Działania na elementach obejmują:

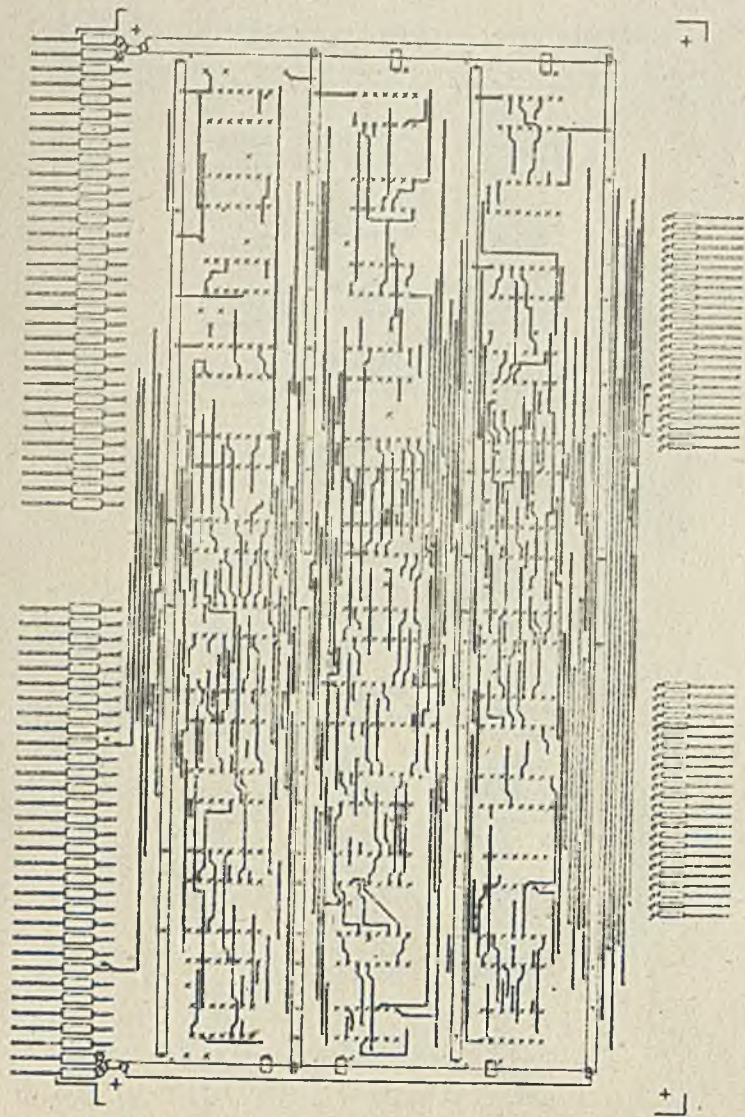
- przesuwanie
- usuwanie
- kopiowanie
- obroty
- dołączanie nowego elementu
- zmianę kodu kontaktu
- informacje o wskazanym elemencie - np: współrzędne jego kontaktów.



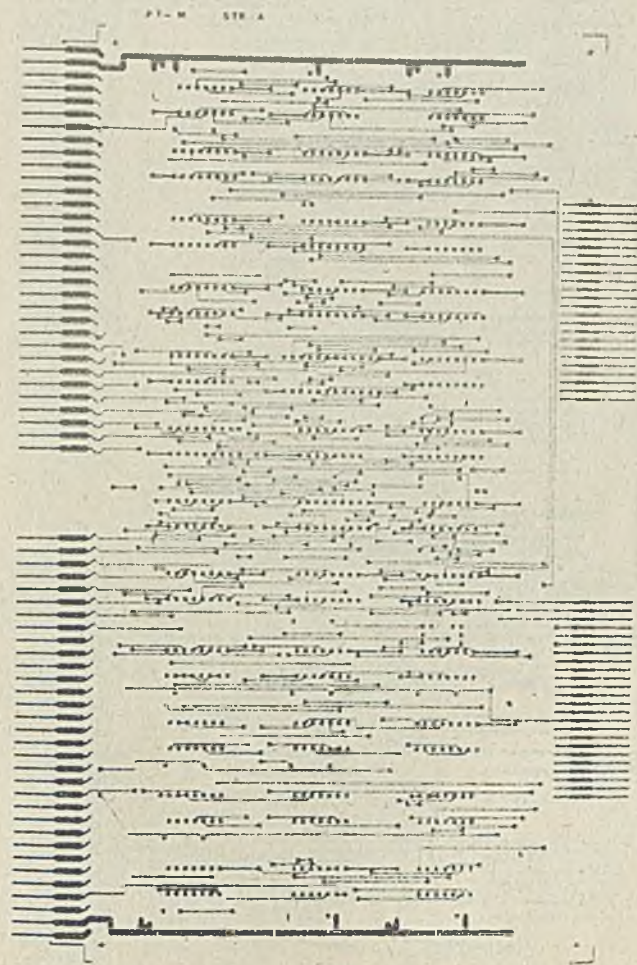
Rys. 4. Przykład A - Rozmieszczenie elementów



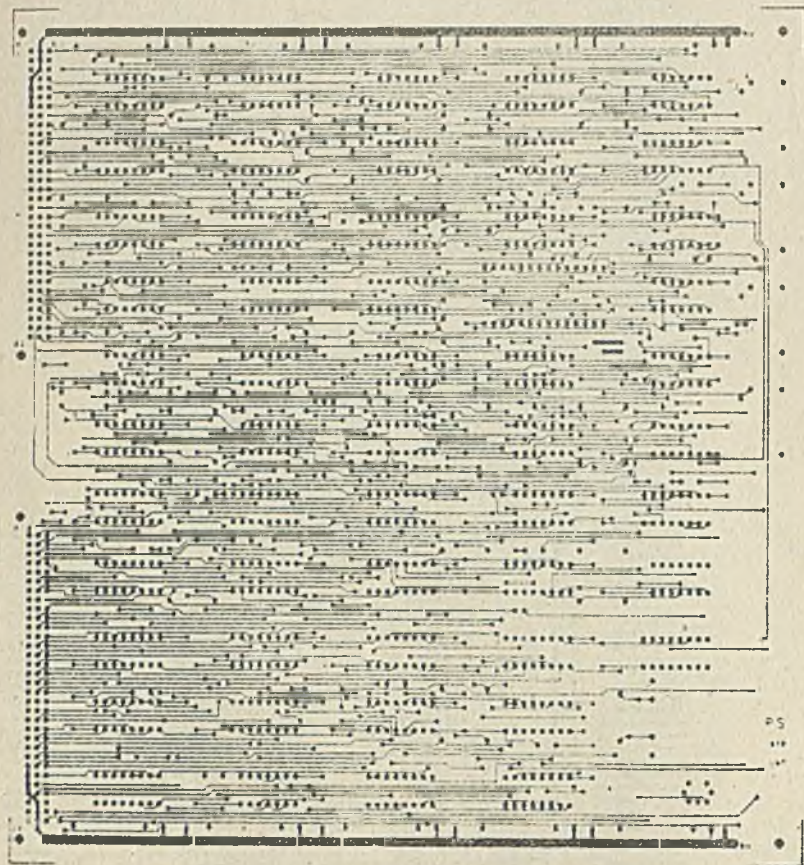
Rys. 5. Przykład A - Ścieżki metalizacji na stronie lutowania



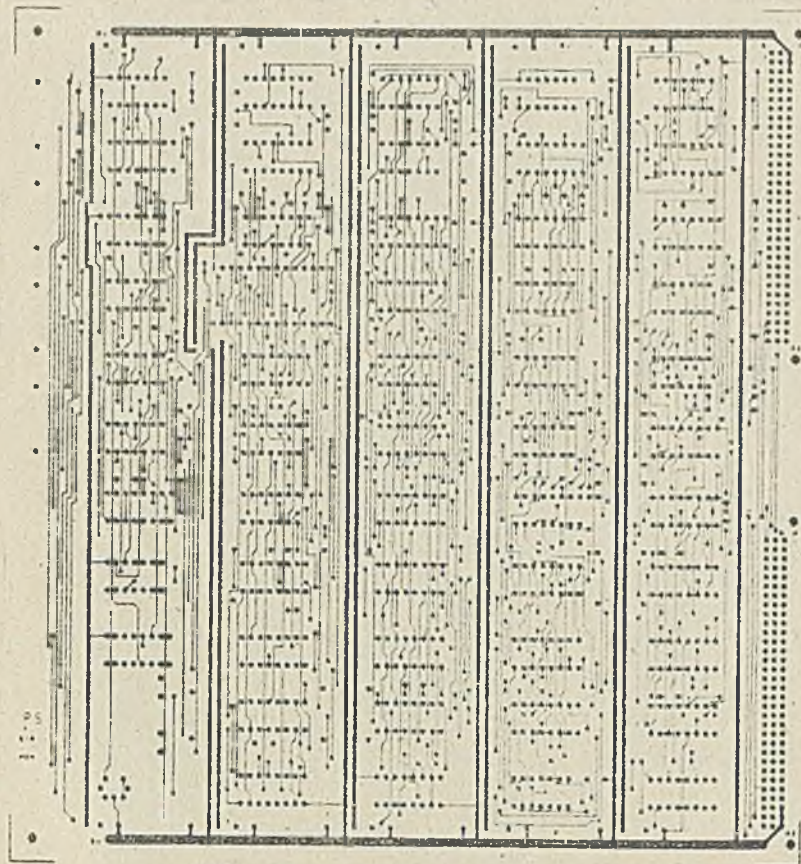
Rys. 6. Przykład A - Ścieżki metalizacji na stronie elementów



Rys. 7. Przykład A - Klisza fotozłotek na stronie lutowniczej.
Kliska wykonana fotokoprecyzyjnym fotografem FOTOMAT



Rys. 10. Przykład B - Klisza ścieżek na stronie lutowania.
Kliska wykonana fotokoordynatografem EKMA



Rys. 11. Przykład B - Kliska ścieżek na stronie elementów.
Kliska wykonana fotokoordynatografem EKMA

Makrooperacje obejmują:

- odblokowanie przejść między kontaktami układów scalonych,
- usuwanie zbędnych kontaktów,
- wykonywanie skosów.

Pamiętanie obejmuje rejestrację całego projektu, jego fragmentu lub obiektów określonego typu np: otworów czy tekstów. Usuwać można ścieżki, ich fragmenty lub przejścia międzywarstwowe.

Sprawdzanie wskazuje miejsce niezgodne z zadanymi regułami technologicznymi. Można je korygować jedną z wyżej omówionych operacji. Czynności pomocnicze obejmują odczytanie zawartości słownika, kompresję zbioru i wydruk zawartości pamięci.

Przy projektowaniu interakcyjnym z użyciem grafoskopu, do wyżej wymienionych funkcji dochodzą jeszcze operacje związane z organizacją obrazu, jak ustalenie okna, powiększenia, rodzajów linii. W zależności od rodzaju grafoskopu pracować można na tablecie lub piórem świetlnym bezpośrednio na ekranie grafoskopu.

Konwersacja z programami projektowania interakcyjnego GAL i GRF, oszkolwiek bardzo prosta, jest dokładnie opisana w podręczniku projektowania graficznego.

Dokumentowanie

Dokumentowanie polega na sporządzeniu raportów i rysunków. Raporty powstają zarówno w trakcie używania poszczególnych programów projektowania automatycznego, jak przy użyciu dwóch specjalnych programów: DBS i RPT. Program DBS sporządza raport o stanie biblioteki ścieżek. Program RPT sporządza raport o stanie projektu. Podaje więc: spis elementów, spis typów, spis sieci, współrzędne rozmieszczonych elementów, obroty.

Do wykonania rysunków służą trzy programy: DKX, DRX oraz DRP. Program DKX służy do wykonywania rysunków elementów znajdujących się w bibliotece kształtów oraz do wykonywania rysunków rozmieszczenia elementów. Program DRX służy do wykonywania rysunków ścieżek metalizacji po trasowaniu automatycznym. Program DRP służy do wykonywania rysunków ścieżek metalizacji po projektowaniu interakcyjnym. W wyniku wykonania tych programów informacje o rysunkach zostają zapisane na taśmie magnetycznej: zapis służy jako dane wejściowe dla autokreślarki.

Zbiór rozkazowy

W celu ułatwienia procesu projektowania opracowano zbiór rozkazowy prowadzący projektanta przez cały proces projektowania. Odciążony od żmudnego niekiedy wprowadzania komend operatorskich projektant, odpowiada jedynie na proste pytania o charakterze decyzyjnym. Korzystanie ze zbioru rozkazowego nie jest przy projektowaniu konieczne, lecz pomocne szczególnie wówczas, gdy projektant nie nabrał jeszcze wprawy w posługiwaniu się systemem PROGRAF.

Eksploatacja

Podjęta w połowie 1986 r. eksploatacja systemu PROGRAF łączyła w sobie następujące zadania:

- projektowanie płytek obwodów drukowanych,
- wyodrębnienie wniosków co do możliwych ulepszeń systemu i kierunków jego rozwoju,
- przeszkolenia projektantów.

Wszystkie zadania realizowano jednocześnie, projektując obwody drukowane przeznaczone do produkcji. Zaprojektowano 12 układów, w tym:

- siedem w standardzie MSWP,
- dwa w standardzie podwójnej eurokarty (E2),
- jeden nietypowy,
- dwa przeznaczone do realizacji w technologii montażu powierzchniowego.

Dalej omówimy jedynie projekty realizowane technologią tradycyjną (montaż przewlekany); projektowanie układów przeznaczonych do realizacji w technologii montażu powierzchniowego będzie przedmiotem oddzielnego opracowania.

W celu umożliwienia porównywania jakości projektowania płytek o różnych zagęszczeniach i wielkościach, wprowadzono liczne parametry określające trudności zadania i znormalizowany nakład pracy.

Przez gęstość upakowania rozumie się liczbę kontaktów przypadającą na jednostkę powierzchni projektowanego obwodu drukowanego. Oznaczając przez S pole powierzchni płytki, przez N₁ - liczbę wykorzystanych kontaktów, a przez N₂ liczbę wolnych kontaktów, mamy

$$N = N_1 + N_2/2 \quad \text{gdzie: } g \text{ jest gęstością upakowania}$$

$$g = \frac{N}{S}$$

Liczba kontaktów N jest miarą wielkości projektu, a gęstość upakowania g miarą trudności wykonania. Iloczyn tych dwóch liczb jest miarą złożoności, a więc pracochłonności projektu.

$$Z = N \cdot g = \frac{N^2}{S}$$

Dzieląc tę liczbę przez czas T potrzebny na wykonanie projektu otrzymamy parametr F będący miarą efektywności projektowania

$$F = \frac{N^2}{3T}$$

Liczba ta powinna zależeć jedynie od sprawności projektanta i efektywności systemu projektowania.

Doświadczenie pokazało, że przy gęstościach $g = 1.4 - 3.0$ kontaktów/cm², omówione wyżej parametry dobrze charakteryzują własności projektu. W tabeli 1 podano zbiorcze charakterystyki wykonanych projektów, a w tabeli 2 porównano dwa projekty, wykonane w odstępie pięciu miesięcy. Widać zarówno różnorodność projektów, jak wyraźny wzrost efektywności projektowania.

Tabela 1

Parametr	min	śr	max
N	293	664	1094
S cm ²	217	350	440
g cm ⁻²	0.86	1.95	2.76
Z cm ⁻²	251	1715	2720
F cm ⁻² . godz ⁻¹	10	17	33

Tabela 2

Data ukończenia projektu	N	S cm ²	g cm ⁻²	Z cm ⁻²	F cm ⁻² . godz ⁻¹
czerwiec 1986	818	342	2.39	2053	10
listopad 1986	1063	440	2.43	2570	33

Eksploatacja systemu PROGRAF zmusiła zespół do rozwiązania wielu praktycznych zagadnień, wypracowania metodyki postępowania, a niekiedy stosowania nawet trików.

Jednym z zagadnień było np. prowadzenie ścieżek zasilania. Sieci zasilania (VCC, GND i inne) trasuje się w sposób bezpośredni przed automatycznym trasowaniem sieci sygnałowych. Ścieżki zasilania są szerokie - obejmują niekiedy kilka rastrów. Ponieważ program trasowania automatycznego może prowadzić połączenia na sąsiednich rastrach i nie uwzględnia szerokości ścieżek metalizacji, należało opracować metodę takiego postępowania, aby podczas trasowania automatycznego nie powierał ścieżek sygnałowych z zasilającymi. Następną sprawą występującą przy projektowaniu pro-

jektowaniu projektów o dużym zagęszczeniu g była konieczność prowadzenia ścieżek gęściej niż po liniach przyjętego rastra (zob. pkt "Rozmieszczanie elementów"). Opracowano więc sposób prowadzenia ścieżek po liniach rastrów podstawowych r_0 .

Trzecim praktycznym zagadnieniem rozwiązany podczas eksploatacji było ustalenie standardów sporządzanej dokumentacji. W skład dokumentacji wchodzi m.in. klisze, wykonane na autokreślarec rysunki oraz wydruki. Na załączonych rysunkach przedstawiono próbki dokumentacji dwóch przykładów pakietów elektronicznych. Pakiet A zaprojektowano na płycie standardu MSMP ze złączami bezpośrednimi, pakiet B - na podwójnej eurokarocie mającej złącza pośrednie.

Zamierzenia rozwojowe

Rozwój systemu PROGRAF będzie przebiegał etapami. W etapie pierwszym - następującym bezpośrednio po wdrożeniu - przewidziano następujące uskokonałenia:

- opracowanie wersji systemu opartego na mikrokomputerze IBM-PC i autokreślarec (plotter) typu DIGIGRAF,
- opracowanie dwóch nowych programów trasowania automatycznego tak, aby można było automatycznie wytrasować wszystkie połączenia,
- zwiększenie możliwości dokumentowania projektu m.in. przez opracowanie programu umożliwiającego automatyczne kreślenie schematów projektowych obwodów drukowanych,
- rozwój programu sprawdzającego projekt pod względem zgodności z wymaganiami procesu technologicznego.

W etapie drugim przewidziano rozszerzenie możliwości zastosowania systemu na następujące dziedziny:

- projektowanie obwodów drukowanych z powierzchniowym montażem elementów (obwodów jednowarstwowych),
- projektowanie obwodów wielowarstwowych,
- projektowanie układów hybrydowych.

W etapie trzecim - perspektywnym - przewidziano następujące kierunki rozwoju:

- symulacja działania projektowanych układów,
- projektowanie układów scalonych,
- zastosowanie zaawansowanego sprzętu, jak np: analizatory obrazu.

Literatura

- [1] Breuer M.: Automatische Projektierung von Maschinen, PWN Warszawa 1976
- [2] Morozov K.K.: Proektirovanie montaznykh plat na VM, Moskva 1979

dr inż. Stanisław GRODZICKI
mgr inż. Grażyna KWAŚNIEWSKA
dr inż. Wojciech MOERZYCKI
mgr inż. Danuta PYZEL
mgr inż. Ewa STENPOSZ
Instytut Maszyn Matematycznych

PSG system programowania grafiki komputerów

WPROWADZENIE DO SYSTEMU PSG

System PSG został opracowany w Pracowni Grafiki Komputerowej Instytutu Maszyn Matematycznych w Warszawie. Zaimplementowano go początkowo na minikomputerze MERA 400, w systemie operacyjnym SOV-1 i sprawdzono w eksploatacji. Następnie został przeniesiony na minikomputer SX4, w systemie operacyjnym DOS R⁸. Wersja SM-owska jest rozszerzona o nowe możliwości, a jej wersja handlowa nosi nazwę PSG 2.2. Ta ostatnia wersja jest opisywana w niniejszym opracowaniu.

Przeznaczenie systemu

Zamiarem autorów pracy jest omówienie koncepcji i zasad działania systemu programowania grafiki komputerów PSG, którego pierwsza realizacja powstała przed pojawieniem się standardów międzynarodowych w dziedzinie grafiki komputerowej /m. in. GKS/. System PSG nie był dotychczas zaprezentowany szerszemu gronu specjalistów, chociaż został nabyty przez ponad 30 instytucji w okresie, gdy w kraju były produkowane urządzenia graficzne, tj. monitory graficzne MERA 7954 oraz kreślaki /ang. plotter/ KL-2.

System PSG - Procedury Syntezy Grafiki - jest sprzętowo niezależnym, ogólnogeometrycznym systemem programowania grafiki komputerów. Jest on przeznaczony do tworzenia, przekształcania, obrazowania i dokumentowania /tzn. przechowywania w postaci kopii kodowej lub trwałej/ danych graficznych. Jest więc bazowym systemem graficznym.

Obrazowanie danych graficznych może odbywać się na dowolnym wyjściowym urządzeniu graficznym komputerów: monitory graficzne i kreślaki.

Mechanizmy systemu PSG nie są zależne ani od sprzętu, systemu operacyjnego ani od języków programowania. Może on być zrealizowany w dowolnym uniwersalnym języku programowania, na dowolnym komputerze, do którego jest dołączone dowolne wyjściowe urządzenie graficzne.

Zawiera on w swej strukturze dość szeroki zestaw modułów, umożliwiających zobrazowanie na płaszczyźnie szerokiej klasy elementów geometrycznych, z których można składać w zasadzie dowolne obiekty budowlane i mechaniczne. Dlatego może on być stosowany w projektowaniu i realizacji wielu dziedzin gospodarki, np. w inżynierii lądowej, architekturze, budownictwie, komunikacji, geodezji i kartografii, w mechanice i elektronice.

Ogólna struktura systemu

Oprogramowanie graficzne systemu PSG tworzą trzy grupy podprogramów:

- Podprogramy Syntezy Grafiki 2-wymiarowej, zaimplementowane w zdecydowanej większości w Fortranie.

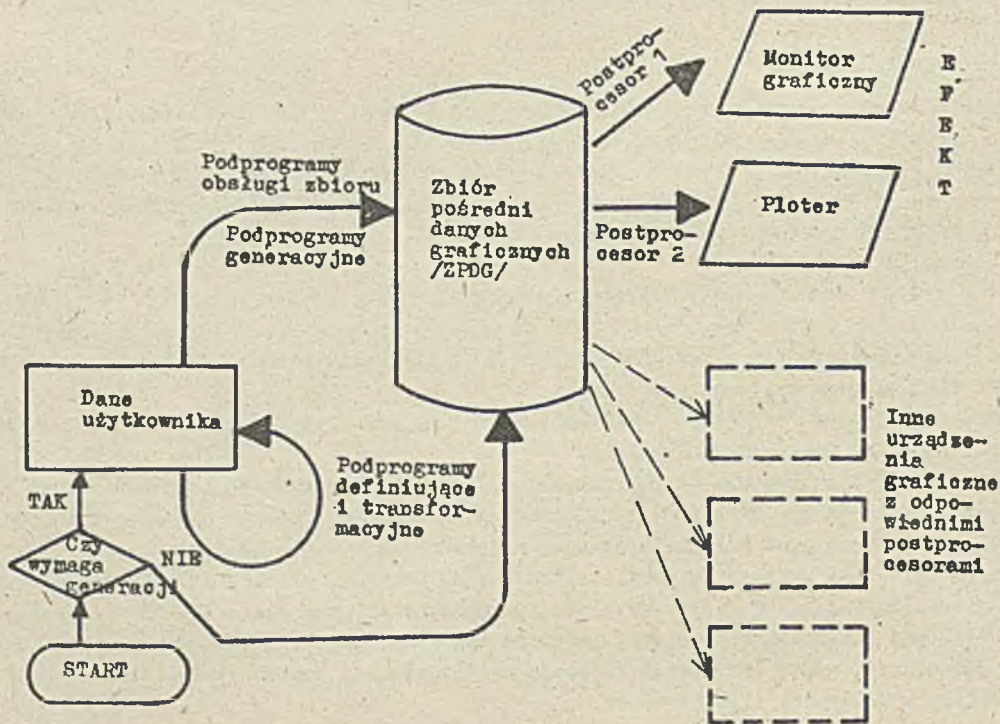
- Podprogramy obsługi Zbioru Pośredniego Danych Graficznych /ZPDG/.
- Podprogramy postprocesorów dostosowujące kody ZPDG do kodów określonego urządzenia graficznego /np. monitora graficznego/.

Syntezyując obraz z zamiarem wyświetlenia go na monitorze lub wykreślenia na kreślaku, użytkownik systemu PSG ma do dyspozycji:

- podprogramy definiowania i transformacji wybranych elementów geometrycznych, za pomocą których określa się, tworzy i przetwarza elementy geometryczne takie, jak: punkty, odcinki, okręgi, elipsy i ich łuki, krzywe algebraiczne 2-go i n-tego stopnia, pola trójkątów i czworokątów, wykresy liniowe i słupkowe, obszary grzałowe krzywych 2-stopnia, znaki alfanumeryczne i specjalne, itp.,
- podprogramy obsługi zbioru ZPDG, które pozwalają na sterowanie informacjami zawartymi w tym zbiorze,
- podprogramy generacyjne, zapisujące w zbiorze ZPDG wykreowane elementy geometryczne,
- podprogramy postprocesorów, pozwalające na pobranie ze zbioru ZPDG odpowiedniej porcji elementów geometrycznych i wyświetlenie ich na ekranie monitora bądź wykreślenie na kreślaku,
- podprogramy funkcji redakcyjnych danych alfanumerycznych, przeznaczone do formatowania i redagowania tych danych na ekranie monitora,
- podprogramy sterujące, przeznaczone do ustawiania żdanego trybu pracy monitora oraz do generowania i wysyłania danych graficznych do komputera.

Chcąc otrzymać obraz na monitorze, użytkownik musi:

- zanalizować rysunek, podzielić go na proste elementy i określić je za pomocą podprogramów definiujących i transformujących,
- za pomocą podprogramów obsługi zbioru pośredniego i podprogramów generacyjnych zapisać te elementy w zbiorze pośrednim,
- wyświetlić na monitorze informację /wywołując postprocesor monitora/ zapisaną w zbiorze pośrednim.



Rys.1. Schemat przepływu informacji w systemie graficznym

Przedstawiony na rys. 1 schemat przepływu informacji w systemie graficznym jest podstawowym schematem działania użytkownika w ramach systemu. System dopuszcza kombinacje tego procesu, np. najpierw zapisanie porcji informacji w zbiorze ZPDG, a następnie wybranie odpowiednich i ich wyświetlenie bądź wykreślenie. Zasadniczy pomysł pozostaje jednak bez zmian.

OGÓLNE ZASADY PROGRAMOWANIA W PSG

Programowanie grafiki /rysunku, obrazu/ w systemie PSG polega na zapisaniu do tzw. Zbioru Pośredniego Danych Graficznych programu graficznego, będącego ciągiem podprogramów syntezy grafiki i podprogramów obsługi ZPDG. Tak zapisany program graficzny nie zależy od urządzenia, na które zaprogramowana grafika /rysunek, obraz/ może zostać zobrazowana. Czynność zobrazowania wymaga użycia w użytkowym programie specjalnego podprogramu, zwanego postprocesorem /określonego/ urządzenia graficznego. Postprocesor przekodowuje postać grafiki ze wskazanego ZPDG na postać rozkazów wewnętrznych /określonego/ urządzenia graficznego. Postprocesor jest więc tym podprogramem, który przekształca sprzętowo niezależną postać grafiki z ZPDG na sprzętowo zależny ciąg rozkazów wewnętrznych urządzenia obrazującego.

ZPDG może być założony i przechowywany w dowolnym zbiorze o dostępie bezpośrednim. Składa się on z elementów zwanych obiektami. Obiekt jest zakodowanym pojedynczym obrazem, tzn. porcją danych graficznych w całości przesyłaną na urządzenie graficzne. Obiekt może składać się z dowolnej liczby odcinków, punktów, okręgów, elips i innych krzywych bądź ich fragmentów, pól oraz znaków alfanumerycznych i specjalnych. Wielkość obiektu zależy od użytkownika. Każdy obiekt ma nazwę, która go identyfikuje i przez którą zapewniony jest do niego bezpośredni dostęp. Ponieważ obiekty są zapisane w zbiorze kolejno, możliwy jest również dostęp przez numer /w podprogramie wyświetlającym PNR2/.

Rozpoczynając pracę mającą na celu zaprogramowanie obrazu należy wywołać podprogram ZBIOR. Zapewnia się w ten sposób dostęp do istniejącego zbioru lub możliwość założenia nowego. W jednym programie fortranowskim można zakładać kilka zbiorów. Przed otwarciem nowego zbioru należy jednak zamknąć poprzednio otwarty zbiór, wywołując podprogram KONZBI.

Do otwartego zbioru można dodawać nowe obiekty bądź rozszerzać istniejące z zachowaniem następującej kolejności operacji:

- wywołanie podprogramu OBIEKT lub OBIUNI,
- zapisanie "treści" obiektu za pomocą podprogramów generacyjnych, definiujących i transformacji,
- wywołanie podprogramu KONIEC zamykającego OBIEKT.

Definiowanie, transformowanie i generowanie obrazów i ich elementów odbywa się w prawoskrętnym układzie współrzędnych.

Między wywołaniem podprogramów OBIEKT bądź OBIUNI i KONIEC nie wolno wywoływać żadnego innego podprogramu obsługującego zbiór pośredni /SKASUJ, SCALAJ, OBROT, WEKTOR, SKALA, LUSTRO, NAZWA, OSIE, INFORM, INFZ, WYPISZ, WYPO/, ani też podprogramów wyświetlających bądź rysujących /PMR, PNR1, PNR2/. Można natomiast wstawić podprogramy generacyjne, definiujące lub transformacji oraz dowolne sekwencje instrukcji fortranowskich. Przerwanie pracy pomiędzy podprogramami OBIEKT /lub OBIUNI/ i KONIEC, spowoduje, że wywołane następnie podprogramy OBIEKT czy OBIUNI, ani inne użyte za nimi podprogramy generacyjne nie pozostawią w zbiorze ZPDG śladu.

Po zamknięciu obiektu podprogramem KONIEC można dokonywać na elementach ZPDG wielu różnych czynności: transformacji obiektów /przez wywołanie podprogramów OBROT, WEKTOR, SKALA, LUSTRO/, kasować obiekty /podprogram SKASUJ/, scalać je /podprogram SCALAJ/, zmieniać ich nazwy /podprogram NAZWA/, zmieniać położenie układu współrzędnych /podprogram OSIE/, wyświetlać lub rysować obiekty /podprogram PNR1/, uzyskać informację o zbiorze i obiektach /podprogramy INFORM, INFZ i WYPISZ, i WYPO/.

Zakończenie pracy ze zbiorem wymaga wywołania podprogramu KONZBI zamykającego ZBIOR. Do zamkniętego zbioru nie można już nic dopisać. Można natomiast otworzyć nowy zbiór lub wyświetlić zamknięty zbiór, lub jego część wywołując podprogram PNR, PNR1 lub PNR2.

Niektóre podprogramy, jak np.: INFZ, WYPO i WYSLIZ nie wymagają wcześniejszego wywołania podprogramu ZBIOR i pozwalają na uzyskanie informacji o zamkniętych zbiorach /INFZ, WYPO/ bądź przesyłania poszczególnych zbiorów lub obiektów z jednego zbioru na drugi.

Sposoby programowania grafiki w systemie PSG przedstawiono w przykładach programów /rys. 2/.

<pre>PROGRAM JEDEN CALL ZBIOR ('PIERWSZY.DGR',Ø) CALL OBIEKT ('OB1',Ø) CALL GTABPB (.....) CALL GCKROB (.....) CALL GCKROB (.....) CALL KONIEC CALL OBIEKT ('OB2',Ø) DO 1Ø I=1,1Ø CALL GLCKRB (.....) 1Ø CONTINUE CALL OSIE ('OB1',.....) CALL KONZBI STOP END</pre>	<pre>PROGRAM DWA CALL ZBIOR ('PIERWSZY.DGR',1) CALL OBIEKT ('OB1',1) DO 1Ø I=1,5 1Ø CALL GCKROB (.....) CALL KONIEC CALL OBJUNI ('XX') DO 2Ø I=1,15 2Ø CALL GCKROB (.....) CALL KONIEC CALL KONZBI STOP END</pre>
--	---

Wykonanie programu JEDEN spowoduje utworzenie na dysku użytkownika zbioru pośredniego o nazwie PIERWSZY.DGR, składającego się z obiektów o nazwach OB1 i OB2, gdzie OB1 składa się z ciągu punktów /GTABPB/ i dwóch okręgów /GCKROB/, a OB2 z dziesięciu łuków /pętla/. Dla obiektu o nazwie OB1 początek układu jest w punkcie, którego współrzędne podano przy wywołaniu podprogramu OSIE, a początek układu współrzędnych obiektu OB2 jest taki, jakie było położenie tzw. współrzędnych bieżących (XB, YB)/kursora graficznego/po drugim wywołaniu podprogramu GCKROB.

Wykonanie programu DWA spowoduje dodanie do zbioru PIERWSZY.DGR /założonego programem o nazwie JEDEN/, do obiektu OB1, pięciu okręgów oraz dodanie obiektu o nazwie XX składającego się z 15 okręgów. Obiekt OB2 pozostał w zbiorze bez zmian.

PROGRAM TRZY

```
CALL WYSLIZ (3, 'DK2: [15Ø,1Ø] DRUGI.DGR',Ø,2, 'PIERWSZY.DGR', 'OB2')
STOP
END
```

Teraz są założone dwa zbiory. Na dysku użytkownika zbiór PIERWSZY.DGR złożony z obiektów OB1, OB2 i XX, oraz na dysku DK2, pod kontem [15Ø,1Ø], zbiór DRUGI.DGR, zawierający jeden obiekt o nazwie OB2.

PROGRAM CZTERY

```
CALL INFZ (5, 'PIERWSZY.DGR')
CALL INFZ (5, DK2:[15Ø,1Ø]DRUGI.DGR')
CALL PPR (4, 'PIERWSZY.DGR')
STOP
END
```

Program ten spowoduje wyprowadzenie informacji o zawartości wcześniej założonych zbiorów na urządzenie 5 oraz wyświetlenie na urządzeniu 4 zbioru DRUGI.DGR.



Program PIEC w celu utworzenia obrazu, jak na rys. 2, generuje obiekt o nazwie A, złożony z dwóch odcinków i łuku koła, tworzących jeden płatek mniejszego kwiatka. Przez wywołanie podprogramów OBROT i LUSTRO otrzymujemy cały kwiatek, a przez wywołanie podprogramów OBROT i SKALA otrzymujemy drugi, większy kwiatek. Właściwe położenie obu obiektów osiągnięto przez dwukrotne wywołanie podprogramu OSIE.

Rys. 2. Przykład rysunku wykonanego programem PIEC lub SZESC

Program SZESC tworzy ten sam rysunek co program PIEC, ale w inny sposób. Pierwszy kwiatek tworzony jest od razu przez wywołanie podprogramów GTABWB /jednokrotne/ i GLOKRB /czterokrotne/. Drugi kwiatek powstanie przez wywołanie podprogramów OBROT i SKALA, a jego właściwe położenie osiągnięto stosując podprogram WEKTOR.

FORTRAN IV V02.2-5 WED 17-SEP 86 15:10:59 PAGE 001
PIEC>TT1:=PIEC/LI:1

```
0001            PROGRAM PIEC
0002            C-----
0002            DIMENSION TAB(2,3)
0003            DATA TAB/5.,15.,0.,0.,15.,5./
0004            CALL ZBIOR('RYS1',0)
0005            CALL OBJEKT('A',0)
0006            CALL GTABWB(TAB,3)
0007            CALL GLOKRB(10.,10.,15.,5.,180.)
0008            CALL KONIEC
0009            CALL OBROT(90.,0.,0.,'R','A')
0010            CALL SKALAJ('A','A','R',0)
0011            CALL SKASUJ('R')
0012            CALL LUSTRO(0.,1.,0.,'R','A')
0013            CALL SKALAJ('A','A','R',0)
0014            CALL SKASUJ('R')
0015            CALL OBROT(45.,0.,0.,'R','A')
0016            CALL SKALA(2.,2.,'R','R')
0017            CALL OSIE('A',120.,70.)
0018            CALL OSIE('R',290.,70.)
0019            CALL KONZBI
0020            CALL PNR(4,'RYS1')
0021            STOP
0022            END
```

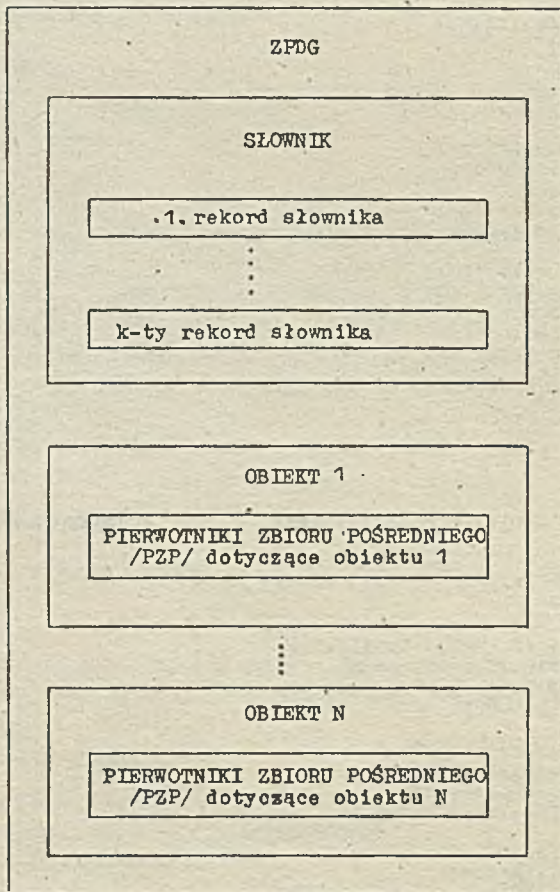
FORTRAN IV V02.2-5 WED 17-SEP 86 15:11:37 PAGE 001
SZESC>TT1:=SZESC/LI:1

```
0001            PROGRAM SZESC
0002            C-----
0002            DIMENSION TAB(2,8)
0003            DATA TAB/5.,15.,-5.,-15.,-15.,-5.,15.,5.,
0004            *15.,-5.,-15.,5.,-5.,15.,5.,-15./
0004            CALL ZBIOR('RYS2',0)
0005            CALL OBJEKT('A',0)
0006            CALL GTABWB(TAB,8)
0007            CALL GLOKRB(10.,-10.,5.,-15.,180.)
0008            CALL GLOKRB(10.,10.,15.,5.,180.)
0009            CALL GLOKRB(-10.,10.,-5.,15.,180.)
0010            CALL GLOKRB(-10.,-10.,-15.,-5.,180.)
0011            CALL KONIEC
0012            CALL OSIE('A',120.,70.)
0013            CALL OBROT(45.,0.,0.,'R','A')
0014            CALL SKALA(2.,2.,'R','R')
0015            CALL WEKTOR(170.,0.,'R','R')
0016            CALL KONZBI
0017            CALL PNR(4,'RYS2')
0018            STOP
0019            END
```

CHARAKTERYSTYKA ZBIORU POŚREDNIEGO DANYCH GRAFICZNYCH

W systemie programowania grafiki komputerowej PSG użytkownik tworzy rysunek i zapamiętuje go w uogólnionej postaci w zbiorze bezpośredniego dostępu. Zbiór ten, zwany Zbiorem Pośrednim Danych Graficznych /ZPDG/, jest zbiorem dyskowym, odpowiednio zakodowanych danych graficznych, które mogą być przesyłane na graficzne urządzenia wyjściowe.

Informację graficzną można przedstawić na ekranie monitora lub na kreślaku jako ciąg obrazów, które ze sobą mogą być logicznie powiązane. Obraz jest tworzony z podstawowych elementów graficznych, zwanych pierwtnikami geometrycznymi lub graficznymi /PG/. Pierwtnikami graficznymi są: punkty, znaczniki, odcinki, okręgi, łuki okręgów, elipsy, łuki elips, znaki alfanumeryczne i specjalne, siatki prostokątne, wykresy liniowe i słupkowe oraz układy współrzędnych.



Rys. 3. Ogólna budowa zbioru pośredniego danych graficznych /ZPDG/

Zbiór ZPDG składa się z słownika i z ciągu obiektów. Obiekt /zakodowany pojedynczy obraz/ jest zbiorem pierwtników graficznych. W skład obiektu wchodzi pierwtniki zbioru pośredniego /PZP/, które są najbardziej złożonymi elementami, jakie mogą być realizowane sprzętowo na dostępnych urządzeniach graficznych. PZP służą zarówno do opisu pierwtników graficznych /np. PG elipsa jest opisany przez PZP łamana/, jak i atrybutów dotyczących PG. Atrybuty opisują cechy związane z PG takie, jak np. kolor, typ linii, wzór do wypełniania wnętrza. Ogólną budowę ZPDG przedstawia rysunek 3.

Użytkownik ma możliwość sterowania strukturą ZPDG dokonując operacji na całych obiektach, które są niepodzielną porcją informacji graficznej. Realizuje się to za pomocą podprogramów obsługi zbioru pośredniego, które umożliwiają dopisywanie nowych obiektów, usuwanie obiektów, scalanie dwóch obiektów w jeden, transformację obiektów /obrót, przesunięcie, odbicie lustrzane, skalowanie/, przenoszenie obiektów z jednego zbioru do drugiego. Za pomocą tych podprogramów można również zrealizować zmianę położenia punktu początkowego układu współrzędnych obiektu, w celu dostosowania go do układu współrzędnych wyjściowego urządzenia graficznego.

Obiekty w PSG identyfikowane są przez nazwę lub numer. Nazwy są nadawane obiektem przez użytkownika, a numery są przy-

dzielane automatycznie zgodnie z kolejnością tworzenia obiektów.

ZPDG jako integralna część systemu, warunkująca niezależność sprzętową PSG, składa się ze słownika oraz zapisanych kolejno pierwtników zbioru pośredniego. Słownik zawiera informację o miejscu zapisu kolejnych obiektów, natomiast pierwtnik zbioru pośredniego, dotyczący danych jednego typu, składa się z nagłówka oraz ciągu danych, które są umieszczone w kolejnych rekordach.

Budowa słownika ZPDG

Słownik składa się z rekordów stałej długości - 256 słów, czyli 512 bajtów. Liczba rekordów słownika jest uzależniona od liczby obiektów w zbiorze. Każdemu rekordowi słownika odpowiada tablica: $SL(4,32)=R(128)$, w której można umieścić ciąg składający się ze 128 liczb rzeczywistych. Zawartość pierwszego rekordu słownika ZPDG przedstawiono w tabeli 1.

Tabela 1. Zawartość pierwszego rekordu słownika ZPDG

1	Liczba obiektów w zbiorze	Numer ostatniego rekordu w zbiorze	*	*
2	nazwa obiektu 1	numer rekordu początkowego obiektu 1	$x_{\emptyset,1}$	$y_{\emptyset,1}$
3	nazwa obiektu 2	numer rekordu początkowego obiektu 2	$x_{\emptyset,2}$	$y_{\emptyset,2}$
1+1	nazwa obiektu i	numer rekordu początkowego obiektu i	$x_{\emptyset,i}$	$y_{\emptyset,i}$
1+2	nazwa obiektu i+1	numer rekordu początkowego obiektu i+1	$x_{\emptyset,i+1}$	$y_{\emptyset,i+1}$
32	nazwa obiektu 31	numer rekordu początkowego obiektu 31	$x_{\emptyset,31}$	$y_{\emptyset,31}$

Pierwsze dwa pola tabeli 1, odpowiadającej pierwszemu rekordowi słownika, zawierają informacje odnoszące się do całego słownika. Informacja zawarta w pierwszym polu umożliwia odczytanie i zapisanie obiektu. Drugie pole podaje informację o liczbie wszystkich rekordów w ZPDG. Liczba ta jest równa numerowi ostatniego rekordu w zbiorze. Obie te informacje są uaktualniane podczas tworzenia zbioru. Następne dwa pola /oznaczone gwiazdką w tabeli 1/ nie są używane.

Wszystkie pozostałe czwórki pól /wiersze tabeli/ mają tę samą strukturę. Każdy wiersz dotyczy jednego obiektu w zbiorze i zawiera /dla i-tego obiektu/ następujące informacje:

- nazwę i-tego obiektu,
- numer rekordu, w którym zapisany jest początek i-tego obiektu,

• $(x_{\rho,1}, y_{\rho,1})$ - współrzędne początku układu współrzędnych, w którym jest zdefiniowany obiekt, odniesione do układu współrzędnych urządzenia graficznego.

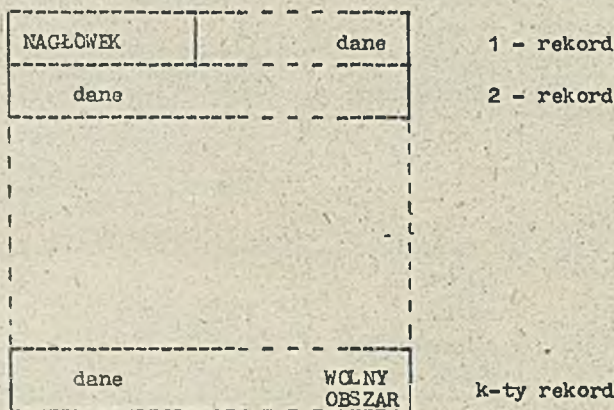
Następne rekordy słownika nie zawierają informacji ogólnych, wobec tego w każdym z nich można umieścić informacje o 32 obiektach.

Pierwotniki zbioru pośredniego

Pierwotnik zbioru pośredniego /PZP/ jest w ZPDG wyodrębnioną jednostką logiczną składającą się z ciągu rekordów. W pierwszym rekordzie każdej takiej jednostki logicznej wyróżnia się nagłówek, w którym znajduje się między innymi informacja o typie danych graficznych zawartych w danej jednostce. W ZPDG wyróżnia się 10 typów danych, a w związku z tym również 10 typów pierwotników zbioru pośredniego:

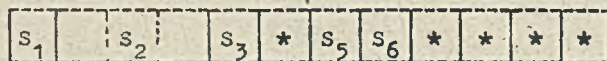
- znaczniki /w szczególności punkty/,
- odcinki prostych oraz łamana,
- okręgi /oraz łuki okręgów/,
- znaki alfanumeryczne,
- znaki specjalne,
- cechy /czyli atrybuty/ danych graficznych,
- wykres /liniowy lub słupkowy/,
- pole komórkowe /czyli obszar komórkowy/,
- ciąg komórkowy /czyli wektor komórkowy/,
- pola prostokątów.

Tabela 2. Ogólna postać PZP /pierwotnika zbioru pośredniego/



Nagłówek zawiera informacje dotyczące całego PZP. Długość nagłówka jest stała i wynosi 12 słów 16-bitowych.

Tabela 3. Postać nagłówka



NAGŁ 1 2 3 4 5 6 7 8 9 10 11 12

- Pola oznaczone na powyższym schemacie symbolami S₁ zawierają dla wszystkich typów PZP ten sam rodzaj informacji. Wartość pól oznaczonych gwiazdką, zależy od typu PZP. Pola S₁ zawierają:
- S₁ - kod typu jednostki logicznej PZP
 - 1 - ZNACZNIKI /PUNKTY/,
 - 2 - ODCINKI /ŁAMANA/,
 - 3 - OKREGI /ŁUKI OKREGÓW/,
 - 4 - ZNAKI ALFANUMERYCZNE,
 - 5 - ZNAKI SPECJALNE,
 - 6 - ATRYBUTY /CECHY/ OBRAZOWANIA,
 - 7 - WYKRESY /LINIOWY LUB SŁUPKOWY/,
 - 8 - POLE KOMORKOWE,
 - 9 - CIĄG KOMORKOWY,
 - 10 - POLA PROSTOKĄTÓW
 - S₂ - nazwa procedury generującej daną jednostkę logiczną,
 - S₃ - dla S₁ = 1,2,3,7,8,9,10
określenie rozmiaru przestrzeni,
dla S₁ = 4,5
liczba znaków do wypisania,
dla S₁ = 6
kod rodzaju informacji,
 - S₅ - liczba rekordów zawierających dane graficzne,
 - S₆ - liczba słów zajętych przez dane graficzne w ostatnim rekordzie.

Pola oznaczone gwiazdką są opisane oddzielnie dla każdego typu jednostki logicznej.

1° ZNACZNIKI /PUNKTY/

1		S ₂		2		S ₅	S ₆					dane
1	2	3	4	5	6	7	8	9	10	11	12	

Dane są ciągiem liczb rzeczywistych określających punkty, w których mają być umieszczone znaczniki.

W jednostce typu 6 są określone następujące atrybuty związane ze znacznikami:

- kolor,
- typ znacznika,
- strona pamięci,
- widoczność,
- migotanie.

2° ODCINKI ŁAMANIA

2		S ₂		2	S ₄	S ₅	S ₆					dane
1	2	3	4	5	6	7	8	9	10	11	12	

S₄ - kod interpretacji parametrów,
 $S_4 = \begin{cases} 1 - \text{dla odcinka,} \\ 2 - \text{dla łamanej.} \end{cases}$

Dane stanowią ciąg liczb rzeczywistych określających współrzędne początków i końców odcinków. Interpretacja danych uzależniona jest od zawartości pola S₄. Z graficznym przedstawieniem odcinka i łamanej wiążą się następujące atrybuty:

- kolor,
- typ linii,
- strona pamięci
- widoczność,
- migotanie.

3° ŁUKI OKREGÓW, OKREGI

3		S ₂		2		S ₅	S ₆					dane
1	2	3	4	5	6	7	8	9	10	11	12	

Dane stanowią ciągi pięciu liczb rzeczywistych określających:

- współrzędne środka okręgu,
- współrzędne punktu na okręgu /punkt ten stanowi początek łuku/,
- kąt skierowany łuku w radianach.

Z jednostką typu 3 wiążą się następujące atrybuty określone w jednostce typu 6:

- kolor,
- typ linii,
- strona pamięci,
- widoczność,
- migotanie.

4° ZNAKI ALFANUMERYCZNE

4		S ₂		2	S ₄	S ₅	S ₆	S ₇				dane
1	2	3	4	5	6	7	8	9	10	11	12	

Wyróżnione pola zawierają:
 S₇ - pierwszą współrzędną początku napisu,
 S₄ - drugą współrzędną początku napisu.

Dane są ciągiem znaków alfanumerycznych. Z jednostką tego typu wiążą się następujące atrybuty uwzględniane w jednostce typu 6:

- kolor,
- szerokość pola znaku,
- wysokość pola znaku,
- kierunek pisania znaków,
- pochylenie znaku,
- tło znaku,
- migotanie,
- strona pamięci,

5° ZNAKI SPECJALNE /GRAFICZNE/

5		S ₂		2	* S ₄	S ₅	S ₆	S ₇	S ₈	S ₉	S ₁₀	dane
1	2	3	4	5	6	7	8	9	10	11	12	

Wyróżnione gwiazdką pola zawierają następujące informacje:

- S_7 - pierwsza współrzędna początku napisu,
- S_4 - druga współrzędna początku napisu,
- S_8 - kąt zawarty między osią OX , a napisem /w stopniach/,
- S_9 - szerokość pojedynozego znaku specjalnego wraz z odstępem /w milimetrach/.

Dane są ciągiem liczb naturalnych, które oznaczają kod znaków specjalnych. Z tą jednostką logiczną wiążą się następujące atrybuty:

- kolor,
- migotanie,
- typ linii,
- widoczność.
- strona pamięci

6° ATRYBUTY

Atrybuty zawarte w jednostce typu 6 dotyczą jednostek logicznych występujących za nimi i obowiązują do chwili wystąpienia następnej jednostki typu 6, która zmieni ich wartości. Pierwotniki zbioru pośredniego określające atrybuty mają następującą budowę:

6		S_2		S_3		\emptyset	S_6						
1	2	3	4	5	6	7	8	9	10	11	12		

P_1		P_2		P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}		P_{11}	P_{12}	P_{13}	P_{14}	P_{15}		
13	14		29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45

Pola S_3 oraz P_1 /dla $i = 1, 15$ zawierają następujące informacje /przy poszczególnych atrybutach gwiazdką oznaczono wartość domyślną - standardową/:

S_3 - wskaźnik rodzaju informacji w jednostce nr 6. Jeżeli w jednostce typu 6 podawana jest informacja w polu P_k , to wskaźnik rodzaju przyjmuje wartość k czyli $S_3 = k$. Gdy zmieniane są jednocześnie wszystkie informacje - parametr $S_3 = \emptyset$. Pole S_3 zawiera liczbę całkowitą.

P_1 - wybór koloru.

Jest to liczba całkowita z przedziału 1 - 16, która umożliwia wybór koloru z ustalonej /standardowo lub przez użytkownika/ tablicy kolorów. Domyślnie przyjmowana jest wartość 13, co w standardowej tablicy barwnej odpowiada kolorowi czarnemu

P_2 - ustawienie tablicy kolorów.

Informacja o wybranym przez użytkownika zestawie kolorów znajduje się w polu o długości 16 słów 16 bitowych. Standardowo jest to tablica całkowita o wartościach od 1 do 16 w kolejnych polach.

P_3 - typ linii.

Typ linii określony jest liczbą całkowitą, która w zależności od wartości ma następujące znaczenie:

- (*) 1 - linia ciągła / _____ /
- 2 - linia przerywana / - - - - - /
- 3 - linia kropkowa / /
- 4 - linia osiowa / - . - . - . - /

P_4 - typ znacznika.

Typ znacznika określony jest liczbą całkowitą:

- (*) \emptyset - kropka .
- 1 - plus +
- 2 - gwiazdka *
- 3 - okrąg jednostkowy o
- 4 - krzyż diagonalny X

P_5 - kolor tła ekranu.

Liczba całkowita określona analogicznie, jak w P_1 . Domyślnie przyjmowana jest wartość 1 odpowiadająca kolorowi białemu.

- P_6 - migotanie.
Liczba całkowita określa:
(\times) \emptyset - migotanie wyłączone
1 - migotanie włączone.
- P_7 - widoczność.
Parametr określający podświetlenie jest liczbą całkowitą i oznacza:
 \emptyset - brak podświetlenia /zobrazowania/
(\times) 1 - podświetlenie
- P_8 - szerokość pola znaku skala wymiaru x
Liczba całkowita określająca, ile razy ma być poszerzony znak
- P_9 - wysokość pola znaku /skala wymiaru y /
Liczba całkowita określająca, ile razy ma ulec zwiększeniu wysokość znaku
- P_{10} - kąt /określony liczbą rzeczywistą w stopniach /ścieżki znaków z osią OX. Naturalny kierunek pisma ma kąt 0° .
- P_{11} - kąt pochylenia linii pionowej znaku względem ścieżki znaków określony w stopniach liczbą rzeczywistą.
- P_{12} - tło znaku.
Pole jest określone analogicznie jak pole P_1 . Liczba całkowita z przedziału 1-16 określa kolor tła znaku. Domyślnie przyjmowana jest wartość 1, co odpowiada kolorowi białemu w standardowej tablicy barwnej.
- P_{13} - wypełnianie słupków lub łączenie punktów wykresu liniowego.
Wypełnianie bądź łączenie określone jest liczbą całkowitą:
(\times) \emptyset - brak wypełniania /łączenia/
1 - wypełnianie /łączenie.
- P_{14} - baza dla wykresu słupkowego. Współrzędna linii bazowej jest liczbą rzeczywistą. Jako standardowa przyjmowana jest wartość \emptyset .
- P_{15} - strona /bufor/ pamięci określona liczbą całkowitą.
- 7° WYKRES /SŁUPKOWY LUB LINIOWY/

					\times					\times		\times
7		S_2		2	S_4	S_5	S_6		S_7	S_8		dane
1	2	3	4	5	6	7	8	9	10	11	12	

Wyróżnione gwiazdkami parametry mają następujące znaczenie:

S_4 - parametr wyboru osi ze stałym krokiem oraz wyboru rodzaju wykresu:

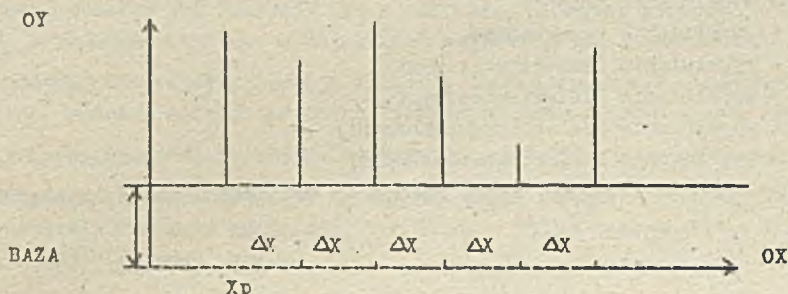
$$S_4 = \begin{cases} 1 - \text{OŚ OX} \\ 2 - \text{OŚ OY} \end{cases} \text{ dla wykresu liniowego,}$$

$$S_4 = \begin{cases} 3 - \text{OŚ OX} \\ 4 - \text{OŚ OY} \end{cases} \text{ dla wykresu słupkowego.}$$

S_7 - wartość kroku dla osi wybranej w polu S_4 /liczba rzeczywista/.

S_8 - wartość początkowa dla ustalonej współrzędnej /liczba rzeczywista/.

Tabela 4. Przykład wykresu słupkowego na bazie osi OY



Na osi OX krok jest stały i ma wartość ΔX , czyli w tym wypadku $S_4 = 3$, $S_7 = \Delta X$. Wartość bazowa jest zadawana w jednostce logicznej typu 6.

Dane dla jednostki logicznej typu 7 stanowią ciąg liczb rzeczywistych określających współrzędne po osi, które są zmienne w wykresie /w powyższym przykładzie będą to wartości z osi OY/.

Z jednostką typu 7 wiążą się następujące atrybuty /określone w jednostce logicznej typu 6/:

- baza dla wykresu słupkowego,
- wypełnianie słupków lub łączenie punktów wykresu liniowego,
- kolor,
- widoczność,
- migotanie.
- strona pamięci,

8° POLE KOMÓRKOWE * * *

8		S_2		2	S_4	S_5	S_6		S_7		S_8	
1	2	3	4	5	6	7	8	9	10	11	12	

Gwiazdkami wyróżnione są następujące pola:

S_4 - parametr wyboru wymiaru macierzy, w której zakodowany jest ciąg punktów.

Ciąg punktów w postaci macierzy kwadratowej ma być przesłany z pamięci centralnej do pamięci obrazu.

$$S_4 = \begin{cases} 0 - \text{macierz } 32 \times 32 \text{ bity,} \\ 1 - \text{macierz } 64 \times 64 \text{ bity,} \\ 2 - \text{macierz } 128 \times 128 \text{ bitów,} \\ 3 - \text{macierz } 256 \times 256 \text{ bitów.} \end{cases}$$

S_7 - współrzędna X początku przesyłanego ciągu /liczba rzeczywista/.

S_8 - współrzędna Y początku przesyłanego ciągu /liczba rzeczywista/.

Pola S_7 , S_8 określają lewy dolny róg macierzy /pierwszy przesyłany punkt/.

Dane są ciągiem 16-bitowych słów. Każde słowo opisuje 16 punktów obrazu.

Z tą jednostką logiczną wiąże się atrybut określający stronę pamięci.

9° CIĄG KOMÓRKOWY * * *

9		S_2		2	S_4	S_5	S_6		S_7		S_8	dane
1	2	3	4	5	6	7	8	9	10	11	12	

Wyróżnione gwiazdkami pola mają następujące znaczenie:

S_4 - liczba słów 16-bitowych / $S_4 \leq 127$ /.

S_7 , S_8 - współrzędne punktu, od którego ciąg ma być wpisany do pamięci obrazu.

Dane analogicznie do POLA KOMÓRKOWEGO, są ciągiem 16-bitowych słów, w których przesyłane są 4 czterobitowe porcje informacji charakteryzujące 4 komórki obrazu /grele/.

Z jednostką logiczną tego typu związany jest atrybut określający stronę pamięci.

10° POLA PROSTOKĄTÓW *

10		S_2		2	S_4	S_5	S_6					dane
1	2	3	4	5	6	7	8	9	10	11	12	

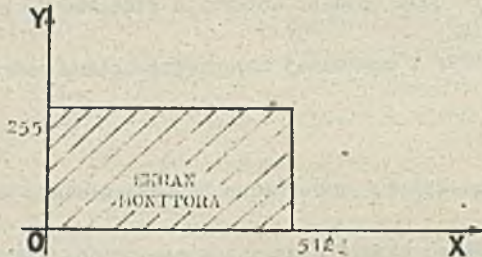
Informacja zawarta w polu S_4 umożliwia wybór podstawowego wzoru do wypełniania prostokąta:

$$S_4 = \begin{cases} \emptyset - \text{wypełnianie kolorem,} \\ 1 - \text{wypełnianie znacznikami,} \\ 2 - \text{wypełnianie liniami poziomymi,} \\ 3 - \text{wypełnianie liniami pionowymi,} \\ 4 - \text{wypełnianie liniami prawoskośnymi,} \\ 5 - \text{wypełnianie liniami lewoskośnymi,} \end{cases}$$

Eventualne rozszerzenia tego podstawowego wzoru zależą od możliwości urządzenia graficznego.

- **WEKTOR** - podprogram przesuwa obiekt o zadany wektor /przesuwa wszystkie elementy graficzne, tworzące pierwotny obiekt/.
- **SKALA** - podprogram skaluje obiekt przez zadane współczynniki skali A, B /wyskalowanie elementów graficznych obiektu na podstawie współczynników skali A, B, oddzielnie dla osi X, oddzielnie dla Y/.
- **LUSTRO** - podprogram dokonuje odbicia lustrzanego obiektu, względem prostej zadanej równaniem ogólnym. Napisy alfanumeryczne nie podlegają odbiciu lustrzanemu.
- **SCALAJ** - podprogram tworzy nowy obiekt z elementów graficznych, wchodzących w skład dwóch obiektów o zadanych nazwach. Zależnie od wartości zadanego parametru KAS podprogram kasuje lub nie poprzednie dwa obiekty.
- **OSIE** - podprogram zapisuje współrzędne początku układu obiektu do słownika zbioru pośredniego /współrzędne te są podawane względem punktu $/\beta, \beta/$ układu współrzędnych urządzenia graficznego.

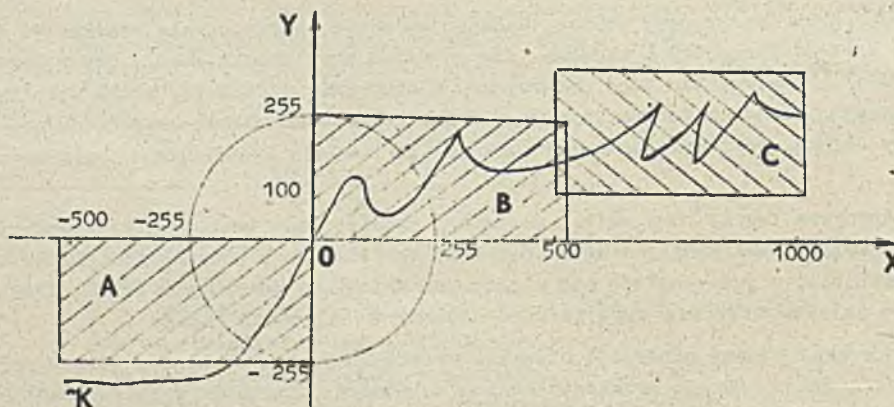
Brak podprogramu OSIE w programie użytkownika spowoduje wyświetlenie /wykreślenie/ obiektu w układzie, którego początkiem jest punkt $/\beta, \beta/$ układu współrzędnych urządzenia obrazującego.



Położenie początku układu współrzędnych obiektu można zmieniać wielokrotnie. Umożliwia to wyświetlanie lub wykreślanie fragmentów dużego obrazu /rysunku/, nie mieszczącego się w całości na ekranie monitora lub stole kreślaka. Przykład takiego działania przedstawia rys. 5.

Rys. 4. Układ współrzędnych monitora MERA 7954

Rysunek 4 przedstawia układ współrzędnych monitora graficznego MERA 7954. Natomiast na rysunku 5 przedstawiono obraz, którego treścią jest krzywa K i okrąg O, a który w całości nie mieści się na ekranie monitora. Aby wyświetlić na ekranie fragmenty A, B i C całego obrazu, trzeba trzykrotnie wywołać podprogram OSIE z odpowiednimi parametrami i trzykrotnie podprogram postprocesora monitora.



Rys. 5. Podział większego obrazu na wybrane fragmenty

Do wyświetlenia fragmentu A /rys. 5/ trzeba podać

```
CALL OSIE ('NOBI', +512., +255.)  
CALL PFR1 (LUN, 'NAZWAZ', 'NOBI')
```

Wyświetlenie fragmentu B wymaga podania

```
CALL OSIE ('NOBI', 0., 0.)  
CALL PFR1 (LUN, 'NAZWAZ', 'NOBI')
```

Natomiast do wyświetlenia fragmentu C trzeba podstawić

```
CALL OSIE ('NOBI', -500., -100.)  
CALL PFR1 (LUN, 'NAZWAZ', 'NOBI')
```

gdzie: NOBI - nazwa obiektu,

LUN - numer logiczny urządzenia, do którego jest przywiązany monitor graficzny,

NAZWAZ - nazwa zbioru.

Gdyby po każdorazowym wywołaniu podprogramu OSIE nie było wywołania podprogramu postprocesora PFR1, ale tylko raz na końcu, to wyświetliłby się tylko fragment C z ostatnimi parametrami.

- WYPISZ - podprogram wypisuje treść otwartego zbioru lub wybranego obiektu tego zbioru w takiej postaci, w jakiej jest ona zapisana w zbiorze pośrednim na dysku. Wypis treści obiektu /obektów/ może być poprzedzony wypisem słownika, zależnie od odpowiedniego parametru.
- WYPO - podprogram ten wykonuje te same funkcje co podprogram WYPISZ, ale dla zamkniętego zbioru. Wobec tego odnosi się do niego uwaga podana przy opisie podprogramu WYSLIZ.
- INFORM - podprogram wypisuje ogólne informacje o otwartym zbiorze, czyli: nazwę zbioru, numery istniejących obiektów i ich nazwy, wartości współrzędnych początków układów współrzędnych, obiektów, liczbę rekordów wchodzących w skład każdego obiektu i całego zbioru.
- IIFZ - podprogram ten wykonuje te same funkcje co podprogram INFORM, ale dla zamkniętego zbioru. Wobec tego odnosi się do niego uwaga podana przy opisie podprogramu WYSLIZ.
- NAZWA - podprogram zmienia nazwę obiektu.
- WYSLIZ - podprogram przepisuje obiekt o wskazanej nazwie lub cały zbiór /nie kasując go/ do zbioru o innej nazwie /istniejącego lub nowo otwieranego/. Podprogram operuje na zamkniętych zbiorach i dlatego przed wywołaniem go należy wywołać podprogram KONZBI, zamykający zbiór ZPIG.

PODPROGRAMY GENERACYJNE

Podprogramy generacyjne pozwalają na wykreślenie podstawowego zbioru elementów geometrycznych, z których można zbudować szeroką klasę obrazów. Elementy te nie są jednak ukierunkowane na konkretną dziedzinę zastosowań. W zbiorze podprogramów generacyjnych występują podprogramy generowania znacznika /ciągu znaczników/ w punkcie o współrzędnych określonych w sposób bezwzględny i względny, podprogramy generowania odcinka /ciągu odcinków - łamanej oraz zbioru wektorów/ również określonych w sposób bezwzględny i względny, podprogramy generowania okręgów i ich łuków, elips i ich łuków, krzywych 2-stopnia, zadanych w postaci ogólnej, podprogramy generowania pól tych krzywych, podprogramy generowania krzywych algebraicznych dowolnego stopnia zmiennych x i y /opisanych za pomocą równania uwikłanego/, podprogramy generowania krzywych gładkich i średniokwadratowych, podprogramy generowania układu współrzędnych z podziałkami liniowymi i logarytmicznymi, siatek, wykresów liniowych i słupkowych, pól komórkowych i ciągów komórkowych, a także znaków alfanumerycznych i specjalnych.

Do grupy podprogramów generacyjnych zalicza się również podprogramy określające właściwości informacji graficznej. Wybrany atrybut /właściwość/ obowiązuje dla wszystkich następujących za nim pierwotników zbioru pośredniego aż do zmiany wartości tego atrybutu.

Na zmiennych bieżących (XB, YB) obrazu, dostępnych podprogramom systemowym, są przechowywane współrzędne bieżące obrazu /kursora graficznego bądź głowicy kreślaka/.

Generacja jest rozumiana jako wpisanie parametrów danego obiektu na zbiór pośredni, co umożliwia późniejsze wyświetlenie bądź wykreślenie jego zawartości, a także archiwowanie.

Podprogramy generacyjne zapisują w zbiorze pośrednim swoją nazwę, odpowiednie wartości parametrów i aktualizują odpowiednio współrzędne bieżące (XB, YB), zależnie od spełnianej funkcji.

System PSG umożliwia generowanie następujących elementów geometrycznych, znaków alfanumerycznych i graficznych oraz atrybutów informacji graficznej, wygodnych i niezbędnych podczas składania obrazu bądź rysunku.

Generacja znacznika /śladu/ i ciągu znaczników

Podprogramy generujące znaczniki związane są z podprogramami określającymi typ i kolor znacznika oraz widoczność i migotanie. Punkt jest jednym ze znaczników.

- Generowanie znacznika zadanego w sposób bezwzględny /podprogram GSLADB/. Podprogram zapisuje w zbiorze pośrednim współrzędne (X, Y) znacznika oraz aktualizuje współrzędne bieżące obrazu (XB, YB) wartościami współrzędnych (X, Y).
- Generacja znacznika określonego w sposób względny /GSLADN/. Podprogram zapisuje w zbiorze pośrednim współrzędne znacznika $(X, Y) = (XB + DX, YB + DY)$, gdzie DX, DY są zadanymi przyrostami względem współrzędnych bieżących (XB, YB). Współrzędne bieżące (XB, YB) są aktualizowane wartościami współrzędnych znacznika.
- Generacja zadanego tablicą ciągu znaczników określonych bezwzględnie /GTARSB/. Podprogram zapisuje w zbiorze pośrednim współrzędne znaczników zadane w tablicy oraz aktualizuje współrzędne bieżące (XB, YB) współrzędnymi ostatniego znacznika z tej tablicy.
- Generacja zadanego tablicą ciągu znaczników określonych względnie /GTABSW/. W zbiorze pośrednim są zapisywane bezwzględne współrzędne ciągu znaczników. Ciąg znaczników obliczony jest względem współrzędnych bieżących przez kolejne dodawanie zadanego tablicą przyrostów. Współrzędne bieżące (XB, YB) przyjmują wartości współrzędnych ostatnio obliczonego znacznika.

Generacja wektora, zbioru wektorów oraz łamanej

Podprogramy generujące wektory i łamaną związane są z podprogramami określającymi kolor i typ linii oraz widoczność i migotanie. Wektor wyznaczony jest przez współrzędne początku i końca, natomiast ciąg wektorów reprezentowany jest tablicą zawierającą ciąg wartości początków i końców wektorów. Łamaną określa ciąg kolejnych współrzędnych wierzchołków, przez które jest ona prowadzona.

1/ Generacja odcinka /wektora/ określonego w sposób bezwzględny /GWEKTB/. Podprogram zapisuje w

zbiorze pośrednim swoje parametry - współrzędne (XP,YP) początku odcinka i współrzędne (XK,YK) końca odcinka. Współrzędne bieżące (XB,YB) są aktualizowane współrzędnymi końca odcinka.

- Generacja odcinka /wektora/ określonego względnie /GWEKTW/. W zbiorze pośrednim są zapisywane współrzędne początku odcinka (XP,YP) jako (XB,YB) i współrzędne (XK,YK) końca odcinka, obliczone, na podstawie danych przyrostów współrzędnych (DX,DY) /parametrów podprogramu/, jako $XK=XB+DX$, $YK=YB+DY$. Współrzędne bieżące (XB,YB) mają wartość obliczonego punktu (XK,YK).
- Generacja zadanego tablicą zbioru wektorów określonych bezwzględnie /GTABWB/. Ten podprogram zapisuje w zbiorze pośrednim współrzędne początków i końców wektorów, aktualizując współrzędne bieżące (XB,YB) współrzędnymi końcowymi ostatniego wektora; aby określić zbiór n wektorów należy zadać tablicę określającą 4n współrzędnych.
- Generacja zadanego tablicą ciągu odcinków /łamanej/ określonych bezwzględnie /GTABLW/. Podprogram zapisuje w zbiorze pośrednim parametry - współrzędne wierzchołków łamanej, w odpowiedniej tablicy. Współrzędne bieżące (XB,YB) są aktualizowane współrzędnymi punktu końcowego łamanej.
Do określenia łamanej składającej się z n odcinków należy określić w tablicy (n+1) wierzchołków, czyli zadać (2n+2) współrzędne.
- Generacja zadanego tablicą ciągu odcinków /łamanej/ określonych względnie za pomocą $\Delta X_i, \Delta Y_i$ /GTABLW/.
W zbiorze pośrednim są zapisywane współrzędne wierzchołków łamanej. Początek łamanej znajduje się w punkcie bieżącym (XB,YB), a pozostałe wierzchołki w punktach o bezwzględnych współrzędnych:

$$X_j = XB + \sum_{i=1}^j TAB(1,i)$$

$$Y_j = YB + \sum_{i=1}^j TAB(2,i)$$

gdzie: $1 \leq j \leq N$, N jest liczbą odcinków łamanej, a TAB - tablica o rozmiarze $[2,N]$, zawierająca pary przyrostów między współrzędnymi kolejnych punktów łamanej. Współrzędne bieżące (XB,YB) mają wartość ostatniego punktu łamanej.

Generacja krzywych drugiego stopnia

Z podprogramami generującymi krzywe drugiego stopnia związane są podprogramy określające kolor i typ linii oraz widoczność i migotanie. Okręgi i łuki okręgów są przechowywane w zbiorze pośrednim w postaci umożliwiającej korzystanie z interpolatorów kołowych w tych urządzeniach graficznych, które mają tę możliwość. Pozostałe krzywe drugiego stopnia zapisane są w zbiorze pośrednim jako linia łamana /w postaci ciągu wierzchołków łamanej (X_i,Y_i) /.

- Generacja okręgu określonego bezwzględnie za pomocą współrzędnych środka (XS,YS) i promienia R /GOKROB/.
Podprogram zapisuje w zbiorze pośrednim współrzędne (XS,YS) środka okręgu i jego promień R, aktualizując współrzędne bieżące (XB,YB) współrzędnymi środka okręgu.
- Generacja łuku okręgu określonego bezwzględnie za pomocą współrzędnych środka (XS,YS), początku łuku (XP,YP) i kąta środkowego łuku α /GOKRBL/.
W zbiorze pośrednim są zapisywane współrzędne środka okręgu (XS,YS), współrzędne początku łuku (XP,YP) i współrzędne końca łuku (XK,YK), które oblicza się na podstawie zadanego kąta środkowego α , współrzędnych środka okręgu i współrzędnych początku łuku. Współrzędne bieżące (XB,YB) są aktualizowane współrzędnymi końca łuku (XK,YK).
- Generacja elipsy /GELIPS/.
Elipsę zadaje się za pomocą współrzędnych jej środka (XS,YS) długości małej i dużej półosi A i B i kąta nachylenia dużej półosi α . W zbiorze pośrednim podprogram zapisuje współrzędne punktów końcowych ciągu odcinków aproksymujących elipsę. Współrzędne bieżące (XB,YB) są aktualizowane współrzędnymi (XS,YS) środka elipsy.
- Generacja łuku elipsy /GELIP/.
Łuk elipsy zadaje się za pomocą współrzędnych środka elipsy (XS,YS), długości małej i dużej półosi A i B, kąta nachylenia dużej półosi α , współrzędnych punktu początkowego łuku elipsy (XP,YP) oraz kąta rozwarcia łuku β . W zbiorze pośrednim są zapisywane współrzędne punktów koń-

cowych ciągu odcinków aproksymujących łuk elipsy. Współrzędne bieżące (XB, YB) są aktualizowane współrzędnymi (XS, YS) środka elipsy.

- Generacja segmentu krzywej /funkcji uwikłanej/ drugiego stopnia /GFUDST/.

Segment krzywej drugiego stopnia zadaje się za pomocą współczynników równania uwikłanego opisującego krzywą 2-stopnia mającego postać $F(x, y) = ax^2 + bxy + cy^2 + dx + ey + f = 0$, punktów: początkowego (XP, YP) i końcowego (XK, YK) segmentu oraz kierunku obiegu krzywej /określonego za pomocą zmiennej logicznej B przyjmującej wartości TRUE i FALSE/. Podprogram zapisuje w zbiorze pośrednim współrzędne punktów końcowych ciągu odcinków /łamanej/ aproksymujących daną krzywą. Współrzędne bieżące (XB, YB) są aktualizowane współrzędnymi końca krzywej.

Generacja krzywych wyższych stopni

Podprogramy generujące krzywe wyższych stopni wiążą się z podprogramami określającymi kolor i typ linii oraz widoczność i migotanie. Krzywe zapamiętywane są w zbiorze pośrednim jako łamana.

- Generacja krzywej algebraicznej /funkcji uwikłanej/ N-tego stopnia /GFUNST/.

Segment krzywej n-tego stopnia zadaje się podobnie jak segment krzywej 2-stopnia za pomocą współczynników równania uwikłanego mającego postać:

$$F(x, y) = \sum_{k=0}^n \sum_{l=0}^{n-k} a_{n-k-l, 1} \cdot x^{n-k-l} \cdot y^l = 0$$

punktów: początkowego (XP, YP) i końcowego (XK, YK) segmentu, kierunku obiegu krzywej określonego przez zmienną logiczną B oraz parametru M, określającego wymiary tablicy współczynników a_{ij} , $M=n+1$ [2]. Podprogram zapisuje w zbiorze pośrednim współrzędne wierzchołków łamanej aproksymującej krzywą. Współrzędne (XB, YB) są aktualizowane współrzędnymi końca krzywej.

- Generacja krzywej średniokwadratowej /GTAKS/.

Krzywą średniokwadratową zadaje się za pomocą ciągu punktów umieszczonych w tablicy, będącej parametrem podprogramu. Punkty te są w podprogramie wykorzystywane do wyznaczenia współczynników wielomianu 4-stopnia zmiennej X. Zastosowany następnie podprogram GFUNST generuje łamaną aproksymującą krzywą wielomianową. Ta łamana jest zapisywana do zbioru pośredniego. Współrzędne bieżące (XB, YB) przyjmują wartości współrzędnych ostatniego punktu łamanej.

- Generacja krzywej gładkiej przechodzącej przez zadane punkty /GTAKG/.

Krzywą gładką zadaje się ciągiem punktów umieszczonych w tablicy będącej parametrem podprogramu. Podprogram przeprowadza przez zadane punkty krzywą gładką /mającą ciągłą drugą pochodną/ oraz wyznacza dodatkowe punkty na krzywej gładkiej. Liczba tych dodatkowych punktów również jest parametrem podprogramu. Punkty dodatkowe łącznie z punktami pierwotnymi tworzą łamaną aproksymującą krzywą gładką, która to łamana zostaje zapisana do zbioru pośredniego.

Współrzędne bieżące (XB, YB) są aktualizowane współrzędnymi ostatniego punktu krzywej.

Generacja pól

Z podprogramami generującymi pola wiążą się podprogramy określające kolor, widoczność i migotanie. Pola są wypełniane liniami równoległymi do osi X i w tej postaci są zapisywane w zbiorze pośrednim.

Typ pola określony jest przez typ ograniczającego go konturu i dlatego generujący to pole podprogram używa takich samych parametrów jak odpowiedni podprogram generacji konturu /krzywej/.

A więc podprogram generacji pola koła wymaga, jako parametrów, współrzędnych środka (XS, YS) i jego promienia R, natomiast podprogram generacji odcinka koła - parametrów identycznych jak podprogram generacji łuku okręgu, itp.

- Generacja pola koła /GPOLKO/.

Podprogram zapisuje w zbiorze pośrednim współrzędne punktów końcowych ciągu odcinków równoległych do osi X i tworzących obszar kołowy o środku w punkcie (XS, YS) i o promieniu R.

Współrzędne bieżące (XB, YB) są aktualizowane wartościami współrzędnych środka okręgu (XS, YS).

- Generacja pól: odcinka koła /GPCLCK/ oraz wycinka koła /GPCLWK/.
Parametry odcinka i wycinka koła są identyczne jak parametry łuku okręgu. Podprogram zapisuje w zbiorze pośrednim współrzędne punktów końcowych ciągu równoległych do osi X odcinków tworzących obszary odcinka i wycinka koła o środku w punkcie (XS,YS) i promieniu R. Współrzędne bieżące (XB,YB) przyjmują wartości współrzędnych punktu początkowego łuku (XP,YP) .
- Generacja pola elipsy /GPOLEL/.
W zbiorze pośrednim są zapisywane współrzędne punktów końcowych ciągu /równoległych do osi X/ odcinków, tworzących obszar eliptyczny o środku w punkcie (XS,YS) .
Współrzędne bieżące (XB,YB) są aktualizowane współrzędnymi środka elipsy (XS,YS) .
- Generacja pola odcinka elipsy /GPLOE/.
Parametry odcinka elipsy są takie same jak parametry podprogramu generującego łuk elipsy. Podprogram zapisuje w zbiorze pośrednim współrzędne punktów końcowych ciągu odcinków, równoległych do osi X i tworzących obszary odcinka elipsy o środku w punkcie (XS,YS) . Współrzędne bieżące (XB,YB) przyjmują wartości współrzędnych punktu początkowego łuku elipsy (XP,YP) .
- Generacja pola ograniczonego konturem krzywej drugiego stopnia /GPCLKD/.
Podprogram zapisuje w zbiorze pośrednim współrzędne punktów końcowych ciągu /równoległych do osi X/ odcinków, tworzących pole obszaru ograniczonego krzywą 2-stopnia i prostą łączącą punkt początkowy i końcowy krzywej. Współrzędne bieżące (XB,YB) są aktualizowane współrzędnymi końca krzywej.
- Generacja pola trójkąta /GPCLT/.
Trójkąt zadawany jest za pomocą współrzędnych jego wierzchołków. Podprogram zapisuje w zbiorze pośrednim ciąg równoległych do osi X odcinków zakreślających pole trójkąta. Współrzędne bieżące (XB,YB) są aktualizowane współrzędnymi trzeciego wierzchołka trójkąta.
- Generacja pola prostokąta zadanego w tablicy za pomocą dwóch przeciwległych wierzchołków /GPCLPR/.
Pole prostokąta może być przez niektóre typy monitorów graficznych wypełnione wskazanym /za pomocą parametru/ rodzajem wzoru. Dlatego podprogram zapisuje w zbiorze pośrednim jedynie ciąg wierzchołków prostokąta, a nie, jak w wypadku innych pól, ciąg linii równoległych do osi X wypełniających pole figury. Po wykonaniu podprogramu punkt bieżący znajduje się w ostatnim wierzchołku prostokąta.

Generacja siatki prostokątnej, osi układu współrzędnych oraz wykresów /liniowego i słupkowego/

Z podprogramami generacji siatki prostokątnej i osi układu współrzędnych wiążą się podprogramy określające kolor, migotanie i widoczność. Natomiast z podprogramami generującymi wykresy wiążą się dodatkowo podprogramy określające wypełnienie wykresu słupkowego oraz łączenie punktów wykresu liniowego, a także podprogram określający bazę dla wykresu słupkowego.

- Generacja siatki prostokątnej /GSIATP/.
Siatkę zadaje się za pomocą współrzędnych przeciwległych rogów /lewego dolnego i prawego górnego/ oraz liczby pól siatki wzdłuż osi X i Y. Podprogram zapisuje w zbiorze pośrednim ciąg odcinków tworzących siatkę prostokątną, równoległą do brzegów ekranu i ograniczoną współrzędnymi odpowiednich punktów i liczbą pól siatki. Współrzędne bieżące przyjmują wartości współrzędnych prawego górnego rogu siatki.
- Generacja osi liniowego układu współrzędnych /GUKWSP/.
Oś wraz z opisem zadaje się za pomocą następujących parametrów:
 - wyboru osi /X bądź Y/,
 - współrzędnych początku osi,
 - wartości początku osi,
 - długości przedziału na osi,
 - liczby przedziałów,
 - przyrostu wartości w przedziale,
 - długości tekstu opisu osi,
 - tekstu opisu osi.Podprogram zapisuje w zbiorze pośrednim ciąg odcinków tworzących oś układu współrzędnych prostokątnych z podziałką liniową i umieszcza tekst opisu osi. Współrzędne bieżące (XB,YB) przyjmują wartości współrzędnych początku osi.

- Generacja osi logarytmicznego układu współrzędnych /GLUKB3/.
Oś logarytmicznego układu współrzędnych zadaje się za pomocą takich samych parametrów jak oś liniowego układu współrzędnych. W zbiorze pośrednim jest zapisywany ciąg odcinków tworzących oś układu współrzędnych prostokątnych w skali logarytmicznej oraz jest umieszczony tekst opisu osi. Współrzędne bieżące (XB, YB) przyjmują wartości współrzędnych początku osi.
- Generacja wykresu liniowego /GWYLI.N/ i wykresu słupkowego /GWYSLP/.
Obydwa rodzaje wykresów: liniowy i słupkowy zadaje się za pomocą następujących parametrów:
 - wyboru osi ze stałym krokiem,
 - współrzędnej początkowej na osi ze stałym krokiem,
 - wartości kroku,
 - liczby punktów wykresu,
 - tablicy współrzędnych punktów wykresu,
 /bazę dla obydwu rodzajów wykresów zadaje się za pomocą podprogramu GRAZAW/.
Podprogramy zapisują do zbioru pośredniego współrzędne początkowe na osi z ustalonym krokiem, wartość kroku oraz ciąg wartości zmieniających się współrzędnych. Współrzędne bieżące (XB, YB) przyjmują wartości współrzędnych początku osi.

Generacja pola komórkowego i ciągu komórkowego

- Generacja obrazu komórkowego /GOBKON/.
Zadaje się go za pomocą współrzędnych jego lewego dolnego rogu, wymiarami i zawartością tablicy zawierającej własności /attributy/ oddzielnych greli /komórek/ obrazu. Podprogram zapisuje w zbiorze pośrednim wartości współrzędnych początkowych obrazu, wymiar tablicy oraz ciąg zadanych tablicą własności /cechy/ kolejnych greli obrazu. Współrzędne bieżące (XB, YB) przyjmują wartości współrzędnych prawego górnego rogu pola.
- Generacja wektora komórkowego równoległego do osi X /GMKKON/.
Jest on zadawany za pomocą współrzędnych początku, wymiaru tablicy i zawartością tablicy zawierającej własności /cechy/ kolejnych greli wektora. Do zbioru pośredniego wpisywane są wartości współrzędnych początku wektora oraz ciąg danych z tablicy, będących cechami poszczególnych greli wektora.

Generacja znaków alfanumerycznych i graficznych /specjalnych/

- Generowanie tekstu alfanumerycznego /GTEKST/.
Podprogram zapisuje w zbiorze pośrednim tekst umieszczony w odpowiedniej tablicy /będącej parametrem podprogramu/, przeznaczony do wyświetlenia na monitorze /alfanumerycznie/. Współrzędne bieżące (XB, YB) przyjmują wartości prawego dolnego rogu pola ostatniego znaku. Atrybuty związane z tekstem alfanumerycznym są przekazywane w podprogramach określających kolor, tło, skalę i kierunek pisania znaków oraz migotanie.
- Generacja znaków graficznych /GZNAKG/.
Znaki graficzne zadaje się za pomocą następujących parametrów:

- współrzędnych początku linii pisma,	- szerokości i wysokości znaków,
- tablicy zawierającej kody znaków,	- kąta nachylenia pisma.
- rozmiarów tej tablicy i liczby znaków,	

 W zbiorze pośrednim zapisuje się ciąg odcinków tworzących kontury znaków alfanumerycznych i graficznych. Współrzędne bieżące (XB, YB) przyjmują wartości współrzędnych prawego dolnego rogu pola ostatniego znaku.
Z tym podprogramem wiążą się podprogramy określające kolor, widoczność i migotanie.

Kreślenie atrybutów informacji graficznej

Podprogramy określające właściwości /attributy/ muszą być wywołane przed podprogramem generującym pierwotnik graficzny, z którym te atrybuty mają być związane. Atrybut jest zmieniany tylko

1) : określenie jego innej wartości, co realizują oddzielne podprogramy.

Istnieje też podprogram /GSTANB/ ustalający wartości wszystkich atrybutów, a także podprogram przywracający tzw. stan normalny wszystkich atrybutów /GSTANØ/.

● Złazanie tablicy kolorów /GTAKO/.

Podprogram pobiera ze wskazanej przez parametr tablicy 1 zapisuje w zbiorze pośrednim 16 liczb naturalnych, określających wybrany zestaw barwny na miejsce dotychczasowego. Każdy kolor powstaje z połączenia trzech podstawowych kolorów: czerwonego, zielonego i niebieskiego branych w różnych proporcjach. Może być 4096 różnych zestawów.

Standardowy zestaw kolorów jest następujący:

1 - biały	6 - karminowy	11 - fioletowy	16 - beżowy
2 - cytrynowy	7 - oliwkowy	12 - brązowy	
3 - żółty	8 - zielony	13 - czarny	
4 - pomarańczowy	9 - błękit	14 - szary	
5 - czerwony	10 - niebieski	15 - różowy	

● Wybranie koloru z tablicy /GKOLOR/.

Podprogram zapisuje w zbiorze pośrednim wybrany /za pomocą parametru/ kolor z tablicy barw.

● Wybór typu linii /GTYPLN/.

Podprogram wpisuje do zbioru pośredniego wybrany /za pomocą parametru/ kod jednego z czterech możliwych typów linii.

● Wybór typu znacznika /śladu/ /GTYPSL/.

Podprogram zapisuje w zbiorze pośrednim wybrany /za pomocą parametru/ kod jednego z pięciu możliwych typów znaczników.

● Określenie skali znaku /GSKAZN/.

Podprogram zapisuje w zbiorze pośrednim parametry określające skalę szerokości i wysokości pola znaku.

● Określenie kierunku ścieżki znaków /GKCIĄG/.

Podprogram umieszcza w zbiorze pośrednim liczbę rzeczywistą określającą kierunek między ścieżką znaków a osią OX.

● Określenie pochylenia znaku /GKZNAK/

Podprogram wpisuje kąt pochylenia linii pionowej znaku względem ścieżki znakowej.

● Wybór barwy tła znaku /GTLOZN/.

Podprogram zapisuje w zbiorze pośrednim swój parametr, określający wybraną w tablicy kolorów barwę tła znaku.

● Włączenie migotania /GMIGOT/.

Podprogram umieszcza w zbiorze pośrednim swój parametr - informację, dotyczącą migotania obrazu.

● Określenie za pomocą parametru podprogramu wykończenia wykresu słupkowego /zakreskowanie/ lub łączenia wykresu liniowego /GTYPSL/.

● Generacja bazy dla wykresu /GBAZAW/.

Podprogram zapisuje do zbioru pośredniego parametr, będący wartością bazy dla wykresu słupkowego.

● Określenie strony pamięci /GSTRPA/

Podprogram wpisuje liczbę całkowitą określającą stronę /bufor/ pamięci.

● Złazanie stanu początkowego wszystkich atrybutów /GSTANØ/.

Podprogram wpisuje do zbioru pośredniego swoje parametry zadające tzw. stan normalny wszystkich atrybutów.

● Złazanie bieżącej wartości wszystkich atrybutów /GSTANB/.

Podprogram zapisuje w zbiorze pośrednim wybrane przez użytkownika atrybuty informacji graficznej.

POSTPROCESORY SYSTEMU PSG

Postprocesory są to podprogramy, które powodują wyświetlanie lub wykreślanie na wyjściowych urządzeniach graficznych, zakodowanych obrazów, pamiętanych w systemowym archiwum, tzw. zbiorze pośrednim danych graficznych - ZPDG. Kody tych obrazów - dane wejściowe dla postprocesorów - są pamiętane w postaci sprzętowo niezależnych tzw. pierwotników zbioru pośredniego PZP, generowanych i umieszczonych w ZPDG za pomocą podprogramów generacyjnych.

Postprocesory przetwarzają zawartość zbioru ZPDG na postać odpowiednią dla danego typu urządzenia obrazującego.

W systemie PSG zrealizowanym na minikomputerze SH4 istnieją trzy postprocesory wyświetlające obrazy na monitorze rastrowym MERA 7954: PIR, PIR1, PIR2.

- PIR (LUN, 'NAZWAZ') - podprogram wyświetla na ekranie monitora MERA 7954 /przyłączonego w systemie DOS RW do numeru logicznego LUN/ zbiór ZPDG o zadanej nazwie NAZWAZ. Nazwę zbioru NAZWAZ specyfikuje się zgodnie z zasadami specyfikacji plików w systemie operacyjnym DOS RW. NAZWAZ może być stałą tekstową lub tablicą logiczną.
- PIR1 LUN, ('NAZWAZ', 'NAZWAO') - podprogram wyświetla na ekranie monitora MERA 7954 obiekt o nazwie NAZWAO, ze zbioru ZPDG NAZWAZ. NAZWAO może być stałą tekstową lub tablicą logiczną. Parametry LUN i NAZWAZ mają takie samo znaczenie, jak w postprocesorze PIR. Użycie w tym podprogramie, zamiast parametru NAZWAO, słowa kluczowego "TAK", powoduje wyświetlenie na ekranie monitora wszystkich obiektów, znajdujących się w zbiorze pośrednim ZPDG.
- PIR2 (LUN, 'NAZWAZ', NROP, NROO) - podprogram wyświetla na ekranie monitora MERA 7954 ciąg obiektów /poczynając od obiektu o numerze NROP do obiektu o numerze NROO/ ze zbioru ZPDG o nazwie NAZWAZ.

PODPROGRAMY DEFINIUJĄCE ELEMENTY GEOMETRYCZNE OBRAZU

Ta grupa podprogramów definiuje różnymi sposobami proste elementy geometryczne, najczęściej w programach graficznych występujące tj. punkty, odcinki prostych i krzywe drugiego stopnia oraz przedstawia je w standardowej postaci, występującej w podprogramach generacyjnych.

W grupie podprogramów definiujących wyróżnia się: 6 podprogramów definiujących punkt, 5 podprogramów definiujących odcinek prostej, 4 podprogramy definiujące okrąg, 2 podprogramy definiujące łuk okręgu oraz podprogram definiujący krzywą 2-stopnia, przechodzącą przez 5 punktów.

Podprogramy definiujące zapewniają odpowiednią diagnostykę błędów przez wysyłanie właściwych komunikatów na urządzenia zewnętrzne.

Kąty występujące jako parametry w omawianych procedurach są wyrażane w mierze stopniowej /w stopniach i częściach stopni/.

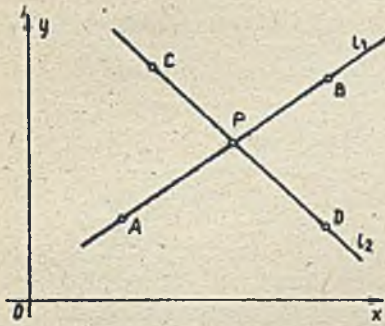
Definiowanie punktu

Dowolny punkt na płaszczyźnie może być zdefiniowany jako punkt przecięcia następujących elementów geometrycznych: dwóch prostych, okręgu z prostą, dwóch okręgów i krzywej drugiego stopnia z prostą, ponadto jako punkt leżący na prostej i odległy od jej początku o wielkość D oraz jako punkt położony na okręgu, wyznaczony przez jego promień i kąt α zawarty między osią OX i promieniem.

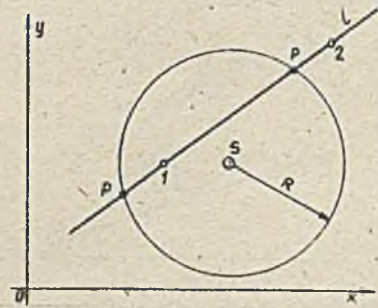
Każdy podprogram wyznacza współrzędne prostokątne punktu w postaci standardowej (X,Y).

System PSG zapewnia następujące sposoby definiowania punktów.

- Definiowanie punktu za pomocą dwóch przecinających się prostych /wektorów/, podprogram o nazwie DPKTWA /rys. 6/. Proste przecinające się są określone odpowiednio przez pary punktów o znanych współrzędnych prostokątnych (X_1, Y_1) .

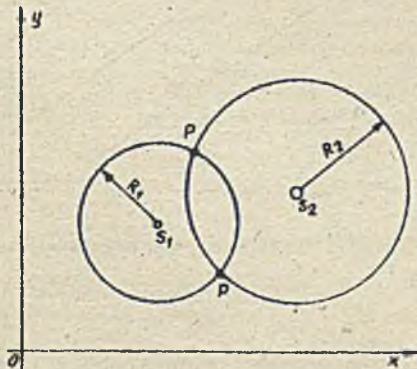


Rys. 6. Punkt przecięcia dwóch prostych

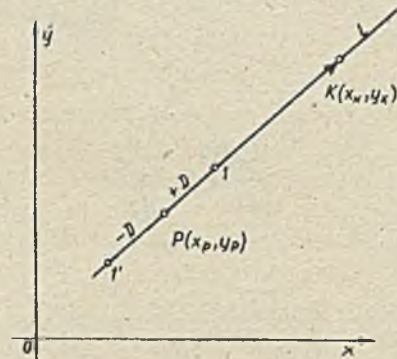


Rys. 7. Punkt przecięcia prostej z okręgiem

- Definiowanie punktu za pomocą okręgu przecinającego się z prostą /wektorem/, podprogram o nazwie DPKTOW /rys. 7 /. Prosta jest określona współrzędnymi prostokątnymi dwóch jej punktów, a okrąg współrzędnymi prostokątnymi środka i wartością promienia. O wyborze jednego punktu przecięcia, z dwóch możliwych, decyduje odpowiedni parametr A, który zadaje użytkownik.
- Definiowanie punktu za pomocą dwóch przecinających się okręgów, podprogram o nazwie DPKTDO /rys. 8/. Okręgi są określone współrzędnymi prostokątnymi środków i wartościami ich promieni. O wyborze jednego punktu przecięcia, z dwóch możliwych, decyduje odpowiedni parametr A, który zadaje użytkownik.
- Definiowanie punktu leżącego na prostej /wektorze/ i odległego od jej punktu początkowego o odległość D, podprogram o nazwie DPTWPD /rys. 9/.



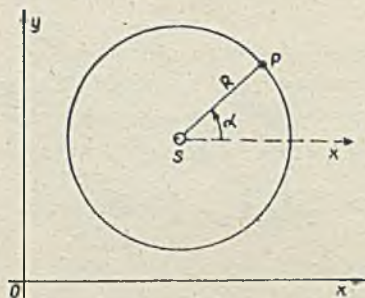
Rys. 8. Punkt przecięcia dwóch okręgów



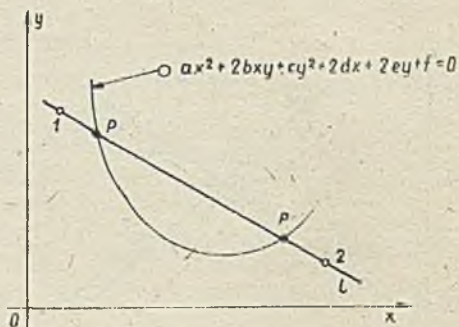
Rys. 9. Definiowanie punktu na prostej

Prosta jest zadana współrzędnymi prostokątnymi dwóch jej punktów. Szukany punkt może znajdować się między danymi punktami prostej lub poza tymi punktami, na jej przedłużeniu. Przyjmuje się, że odległość D jest dodatnia w kierunku wprost, a ujemna w kierunku wstecz od punktu początkowego.

- Definiowanie punktu położonego na okręgu i wyznaczonego przez promień, tworzący kąt α z osią OX, podprogram o nazwie DPKTK /rys. 10/. Okrąg jest zadany współrzędnymi jego środka i wartością promienia R. Dany kąt $0 \leq \alpha \leq 360^\circ$ jest położony na lewo od dodatniego kierunku osi OX.
- Definiowanie punktu za pomocą krzywej drugiego stopnia /stożkowej/ i przecinającej ją prostej /wektora/, podprogram o nazwie DPKTW /rys. 11/. Prosta jest określona współrzędnymi prostokątnymi dwóch jej punktów, natomiast krzywa stożkowa parametrami jej równania ogólnego $ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0$. O wyborze jednego punktu przecięcia, z dwóch możliwych, decyduje odpowiedni parametr A, który zadaje użytkownik.



Rys. 10. Definiowanie punktu na okręgu



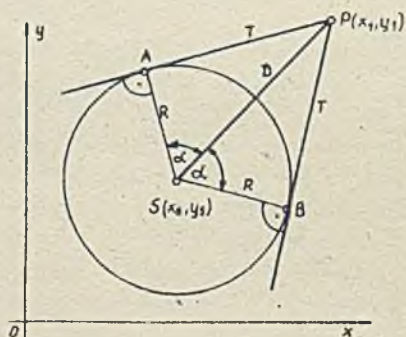
Rys. 11. Punkt przecięcia prostej z krzywą drugiego stopnia

Definiowanie prostej

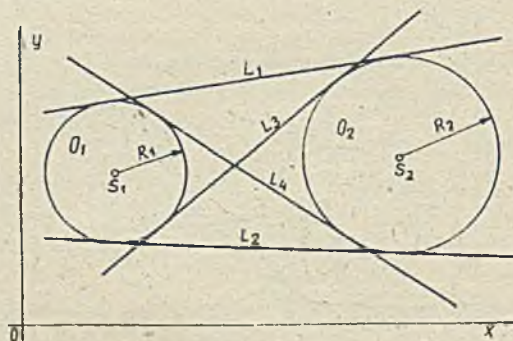
Prostą na płaszczyźnie można definiować jako styczną do okręgu, jako styczną do dwóch okręgów, jako prostą nachyloną do osi OX pod zadaniem kątem, jako prostą nachyloną do danej prostej pod zadaniem kątem i jako prostą równoległą do danej prostej. W każdym z podprogramów definiowana prosta jest wyznaczona w postaci standardowej przez współrzędne prostokątne dwóch punktów, zapisane w odpowiedniej tabeli.

System PSG umożliwia następujące sposoby definiowania odcinków prostych.

- Definiowanie prostej /wektora/ przechodzącej przez punkt i stycznej do okręgu, podprogram o nazwie DWEKPO /rys. 12/.



Rys. 12. Definiowanie prostej stycznej do okręgu i przechodzącej przez punkt poza okręgiem

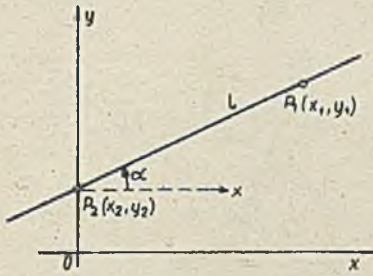


Rys. 13. Prosta styczna do dwóch okręgów

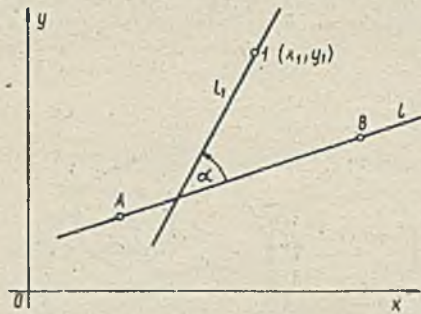
Okrąg jest zadany za pomocą współrzędnych prostokątnych jego środka i wartości promienia.

Współrzędne danego punktu, który leży poza okręgiem, są jednocześnie współrzędnymi drugiego punktu definiowanej prostej. O wyborze jednego z dwóch punktów styczności decyduje wartość odpowiedniego parametru IP, którą zadaje użytkownik.

- Definiowanie prostej /wektora/ stycznej do dwóch okręgów, podprogram o nazwie DWEKDO /rys. 13/. Okręgi są zadawane współrzędnymi ich środków i wartościami promieni. Mogą być cztery możliwe położenia prostej względem okręgów. O wyborze jednej z tych możliwości decydują wartości odpowiednich parametrów A i B, które zadaje użytkownik.
- Definiowanie prostej /wektora/ przechodzącej przez punkt, o znanych współrzędnych prostokątnych i tworzącej kąt α z osią OX, podprogram o nazwie DWEKPX /rys. 14/. Wartość kąta α jest liczona w lewo od dodatniego kierunku osi OX do kierunku prostej.
- Definiowanie prostej /wektora/ przechodzącej przez punkt o znanych współrzędnych prostokątnych i tworzącej kąt α z daną prostą /wektorem/, podprogram o nazwie DWEKPA /rys. 15/.



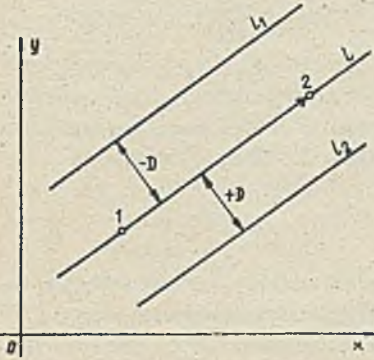
Rys. 14. Definiowanie prostej przechodzącej przez punkt i nachylonej do osi OX



Rys. 15. Definiowanie prostej przechodzącej przez punkt pod danym kątem w stosunku do danej prostej

Zadana prosta jest określona przez współrzędne dwóch jej punktów.

- Definiowanie prostej /wektora/ równoległej do danej prostej i odległej od niej o odległość D, podprogram o nazwie DWEKRD /rys. 16/.



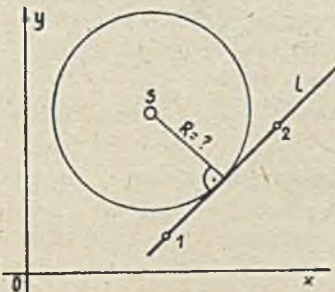
Rys. 16. Prosta równoległa

Prosta zadana jest określona współrzędnymi prostokątnymi dwóch jej punktów. Z dwóch możliwych prostych równoległych, położonych w odległości D, wybiera się właściwą, nadając wartości D odpowiedni znak.

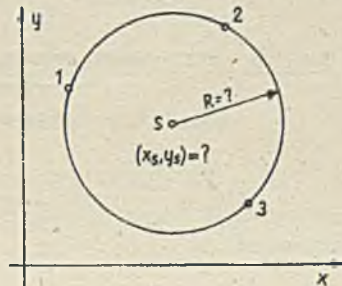
Definiowanie okręgu koła

Każdy z podprogramów definiujących okrąg wyznacza standardowe parametry geometryczne okręgu, to jest współrzędne środka okręgu i jego promień. Okrąg na płaszczyźnie może być zdefiniowany czterema różnymi sposobami.

- Definiowanie okręgu o danych współrzędnych prostokątnych jego środka i stycznej do prostej /wektora/, podprogram o nazwie DCKRSW /rys. 17/. Prosta, styczna do okręgu, jest zadana współrzędnymi jej dwóch punktów.
- Definiowanie okręgu przechodzącego przez trzy punkty o znanych współrzędnych prostokątnych, podprogram o nazwie DCKRTP /rys. 18/.

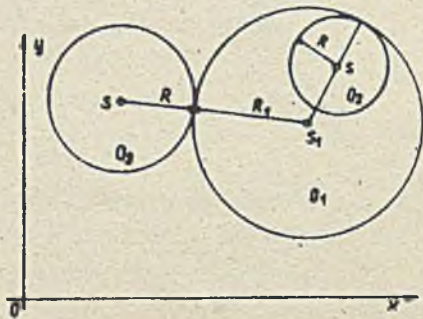


Rys. 17. Okrąg styczny do prostej

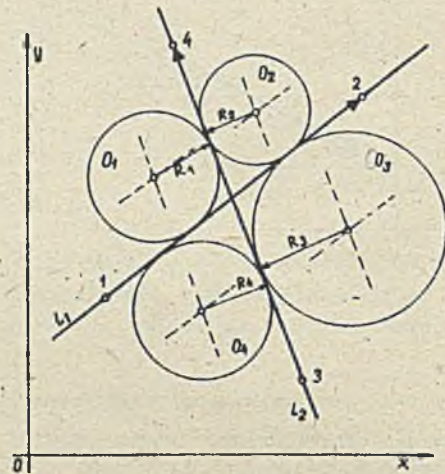


Rys. 18. Okrąg przechodzący przez trzy punkty

- Definiowanie okręgu o danych współrzędnych prostokątnych jego środka i stycznego do drugiego okręgu, podprogram o nazwie DOKRSO /rys. 19/. Drugi okrąg jest określony współrzędными jego środka i wartością promienia. Okrąg definiowany i dany mogą być wzajemnie styczne zewnętrznie i wewnętrznie. O wyborze odpowiedniego okręgu decyduje wartość parametru λ , którą zadaje użytkownik.



Rys. 19. Definiowanie okręgów stycznych



Rys. 20. Okrąg styczny do dwóch prostych

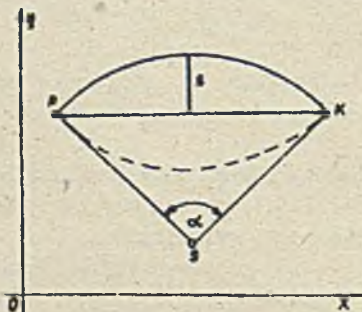
- Definiowanie okręgu o danym promieniu i stycznego do dwóch prostych /wektorów/, podprogram o nazwie DOKRDW /rys. 20/. Dane proste są określone współrzędными prostokątnymi dwóch odpowiednich punktów każda. Występują tu cztery możliwości położenia okręgu względem dwóch prostych. O wyborze jednego z czterech okręgów decyduje znak i wartość promienia R okręgu.

Definiowanie łuku okręgu

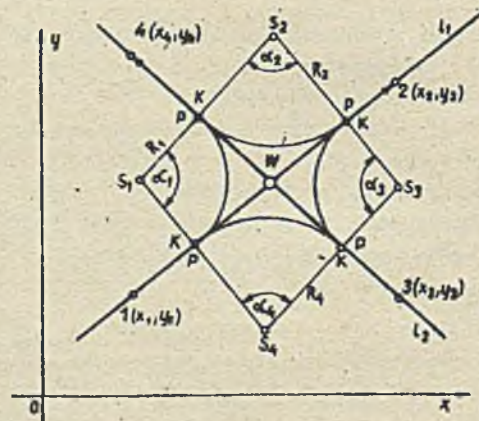
W FSG standardową postać łuku okręgu wyznacza się za pomocą następujących parametrów: współrzędnych prostokątnych środka okręgu, kąta środkowego opartego na tym łuku i promienia okręgu.

System zapewnia 2 następujące sposoby definiowania łuku okręgu.

- Definiowanie łuku okręgu zadanego współrzędными punktów początkowego i końcowego oraz strzałką łuku, podprogram o nazwie DLKRS /rys. 21/. Istnieją tu dwa możliwe położenia środka okręgu. O wyborze jednego z dwóch możliwych położen środka okręgu decyduje wartość i znak strzałki łuku.



Rys. 21. Definiowanie łuku okręgu za pomocą strzałki i współrzędnych punktów końcowych



Rys. 22. Łuk kołowy styczny do dwóch prostych

Definiowanie łuku okręgu stycznego do dwóch prostych i zawartego między nimi /naroże/, podprogram o nazwie DLCKRN /rys. 22/. Proste są zadane współrzędnymi dwóch punktów, położonych odpowiednio na każdej prostej. Natomiast łuk ma znany promień R. O wyborze jednego z czterech łuków kołowych /naroży/ decyduje znak i wartość promienia łuku R.

Definiowanie krzywej 2-stopnia

Podprogram DFUDST umożliwia definiowanie krzywej /funkcji uwikłanej/ drugiego stopnia, o równaniu ogólnym

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0$$

i przechodzącej przez 5 punktów o znanych współrzędnych prostokątnych.

W wyniku realizacji podprogramu otrzymuje się wartości współczynników a, b, c, d, e, f równania ogólnego krzywej, zapisane w standardowej /dla krzywych 2-stopnia/ tablicy, która zawiera /oprócz współczynników a, b, c, d, e, f/ również punkty: początkowy (XP, YP) i końcowy (XK, YK) segmentu krzywej, /którymi są pierwszy i piąty z punktów zadających linię/ oraz kierunek obiegu krzywej wyznaczony przez kolejność występowania punktów zadających krzywą /przy założeniu, że w podprogramie definiującym wymienione zostały w tej kolejności, w jakiej znajdują się na krzywej/.

PODPROGRAMY TRANSFORMACJI

Podprogramy transformacji umożliwiają dokonywanie na standardowych postaciach elementów pierwotnych, trzech podstawowych przekształceń: obrotu o dowolny kąt względem dowolnego punktu płaszczyzny, skalowania przez dowolny współczynnik oraz przesunięcia o dowolny skończony wektor.

Podane niżej parametry formalne tych podprogramów mają stałe znaczenie. Są to parametry transformacji:

(XO, YO) - współrzędne środka obrotu,

ALFA - kąt obrotu w prawoskrętnym układzie współrzędnych /liczony na lewo od kierunku obrotu - środek obrotu i punkt obracany/, w stopniach i jego częściach,

S - współczynnik skali,

(DX, DY) - wektor translacji /przesunięcia/,

TABTR - tablica o rozmiarze /6/, zawierająca parametry transformacji w podanej wyżej kolejności.

Parametry transformacji mogą występować bezpośrednio w podprogramie jako parametry formalne, wówczas drugą literą w nazwie tego podprogramu jest B. Mogą one być przechowywane w tablicy TABTR, która jest parametrem formalnym podprogramu i wówczas w nazwie podprogramu jako druga występuje litera T.

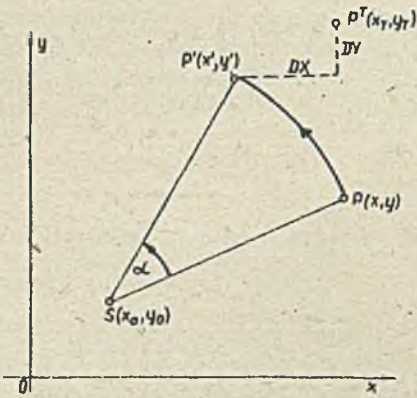
Transformacja punktu i wektora oraz ciągu punktów i wektorów

System PSG zapewnia transformacje punktów i wektorów oraz ich ciągów w sposób bezwzględny, czyli w układzie współrzędnych prostokątnych urządzenia graficznego i w sposób względny, czyli względem współrzędnych bieżących kursora graficznego bądź głowicy kreślaka. Występują tu następujące możliwości.

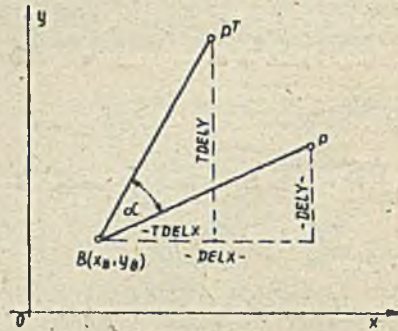
● Transformacja punktu określonego w sposób bezwzględny, podprogramy o nazwach TBKTB i TTKTB /rys. 23/. Podprogramy obliczają współrzędne (XT, YT) punktu P^T, dokonując, w tym samym układzie współrzędnych, następujących przekształceń:

- obrotu punktu P(x, y), o dany kąt α , względem dowolnego punktu płaszczyzny, zwanego środkiem obrotu S(x₀, y₀), o znanych współrzędnych, do położenia P'(x', y'),
- skalowania współrzędnych punktu P' przez dowolny zadany współczynnik zwiększający lub zmniejszający, oprócz zera,
- przesunięcia /translacji/ punktu P' o dowolny zadany wektor (DX, DY).

Podprogram TTKTB tym się różni od podprogramu TBKTB, że parametry transformacji są podawane w tablicy TABTR w kolejności: XO, YO, ALFA, S, DX, DY, a nie jako oddzielne parametry.

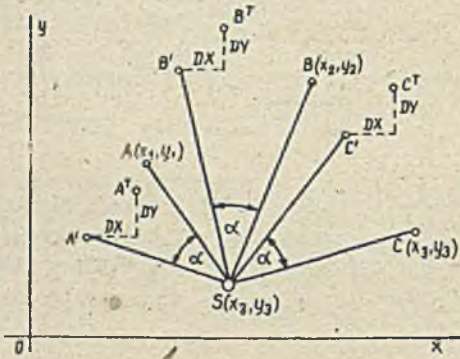


Rys. 23. Transformacja bezwzględna punktu

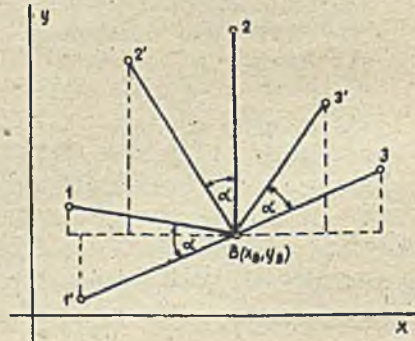


Rys. 24. Transformacja względna punktu

- 2/ Transformacja punktu lub wektora zadanego w sposób względny, podprogramy o nazwach TBFKTW i TTKTW /rys. 24/. Podprogramy dokonują transformacji przez obrót i skalowanie na podstawie danych przyrostów współrzędnych (DELX, DELY) oraz wartości parametrów transformacji ALFA i S.
- 3/ Transformacja zadanego tablicą ciągu punktów lub wektorów określonych w sposób bezwzględny, podprogramy o nazwach TBTPWB i TTPWB /rys. 25/. Podprogramy obliczają współrzędne szeregu punktów, wykonując w tym samym układzie współrzędnych trzy przekształcenia: obrót, skalowanie i przesunięcie. Danymi są współrzędne punktów do transformacji, zawarte w odpowiedniej tablicy, liczba punktów oraz parametry transformacji.

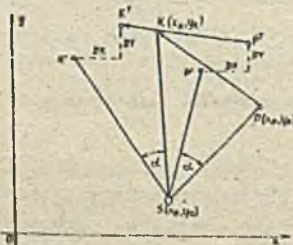


Rys. 25. Transformacja bezwzględna ciągu punktów



Rys. 26. Transformacja względna ciągu punktów

- 4/ Transformacja zadanego tablicą ciągu punktów lub wektorów określonych w sposób względny, podprogramy o nazwach TBTPWB i TTPWB /rys. 26/. Podprogramy obliczają przyrosty współrzędnych dowolnej liczby punktów względem współrzędnych bieżących XB, YB .. Jest to transformacja przez obrót i skalowanie. Danymi są przyrosty współrzędnych między współrzędnymi transformowanych punktów i współrzędnymi bieżącymi, liczba punktów transformowanych oraz parametry transformacji ALFA i S.
- 5/ Transformacja wektora zadanego bezwzględnie, podprogramy o nazwach TBWEKB i TWEKB /rys. 27/.



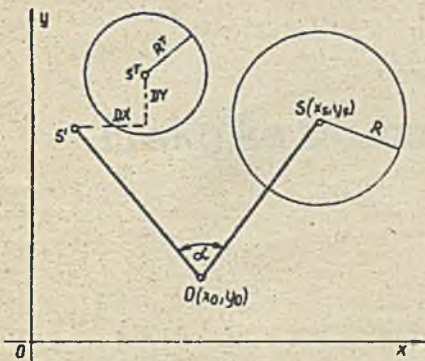
Rys. 27. Transformacja bezwzględna wektora

Podprogramy obliczają współrzędne punktów końcowych wektora, dokonując, w tym samym układzie przekształceń: obrotu, skalowania i przesunięcia. Transformacja ta jest realizowana na podstawie współrzędnych punktów początkowego i końcowego wektora oraz parametrów transformacji.

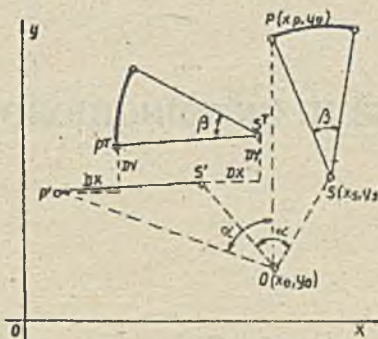
Transformacja okręgu oraz jego łuku

System PSG stwarza tu dwie możliwości.

- 1/ Transformacja okręgu określonego w sposób bezwzględny, podprogramy o nazwach TEOKR i TFOKR /rys. 28/. Podprogramy dokonują transformacji współrzędnych środka okręgu przez obrót, skalowanie i przesunięcie. Natomiast promień okręgu podlega tylko skalowaniu. Danymi są współrzędne środka okręgu i promień okręgu przed transformacją oraz parametry transformacji.



Rys. 28. Transformacja bezwzględna okręgu



Rys. 29. Transformacja łuku kołowego

- 2/ Transformacja łuku okręgu, podprogramy o nazwach TELOKR i TFOKR /rys. 29/. Podprogramy dokonują transformacji łuku okręgu za pomocą trzech przekształceń: obrotu, skalowania i przesunięcia. Praktycznie jest to transformacja wektora SP, czyli obliczenie współrzędnych punktów S^T i P^T przy zachowaniu tej samej wartości kąta środkowego β /rys. 29/. Danymi są współrzędne środka okręgu, współrzędne punktu początkowego łuku, skierowany kąt środkowy łuku β oraz parametry transformacji.

Transformacja segmentu krzywej 2-stopnia

Transformacji, zadanego tablicą segmentu krzywej 2-stopnia /stożkowej/, dokonują podprogramy o nazwach TEFUDS i TFOFUDS. Danymi są parametry równania ogólnego krzywej 2-stopnia, omówione wcześniej oraz parametry transformacji.

Transformacja segmentu krzywej n-stopnia

Transformacja krzywej n-stopnia /funkcji uwikłanej/ jest realizowana podprogramami TEFUNS i TFOUNS przez obrót, skalowanie i przesunięcie. Danymi są parametry krzywej, stopień krzywej i parametry transformacji.

Literatura

- [1] Bronowska M., Mokrzycki W.: Oprogramowanie rastrowego monitora graficznego w systemie MERA 400. Warszawa 1979 Archiwum Opracowań IMM
- [2] Mokrzycki W.: Dyskretyzacja krzywych algebraicznych na jednorodnych siatkach kwadratowych k-spójnych /algorytm II/, Prace Naukowo-Badawcze IMi 1982 z. 2 s. 3-40
- [3] Poradnik matematyczny. PWN Warszawa 1980
- [4] System PSG dla minikomputera SM4. Dokumentacja użytkowa. IMi Warszawa 1984

Ludmiła W. LIBIZOWA
Władimir P. MUCHANOW
Szkoła przy Ambasadzie ZSRR

Praktyka stosowania mikrokomputerów w szkole

Znajomość podstaw informatyki w dobie obecnej stała się poważnym elementem ogólnej kultury, dlatego też w Związku Radzieckim od paru lat do programów nauczania w klasach 9. i 10. szkół ogólnokształcących został wprowadzony nowy przedmiot - "Podstawy informatyki i techniki obliczeniowej". Do nauczania tego przedmiotu służy tzw. próbny podręcznik pod redakcją członków Akademii Nauk SRR - Jerszowa i Monachowa.

Pierwsza część tego podręcznika jest przeznaczona dla uczniów klas dziewiątych i określa zasady opracowania algorytmów dostosowanych do rozwiązywania zadań. Są więc sprecyzowane podstawowe zasady niezbędne dla sformułowania zadania w języku algorytmu.

Autorzy podręcznika zakładają "bezmaszynową" naukę informatyki, ponieważ w Związku Radzieckim wyposażenie szkół w mikrokomputery będzie odbywało się stopniowo, w miarę ich produkcji.

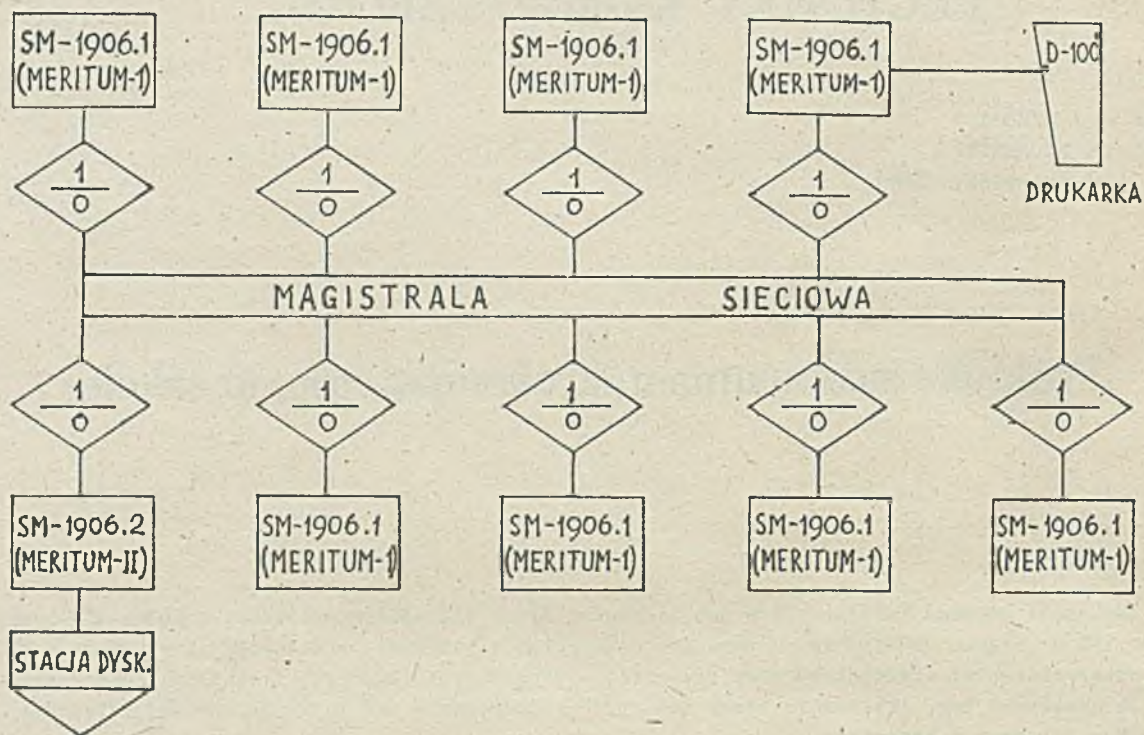
Druga część podręcznika jest przeznaczona dla uczniów klas dziesiątych. Obejmuje naukę elementów informatyki z użyciem (tam, gdzie jest to możliwe) komputera szkolnego. W podręczniku znajdują się podstawowe informacje o pracy komputerów i wstępne wiadomości o programowaniu. Dla opracowania programów w podręczniku proponuje się wykorzystanie jednego z języków algorytmicznych, w szczególności języka BASIC.

W szkole przy Ambasadzie Związku Radzieckiego w Warszawie konsekwentnie realizowane jest nauczanie podstaw informatyki, zgodnie z zakresem proponowanym w podręczniku Jerszowa w wersji z wykorzystaniem komputerów. Jest to możliwe ponieważ od lutego 1986 r. w szkole utworzono Pracownię Informatyki, wyposażoną już obecnie w 8 osobistych mikrokomputerów SM-1906, czyli "Meritum-1" (przeznaczonych dla uczniów) oraz jeden "Meritum-2" (dla wykładowcy). Mikrokomputery te tworzą sieć. Pozwala to na realizację bezpośredniej łączności między komputerem nauczyciela a komputerem ucznia (rys. 1); komputer "Meritum-2", oprócz funkcji mikrokomputera-wykładowcy spełnia funkcje "file servera".

Zgodnie z obecnymi tendencjami w Związku Radzieckim wykorzystywanie mikrokomputerów osobistych w szkole idzie w dwóch kierunkach:

- nauka przedmiotu "podstawy informatyki i techniki obliczeniowej" - przedmiot ten obejmuje takie zagadnienia, jak: algorytmizacja zadań, programowanie w języku algorytmicznym, prace na mikrokomputerze indywidualnym i na mikrokomputerach pracujących w sieci;
- nauka innych dyscyplin przedmiotowych z pomocą komputera.

Sposób, w jaki realizowany jest każdy z wymienionych kierunków nauki w szkole przy Ambasadzie Radzieckiej w Warszawie, omówimy nieco dokładniej w dalszej części artykułu.



Struktura lokalnej sieci mikrokomputerowej w szkolnej klasie komputerowej

Nauka nowego przedmiotu "Podstawy informatyki i techniki obliczeniowej" w klasach dziewiątych szkoły zorganizowana jest w sposób następujący:

- na lekcjach - raz w tygodniu - wykładana jest teoria zasad informatyki i techniki obliczeniowej,
- na ćwiczeniach prowadzone są praktyczne prace na minikomputerach działających w sieci: młodzież programuje w języku BASIC i rozwiązuje zadania za pomocą minikomputera, itp.

Na ćwiczeniach prowadzone są również lekcje z zakresu fizycznych, arytmetycznych i logicznych podstaw informatyki. Nauczanie nowego przedmiotu w klasach dziesiątych szkoły w pierwszym kwartale było zorganizowane w sposób analogiczny, jak w klasach dziewiątych. Następnie, do nauki zasad programowania podzielono uczniów na dwie grupy - pracownia informatyki dysponuje miejscami tylko dla 16 osób.

Zajęcia odbywają się raz w tygodniu, co tydzień następuje zmiana grupy.

Każda grupa na zajęciach bez komputera zajmuje się teoretycznymi zasadami programowania oraz opracowuje algorytmy, według których sporządza na zajęciach z komputerami programy w języku BASIC, wprowadza je do komputera, uruchamia usuwając błędy formalne, a następnie błędy merytoryczne oraz zapisuje gotowe programy na dyski elastyczne.

Na ćwiczeniach uczniowie klas dziesiątych opracowują programy dydaktyczne z innych przedmiotów, które wjdą do biblioteki użytkowych problemowo-ukierunkowanych programów do nauki różnych dyscyplin.

Nauka programowania w języku algorytmicznym BASIC odbywa się według metodyki ścisłego współdziałania wykładowcy z uczniem. Programy całkowicie poprawne formalnie i merytorycznie oraz prawidłowe pod względem dydaktycznym są zapisywane na dysku elastycznym, przy czym jak pokazano na rysunku, stacja dysków elastycznych jest podłączona do minikomputera przeznaczonego dla wykładowcy. Na dysku tym zapisane są liczne programy lekcyjne oraz programy sprawdzania wiedzy ucznia; między innymi na dysku (nauczycielskim) zapisany jest "Kurs nauki programowania w języku BASIC".

Kurs teoretyczny nauki programowania w języku BASIC składa się z 8 lekcji, z których każda podzielona jest na trzy części.

Uczeń po osiągnięciu do pamięci swojego komputera programu lekcji z "nauczycielskiego" dysku elastycznego, może go odczytywać na ekranie monitora tak długo, aż się nauczy na pamięć danej partii materiału. W lekcjach, oprócz opisów instrukcji języka i sposobu wykorzystywania ich w programach, zadawane są pytania, na które uczeń musi udzielić odpowiedzi. Pozwala mu to dokonywać własnej oceny przyswajania odczytywanego materiału. To samo może wykonywać nauczyciel, ponieważ jego nauczycielski komputer jest wyposażony (programowo) w identyfikator każdego ucznia (właściwie identyfikator mikrokomputera uczniowskiego). Nauczyciel, kontrolując ze swego komputera prace uczniów na ich komputerach, może przesyłać zadania każdemu uczniowi. Ma więc możliwość jednocześnie nauczać i sprawdzać przyswojenie materiału.

Sekwencje kolejno oddziałujących na siebie elementów: uczeń - personalny komputer nauczyciela - nauczyciel, można określić jako formę programowanej nauki. Warto zaznaczyć, że przy tym sposobie organizacji nauczania istnieją duże możliwości indywidualnego podejścia do procesu nauczania, a jednocześnie uczniowie zwolnieni są od obowiązku uczenia się w domu.

Uczeń, który łatwo przyswaja materiał i może sam sprawnie ułożyć program do rozwiązania konkretnego zadania, uzyskuje "zaliczenie" wcześniej, przy czym zaliczeniowe są jedynie programy zapisane w postaci algorytmicznej w "bibliotece algorytmów" podręcznika do przedmiotu "podstawy informatyki i techniki obliczeniowej". "Zaliczenie" oznacza, że uczeń opanował odpowiedni fragment wiedzy o komputerach. Wiedza ta, jak to zostało zaznaczone w materiałach "III Międzynarodowej konferencji o stosowaniu ETO w szkołach" - Lozanna 1981, wyraża się następującymi umiejętnościami:

- posługiwanie się ogólnoprzyjętymi terminami takimi, jak wprowadzanie informacji, wyprowadzanie informacji, peryferia, bit, bajt, uruchomienie, załadowanie programu, itp.;
- opracowanie najprostszych programów komputerowych;
- analizowanie zadań i przedstawianie ich w zapisie logicznym;
- proponowanie zadań zdalnych do wykonania na komputerze;
- ocenianie, czy program odpowiada zadaniu;
- proponowanie zmian danego programu w celu lepszego wykonania zadania.

Uczeń, który uzyskał "zaliczenie" - w następnym etapie zajmuje się opracowaniem programów dydaktycznych w zakresie tych opanowanych już przedmiotów. Lepsze programy zapisywane są na dyski elastyczne, a następnie włączane do banku wiedzy o określonych przedmiotach.

Ogólnie można powiedzieć, że komputerowe banki wiedzy praktycznie spełniają rolę informatorów. Zapisywana w tych bankach informacja jest tak uporządkowana, że umożliwia łatwe jej uzupełnienie, szybki dostęp do konkretnej informacji i możliwości operatywnej jej wymiany. Wszystko to znacznie ułatwia pracę z materiałem przeznaczonym do nauczania, wzbogaca zasoby informacji wykorzystywane na lekcjach, pozwala znacznie ekonomiczniej wykorzystywać czas przeznaczony na nauczanie.

Jak wynika z obserwacji, uczniowie z dużym zainteresowaniem i ochotą zdobywają wiedzę o komputerach. Powstają przy tym warunki dla znacznego wzmocnienia aktywności i samodzielności uczących się i przezwyciężenia negatywnego stosunku do zajęć szkolnych. Podobnie obserwuje się pozytywne oddziaływanie techniki komputerowej na pracę nauczyciela, na intensyfikację procesu edukacyjno-pedagogicznego.

Nauczanie innych dyscyplin przedmiotowych prowadzone z pomocą komputera możliwe jest tylko pod warunkiem posiadania odpowiednich programów dydaktycznych. Opracowanie takich programów wymaga, z jednej strony, znajomości materiału dydaktycznego przedmiotu nauczania, a z drugiej - dokładnej znajomości języka programowania, np: BASIC oraz szczegółowej wiedzy o możliwościach komputerów "MERITUM-1" i "MERITUM-2".

Programy zawierające materiał dydaktyczny niezbędny do prowadzenia lekcji z użyciem komputera muszą być dostarczone nauczycielowi danego przedmiotu. Na przykład do opanowania języka

BASIC na wyższym poziomie, niż jest to używane na lekcjach w klasach dziewiątych a zapisanego na dyskach elastycznych, niezbędny jest poradnik, w którym uczniom byłyby zaprezentowane różne sposoby programowania, w celu osiągnięcia odpowiednich efektów dydaktycznych.

Ponieważ brakuje na razie takiej pomocy naukowej, wspólnie z mgr Krzysztofem Krzemińskim został opracowany wstępny poradnik-podręcznik z tej dziedziny. Opracowanie to będzie służyło do opracowania jednej z ostatnich części programu nauczania informatyki i techniki obliczeniowej; cały kurs będzie zawierał:

- teoretyczny kurs programowania pt. "Podstawy informatyki i techniki obliczeniowej pod redakcją A.P. Jerszowa i W.M. Mońchowa,
- praktyczny kurs programowania w języku BASIC - 8 lekcji wydrukowanych - zapisany na dysku elastycznym.

Taki poradnik będzie nosił tytuł "Rozwiązywanie zadań na mikrokomputerze personalnym" i zostanie wydrukowany już w najbliższym czasie.

Logiczną konsekwencją będzie też doprowadzenie do triady: teoria, praktyka, przygotowanie uczących się do zawodowego ukierunkowania.

Opanowanie teoretycznych i praktycznych nawyków w zakresie programowania powinno być takie, aby znalazły one zastosowanie w praktyce. Wspomniany poradnik-podręcznik jest więc instrukcją działania, która pomoże wytworzyć takie przyzwyczajenia, to znaczy da możliwość opracowywania w formie komputerowej, programów pozwalających uczniom dostrzec już obecnie realne rezultaty z nauki przedmiotu "podstawy informatyki i techniki obliczeniowej". Zachowanie tej zasady pracy prowadzi do zgodności programów opracowywanych przez uczniów z programami nauczania poszczególnych przedmiotów i będą odpowiadały wymaganiom stawianym programom dydaktycznym. Dzięki temu szkoła, bez zaangażowania wykwalifikowanych programistów, będzie mogła rozpocząć formowanie pakietów programów użytkowych do poszczególnych przedmiotów. W szkole przy Ambasadzie prace w tym kierunku są już prowadzone. Zgromadzono programy dydaktyczne z algebry, geometrii, fizyki, techniki samochodowej, śpiewu, gimnastyki, chemii i języka angielskiego.

W celu udoskonalenia opanowanych już nawyków w opracowywaniu programów dydaktycznych, dobrze jest programy te omawiać na odpowiednich spotkaniach. W tym celu muszą jednak one być znane szerszemu gronu, np. przez opublikowanie opisów i listingów. W tym też celu grono uczniów rozpoczęło wydawanie drukiem szkolnych podręcznikowych zeszytów z zakresu informatyki i techniki obliczeniowej. Już treść pierwszego zeszytu nie została ograniczona do problematyki, która może interesować tylko uczniów klas 9 i 10, lecz opracowano go pod kątem zainteresowań także uczniów klas od szóstej do ósmej już obecnie zdobywających wiedzę w kółkach zainteresowań w zakresie informatyki.

Zespół opracowujący ogłosił konkurs na nazwę wydawnictwa. Wybrano nazwę "ALGORYTM", a dewizą stało się motto: "EMO²INTELEKT-KY". Rocznie będzie ukazywał się 5 zeszytów. Pierwszy numer "ALGORYTMu" już został wydany. Zawiera on:

- I. Nowości "Elektronika MO 1211"
- II. Wyniki konkursu
- III. Informatyka w szkole
- IV. Ogólne wiadomości o pracy "Meritum-1"
- V. Programowanie w języku BASIC
- VI. Programy nauczające: lekcja 1
 - słownik języka BASIC
 - muzyka
 - rozwiązywanie równań kwadratowych
 - sinusoida
 - czworokąt
 - egzaminator dla uczniów klas młodszych

VII. Grafika PPTO "Meritum-1"

VIII. Zadania dla zapaleńców

x/ Elektroniczna Maszyna Obliczeniowa

Wymieniony materiał opracowano na 20 stronioach.

W rozdziale VI przedstawione są tylko programy będące wynikami prac uczniów. Obecnie jest już przygotowywany do druku drugi numer "Algorytmu", do którego wejdą opracowane nie tylko programy dla uczniów dziewiątych i dziesiątych, lecz również programy dla uczniów klas siódmych i ósmych.

Tak jak poradnik-podręcznik "Rozwiązywanie zadań na mikrokomputerze", tak również i kwartalnik "ALGORYTM" mają jedyny cel: wykorzystać i udoskonalić wiedzę uczniów w zakresie informatyki. W ten sposób powstanie rezerwa intelektualna, która pomoże opracować kompletne pakiety programów użytkowych z poszczególnych przedmiotów i wykładowca nie musi dysponować całym pakietem programów po to, aby wykorzystywać komputer na lekcjach. Dla osiągnięcia określonych efektów, można i należy wykorzystać oddzielnie poszczególne programy, włączając je do materiałów dydaktycznych na lekcji. Pozwala to już we wstępnym etapie wykorzystywać komputer przy nauczaniu poszczególnych przedmiotów; spełnia on tutaj rolę instrumentu formalizacji wiedzy o świecie. Efektywna bywa ta lekcja, na której uczniowie są podstawowymi osobami działającymi, a nauczyciel - jest jedynie organizatorem ich pracy. Przede wszystkim to dla uczniów muszą być jasne zadania i osiągnięte końcowe rezultaty - wiedza, umiejętności, nawyki, motywacje i poznawcza samodzielność.

Wprowadzenie komputera do procesów nauczania uczyni go dostępnym i zrozumiałym dla każdego. Wszyscy będą umieli posługiwać się nim, a to oznacza, że szkoła potrafi z powodzeniem efektywnie realizować zadania wytyczone jej na XXVII Zjeździe KPZR. Bez specjalnych wymagań w odniesieniu do tworzenia jej specjalnych warunków potrafi przygotowywać specjalistów, na przykład operatorów-programistów.

Nasza szkoła dysponuje takimi możliwościami, a nagromadzone doświadczenie udowodniło, że uczniów bardzo interesuje programowanie i prace na mikrokomputerze. Należy się spodziewać, że wyrosną z nich wartościowi specjaliści, lubiący swą pracę, dążący do pogłębiania zdobytej wiedzy.

Nowa forma realizacji procesu wykształcenia wiąże się jednak z zasadniczą przebudową całego systemu nauczania, zmianą całej "technologii pedagogicznej", stylu, metod i środków nauczania. Niezbędna jest też radykalna zmiana metodologicznego myślenia, tak pedagogów-badaczy, jak również i nauczycieli-praktyków. Problemem kluczowym udoskonalenia całego systemu naukowo-wychowawczego jest realizacja wymagań stawianych lekcjom przez współczesność.

sprawozdania

Trzecia Jesienna Szkoła PTI

"Współczesne kierunki rozwoju informatyki" - Mrągowo 1986

Szkoła odbyła się w dniach 3-7 listopada 1986 r. Uczestniczyło w niej ok. 190 słuchaczy z całego kraju i ośmiu wykładowców z następującymi referatami:

- doc. Jan Madey (Uniwersytet Warszawski) - "Problematyka systemów operacyjnych na przykładzie systemu UNIX",
- prof. Zdzisław Pawlak (PAN) - "Komputerowe wspomaganie procesorów decyzyjnych",
- dr Stefan Sokołowski (Uniwersytet Gdański) - "Programowanie funkcjonalne",
- doc. Antoni Kreczmar (Uniwersytet Warszawski) - "Języki obiektowo zorientowane",
- doc. Jan Zabrodzki (Politechnika Warszawska) - "Cyfrowa generacja obrazów",
- prof. Cliff B. Jones (Manchester University) - "VDM and its application in program development",
- mgr Włodzimierz Grudziński (Uniwersytet Warszawski) - "Zastosowanie Prologu w bazach danych",
- doc. Stanisław Waligórski (Uniwersytet Warszawski) - "Informatyka szkolna i jej problemy".

Każdy dzień był dosyć pracowity, co ilustruje poniższy plan zajęć:

- 9 - 12³⁰ sesja poranna (wykład plus dyskusja)
- 15 - 18³⁰ sesja popołudniowa (wykład plus dyskusja)
- 20 - 21 konwersatorium

W dalszej części sprawozdania pokrótce przedstawiam treść poszczególnych wykładów.

Z. Pawlak w swoim referacie pt. "Komputerowe wspomaganie procesów decyzyjnych" zaprezentował inne spojrzenie na jeden z ośmioletnich dzisiaj tematów. Od początku istnienia komputerów były czynione próby zastosowania komputerów do wspomagania wnioskowania, jednakże dotąd nie udało się rozwiązać tego problemu ani od strony teoretycznej ani praktycznej. Prelegent wprowadził następującą klasyfikację wnioskowań:

- dedukcyjne, ● indukcyjne, ● potoczne

Wnioskowanie dedukcyjne jest stosowane w matematyce - próby zastosowania w innych dziedzinach zakończyły się fiaskiem.

Wnioskowanie dedukcyjne opiera się na teorii mnogości i logice. Nazywane jest niezawodnym gdyż przy prawdziwych przesłankach daje zawsze prawdziwe wnioski. Prowadzi od ogółu do szczegółu. Przykład: człowiek jest śmiertelny \rightarrow Kowalski jest śmiertelny. Wnioskowanie dedukcyjne znalazło zastosowanie w implementacjach komputerowych np. automatycznego dowodzenia twierdzeń; jest w pełni zbadane - można spotkać poglądy, że ta dziedzina została zakończona - i ma bogatą teorię. Wnioskowanie to wiąże z komputerami przede wszystkim takie nazwiska jak Alfred Tarski (1939), Haro Wang (1961) i Andrzej Trybulec (1980).

Wnioskowanie indukcyjne jest głównym narzędziem nauk eksperymentalnych i polega na uogólnieniu częściowych danych obserwacyjnych; prawdziwość jest określana z pewnym prawdopodobieństwem. Ma ono charakter odwrotny do wnioskowania dedukcyjnego, tzn. przechodzi od szczegółu do ogółu. Przykład: temperatura wody 100°C ⇒ woda wrze.

Opiera się na prawdopodobieństwie i statystyce, odbywają się próby tworzenia teorii. W implementacjach komputerowych znalazło zastosowanie w algorytmach uczących się, rozpoznawaniu obrazów i sygnałów, symulacji itd. Wiąże się to z takimi postaciami jak: John S. Mill i Kazimierz Ajdukiewicz (1965). Droga ta raczej jednak nie przyczyniła się do nowego spojrzenia na wnioskowanie indukcyjne.

Wnioskowanie potoczne jest rozpowszechnione w życiu codziennym i niektórych naukach humanistycznych. Przykład: każdy mówi, że Jasio jest głupi ⇒ Jasio jest głupi.

Prawdziwość wyników jest tu problematyczna, mimo to wnioskowanie potoczne znajduje zastosowanie w implementacjach komputerowych w najmłodniejszym w USA kierunku, w tzw. systemach ekspertrymialnych. Brak jakiegokolwiek metodologii stawia jednak uzyskiwane wyniki pod znakiem zapytania. Nieznana jest nam dotąd jakakolwiek teoria.

W dalszej części referatu prelegent przedstawił opartą na koncepcji tzw. zbiorów przybliżonych pewną metodę komputerowego wspomaganie wnioskowania indukcyjnego, jej raczej rokującą przyszłość. Istota tej metody była wyjaśniana na trzech przykładach praktycznych. Mimo pozornych różnic, we wszystkich przykładach sprawa sprowadza się do analizy tablic decyzyjnych a w szczególności ich dekompozycji i niesprzeczności. Wspólną cechą przeprowadzonych wnioskowań był brak pełnej wiedzy o interesujących procesach. Istnieją liczne metody matematyczne pozwalające wyciągnąć prawidłowe wnioski w warunkach niepełnej wiedzy (statystyka, zbiory rozmyte) nie zawsze jednak przydatne, np. ze względu na małą liczbę danych lub ich charakter. W tych sytuacjach wydaje się rokować nadzieję zaproponowana metoda zbiorów przybliżonych. Mierozróżnialność na bazie pewnych cech zmusza do przybliżania jednych zbiorów innymi. Celem tej teorii jest uzyskanie logiki wyciągania wniosków z "pewną dokładnością" oraz stworzenie matematycznych podstaw analizy tego typu rozumowania.

Przeglądu podstawowych zagadnień systemów operacyjnych wspieranego przykładami z modnego od pewnego czasu systemu UNIX dokonał w swoim referacie, pt.: "Problematyka systemów operacyjnych na przykładzie systemu UNIX" doc. J. Madey.

System operacyjny stanowi nadbudowę sprzętu i ukrywając tym samym pewne cechy sprzętu daje użytkownikowi maszynę wygodniejszą do rozwiązywania problemów. W szczególności zarządza pracą elementów komputera i realizuje podział jego zasobów pomiędzy użytkowników. Udostępnia liczne pakiety programistyczne i języki programowania.

Historia systemów operacyjnych jest już dosyć bogata i obejmuje takie produkty, jak mający złą reputację uniwersalny moloch OS/360 na maszynie IBM 360/370 czy ambitny MULTICS poprzez systemy badawcze typu THE czy RC4000 do prostych, dominujących na obecnych mikrokomputerach MS DOS, CP/M czy UNIX.

Według autora czymś wyjątkowym jest właśnie UNIX. UNIX to środowisko programistyczne obejmujące edytory, kompilatory, pocztę elektroniczną i inne. Zajmował on ma taką pozycję wśród systemów operacyjnych jak Pascal wśród języków programowania - mimo swych już 15 lat, staje się standardem na mikrokomputery. Ma wszakże, zważywszy na panujące kierunki, istotną wadę - nie jest "user-friendly".

UNIX jest rozwijany głównie w trzech ośrodkach: Research Bell Labs (wersje V5 do V8), Bell USG (od UNIX/RT do System V) i uniwersytet Berkeley (od wersji 2BSD do 4.2 BSD). Powstały liczne mutacje, takie jak Xenix firmy Microsoft dla mikroprocesorów Intel 8086/88 czy Unix z uniwersytetu w Toronto. Wiele cech UNIXa przejmują inne systemy np. MS IOS. W dalszej części autor zarysował podstawowe problemy systemów operacyjnych ilustrując je rozwiązaniami przyjętymi właśnie w UNIX.

Strukturę systemu, poza patologicznymi przypadkami można sprowadzić do dwu warstw, wewnętrznej i zewnętrznej. Warstwa wewnętrzna tzw. jądro jest odpowiedzialna m.in. za obsługę przerwań, przydział procesora, operacje wejścia/wyjścia, system plików oraz tworzenie i usuwanie procesów warstwy zewnętrznej. Warstwa zewnętrzna obejmuje procesy systemu, które mogą "trochę poczekać" na zasoby komputera i procesy użytkownika. W warstwie tej znajduje się też proces odpowiedzialny za konwersację z użytkownikiem - w UNIX wymiennalną tzw. powłoką (shell).

Równoczesna praca niektórych elementów instalacji komputerowej rodzi naturalną potrzebę równoległego przetwarzania programów lub nawet ich fragmentów. Spowodowało to narodziny takiej dyscypliny jak programowanie współbieżne. Zajmuje się ona notacją i technikami wyrażania potencjalnej równoległości.

Realizację programu sekwencyjnego nazywamy procesem sekwencyjnym. Dwa procesy są współbieżne, jeśli przedziały czasu ich wykonywania nie są rozłączne. Procesy współbieżne mogą być rozłączne - gdy nie współpracują ze sobą, lub interakcyjne - w przeciwnym razie.

W procesach interakcyjnych może wystąpić zjawisko tzw. blokady (deadlock) - istotną sprawą jest tu więc synchronizacja procesów, przy wykluczającym się dostępie do wszystkich zasobów. Pierwszym zaakceptowanym mechanizmem synchronizacyjnym były zaproponowane 20 lat temu przez Dijkstra semafory. Semafory są jednak mechanizmem niestrukturalnym, nie wykluczającym w przypadku złożonego systemu popełniania niewykrywalnego przez translator błędu - translator nie może sprawdzić prawidłowości używania semaforów, a np. kolejność operacji P/S/ jest bardzo istotna i może całkowicie zmienić wynik.

W połowie lat siedemdziesiątych za sprawą B. Hansena, Dijkstra i Hoare zaproponowano lepszy, strukturalny mechanizm tzw. monitor.

System operacyjny zarządza sprzętowymi i programowymi zasobami komputera stwarzając użytkownikowi iluzję wyłączności - dla każdego z nich tworzy, utrzymuje a następnie likwiduje komputer wirtualny. Podział zasobów między użytkowników realizują programy zarządzające. Przydział procesora procesom odbywa się według pewnych strategii. Można je podzielić na dwie podstawowe grupy: przydział z wyłączeniem (np. klasyczny podział czasu - time sharing) lub bez wyłączenia (np. nadawanie priorytetów procesom lub FIFO). Często stosuje się strategię łączną i taka realizacja odbywa się w UNIX - jest to forma łącząca podział czasu z priorytetami. Strategie przydziału pamięci operacyjnej zmieniały się wraz ze zmianą jej podstawowych cech. Możemy tu wyróżnić:

- strategie zakładające obecność w spójnym obszarze pamięci całego obszaru adresowego procesu,
 - . statyczny podział na strefy,
 - . dynamiczny podział na strefy,
 - . wymiana (swapping)
- pamięć wirtualna.

Przyjęta strategia miała oczywiście jakieś konsekwencje uwidaczniające się czy to w ochronie pamięci, fragmentacji czy migotaniu stron. W systemie UNIX przydział pamięci zmieniał się od wymiany w pierwszych wersjach systemu - do pamięci wirtualnej.

Istotnym elementem każdego systemu operacyjnego jest zarządzanie zbiorami. UNIX używa systemu plików opartego na drzewiastym, hierarchicznym katalogu. Każdy plik może być odczytany lub/i pisany lub/i wykonywany. Takie kombinacje praw dostępu do pliku są określane dla trzech grup użytkowników: właściciela, wcześniej określonej grupy i wszystkich.

Klasyczne, imperatywne programowanie polega na instruowaniu maszyny co ma robić, krok za krokiem kolejnymi instrukcjami programu napisanego w imperatywnym języku programowania, np. Pascala, Fortranie. Obecnie coraz częściej ciężar przenosi się na sformułowanie problemu a takie właśnie, właściwsze podejście daje przebieg ostatnich lat - według S. Sokołowskiego, autora referatu "Programowanie funkcjonalne" - programowanie funkcjonalne (functional programming, applicative programming, declarative programming).

Na przykładzie listy płać autor pokazywał zalety programowania funkcjonalnego, tj. skrócenie zapisu programu i poprawę jego czytelności. Inną zaletą jest brak efektów ubocznych - wszystko trzeba robić wprost. Oczywiście są i wady. Języki funkcjonalne wymagają bardzo dużo pamięci,

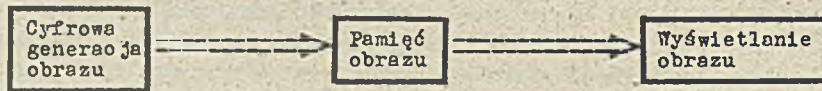
są również powolne. Nie wynika to jednak z samego języka a raczej faktu, że dotychczasowe architektury maszyn są dogodniejsze dla programowania imperatywnego. Można skrócić czas obliczeń wybierając właściwą kolejność działań - wiąże się to z pojęciem obliczania ohoętnego (eager evaluation) i leniwego (lazy evaluation).

Jeszcze bardziej optymistycznie wyglądają perspektywy. W językach funkcyjnych mamy możliwość łatwego dzielenia obliczeń na wiele niezależnych procesów, które mogą być równocześnie wykonywane w środowisku wieloprocesorowym - i zrobi to translator bez udziału programisty.

Teoretyczną podstawę jęz. funkcyjnych stanowi lambda - rachunek. Można w nim zdefiniować wszystko czego potrzebuje programista - języki funkcyjne mają więc moc nie mniejszą niż imperatywne; nie mają wprawdzie przypisań ale istnieje możliwość definiowania stałych, brak pętli jest wynagradzany rekursją (rekursja też nie jest cechą pierwotną - można ją zdefiniować). Głośno dziś języki specjalizacji też są językami funkcyjnymi.

O rastrowej technice graficznej - głównie z punktu widzenia sprzętu traktował referat J. Zabrodzkiego pt. "Cyfrowa generacja obrazów".

Schemat blokowy systemu rastrowego przedstawia rysunek



W bloku cyfrowej generacji obrazu z danych dostarczonych przez program aplikacyjny jest generowana scena, trójwymiarową scenę poddaje się przekształceniu geometrycznemu w celu utworzenia obrazu sceny w dwuwymiarowej płaszczyźnie ekranu i wreszcie generują się kolejne punkty obrazu w pamięci obrazu. Problem doskonale ilustruje fakt, że dla transformacji geometrycznej jednego punktu przestrzennego trzeba wykonać ok. 20 operacji zmiennoprzecinkowych.

Blok wyświetlania obrazu pobiera informację z pamięci obrazu, przekształca ją na postać analogową i wyświetla, może tu występować jeszcze, np. pamięć kolorów (bardzo szybka).

Centralnym elementem systemu jest pamięć obrazu. O wadze sprawy niech świadczy dane: przy rozdzielczości 1024 x 1024 i ośmiobitowej informacji o kolorze oraz przy założeniu, że obraz musi być powtarzany ok. 60 razy w ciągu sekundy aby nie migał i że pamięć obrazu musi być dostępna z dwu pozostałych bloków - musi ona mieć pojemność nie mniej niż 1 MB z czasem cyklu ok. 5 ns. Tymczasem dzisiejsze kości pamięciowe nie schodzą z czasem cyklu poniżej 25 ns. Wprowadzono więc pamięci typu VRAM, które są konstrukcją łączącą pamięć DRAM o typowym czasie cyklu, z możliwością równoległego odczytu/zapisu kilkuset słów z szybkim rejestrzem szeregowym.

Dalej przedstawiono inne powszechne rozwiązania sprzętowe na procesorach graficznych skończywszy.

Rosnące wymagania co do jakości grafiki powodują wzrost złożoności algorytmów transformacji sceny - np. metoda śledzenia promieni - a więc i gwałtowny wzrost potrzeb obliczeniowych. Prowadzi to do powstania pracujących samodzielnie lub współpracujących z komputerem tzw. stacji graficznych. Są to bardzo skomplikowane architektury o dużej mocy obliczeniowej i wysokiej cenie; przedstawiono kilka przykładów.

W referacie omówiono też podstawowe typy monitorów kolorowych. Podano wiele interesujących przykładów i danych liczbowych o graficznych układach VLSI.

A. Kreczmar wygłosił referat na także w ostatnich latach aktualny temat, a mianowicie "Języki obiektowo zorientowane". Zamiast wkleść się w niejasne definicje, co to za języki zdefiniujmy je - za radą prof. Blikle - przez wskazanie. Do ważniejszych należą Simula - 67, Smalltalk - robiący według referenta prawdziwą furorę, Loglan - zrobiony w Polsce, i Paragon.

Soharakteryzujemy - za referentem - podstawowe cechy tych języków. Obiekt jest strukturą utworzoną według pewnego wzoru, tzw. klasy. Z jednej klasy można utworzyć dowolną liczbę unikalnych obiektów. Definicja klasy może być poglądowo przedstawiona w sposób następujący:

`nazwa_klasy: .class (parametry)`

`atrybuty klasy`

`ciąg akcji`

`end nazwa_klasy;`

Atrybuty klasy mogą znowu być klasami. Zwykle jednak występują też atrybuty, które nie mają struktury wewnętrznej – są to atrybuty ilości i jakości. Wielkości liczbowe są typami pierwotnymi integer i real. Typy jakościowe użytkownik może zdefiniować sam, np: kolor = (biały, niebieski, zielony). Atrybutami mogą być również procedury i funkcje. Klasa może mieć także parametry, sposób przekazywania których jest identyczny jak w procedurach. W klasie można zdefiniować pewien ciąg akcji wykonywanych w momencie generowania obiektu danej klasy.

Wcześniej zadeklarowany obiekt `nazwa_obiektu: nazwa_klasy` generuje (tworzy) się w sposób następujący:

`nazwa_obiektu : = new nazwa_klasy;`

W trakcie generacji parametry formalne są zamieniane na aktualne. Po utworzeniu obiektu określa się jego atrybuty. Dostęp do atrybutu uzyskiwany jest w sposób tzw. zdalny, tzn. wygląda następująco:

`nazwa_obiektu . nazwa_atrybutu`

Atrybuty nie będące klasą określa się za pomocą instrukcji podstawienia

`nazwa_obiektu . nazwa_atrybutu := wartość`

Atrybuty klasowe określa się instrukcją generacji `new` obiektu. Korzystając z możliwości definiowania w klasie pewnych akcji możemy określać atrybuty (w tym także klasowe) samoczynnie bez konieczności wykonywania tego z zewnątrz.

We wszystkich językach obiektowych istnieje możliwość tworzenia hierarchii klas, odwzorowywanej automatycznie na generowanych obiektach – jest to tzw. dziedziczenie. Klasy wywodzące się z jakiejś klasy dziedziczą jej cechy. Za autorem referatu zilustrujemy to przykładem. Definiuje:

```
pojazd: class
    waga: real;
end pojazd;
samochód of pojazd: class (nr_prawa_jazdy: integer);
end samochód;
autobus of samochód: class
    liczba_miejsa: integer;
end autobus;
ciężarówka of samochód: class
    pojemność: real;
end ciężarówka;
```

oddają następującą strukturę jak na schemacie obok

Dalej można zadeklarować obiekty, np.

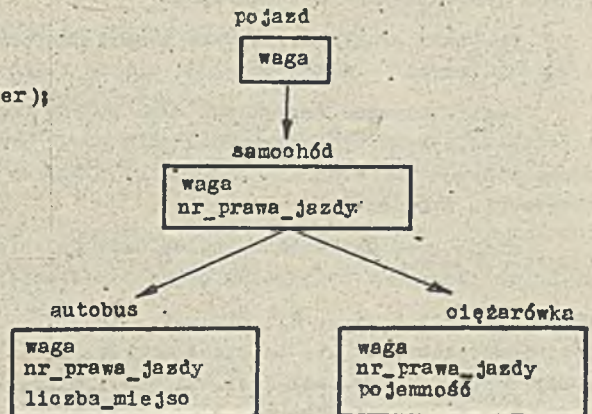
`ikarus: autobus;`

utworzyć je

`ikarus: = new autobus (19876);`

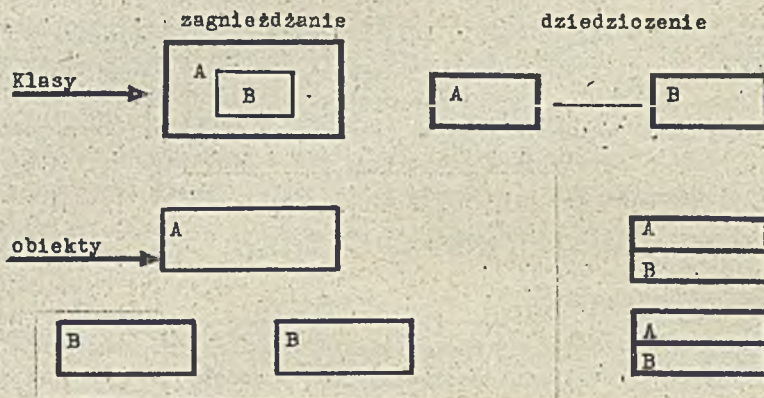
i określić atrybuty

`ikarus . waga: = 10000; ikarus.liczba_miejsa: = 89;`



W trakcie generacji obiektu, akcje klasy dziedziczonej - jeśli takie zostały wyspecyfikowane - są wykonywane przed akcjami klasy dziedziczącej. Należy podkreślić, że w trakcie generowania automatycznie (bez potrzeby użycia new) na potrzeby obiektu dziedziczącego jest tworzony obiekt dziedziczony. Inaczej jest w przypadku zagnieżdżenia - kolejnej formy spotykanej w językach obiektowo-zorientowanych.

Tu po wygenerowaniu obiektu danej klasy można wygenerować wiele zagnieżdżonych obiektów zależnych od tego obiektu. Różnice między zagnieżdżaniem i dziedziczeniem ilustruje poniższy rysunek.



Ze względu jednak na trudności we współistnieniu obu wyżej wymienionych konstrukcji spotykamy się z różnymi, bardziej lub mniej ograniczonymi implementacjami w wymienionych na wstępie językach. Mimo że - według autora wystąpienia - języki obiektowo zorientowane stawiają duże wymagania praktyczne i teoretyczne w stosunku do użytkownika, to ich stosowanie przynosi korzyści gdyż ułatwia odwzorowanie opisywanego świata rzeczywistego, a szybciej utrzymany program będzie również szybciej działał.

Po uporaniu się z podstawowymi problemami baz danych stawia się im dzisiaj nowe wymagania, wśród których występują również takie, jak przechowywanie wiedzy i wnioskowanie, które wchodzi w zakres tzw. systemów ekspertowych czy mówiąc ogólniej obszar sztucznej inteligencji.

Do najbardziej obiecujących ze stosowanych do podołania tym wymaganiom narzędzi należy powstałe na początku lat 1970 tzw. programowanie w logice. Zainteresowanie nim wzrosło po przyjęciu wygeneracji Prologu, najpopularniejszego przedstawiciela tego kierunku, jako podstawowego języka programowania systemowego.

Powyzsza problematyka była treścią referatu W. Grudzińskiego pt. "Zastosowanie Prologu w Bazach Danych" - autor odwołuje się do relacyjnych baz danych.

Logika używana dotychczas jako aparat formalny, np. w różnych zagadnieniach sztucznej inteligencji może być traktowana jako język programowania. Zdania logiki uzupełnione informacjami sterującymi mogą być uważane za programy a obliczenie polegać będzie na kierowanym poszukiwaniu dowodu.

Język programowania w logice jest podzbiorem logiki pierwszego rzędu nazywanym klauzulami Horna. Mogą one przybierać jedną z postaci:

- implikacja: $A \leftarrow B_1, B_2, \dots, B_n$
 - stwierdzenie (gdy nie ma przesłanek): A
 - zaprzeczenie (gdy brak jest wniosku): $\sim (B_1, \dots, B_n)$
- gdzie: B_i są przesłankami a A wnioskiem.

Do formalnego wnioskowania umożliwiającego wyprowadzanie ze zbioru zdań innych zdań używa się zwykle reguły rezolucji. Mówi ona, że ze zdań $Y \leftarrow X$ i $X \vee Z$ wynika zdanie $Y \vee Z$.

W bazach danych mamy zwykle duży zbiór niezinterpretowanych danych przechowywanych w rekordach oraz metadane tworzące tzw. schemat bazy pomagający interpretować dane i wyprowadzać z nich nowe informacje, przy czym różne funkcje systemu wspomagają inne języki (język definicji danych, język zapytań itd.). Tymczasem pozwala uporać się z tym ten sam język programowania w logice. Jest to język deklaracyjny, tzn. opisuje co chce się osiągnąć bez podania jak.

Opracowany już w 1972 r. w Marsylii PROLOG (PROGRAMMATIONEN LOGIQUE) należy do najbardziej znanych obecnie języków programowania w logice. Jest ciągle jeszcze udoskonalany - zmiany zmieniają w kierunku zwiększenia siły wyrazu i efektywności wykonania.

W Prologu nie ma w zasadzie pojęcia typów danych, choć są implementacje, np. Prolog Turbo na IBM PC, gdzie wprowadzono proste typy danych. Występują obiekty proste - zwykle są to stałe, i złożone - drzewa i listy. Opisy obiektów nazywane są termami. Obiekty o zmiennej strukturze nazywane są zmiennymi. Mogą one być związane z termami i wtedy reprezentowany przez nie obiekt staje się lepiej określony. Programem jest zbiór procedur składających się z klauzul.

Przez swoją zwężkość i moc Prolog nadaje się do szybkiego tworzenia prototypów baz danych o dużej liczbie reguł lecz niezbyt dużej liczbie danych, języków zapytań i translatorów. Łatwo pozwala dołączyć, lub wydobyć wiedzę z bazy danych; ma jednak i wiele wad, np. małą szybkość i duże potrzeby pamięciowe. W referacie krótko pokazano jak tworzy się w Prologu relacyjną bazę danych.

Jednoczne traktowanie programów i danych rodzi problemy z ochroną, współbieżnością - aktualizacja jest realizowana jako złożenie dwóch operacji "usuń" i "dodaj".

Dedukcyjne właściwości Prologu sprawiają, że łączy się go jako tzw. składnik dedukcyjny z istniejącymi systemami zarządzania bazami danych.

Głośną i w naszym kraju tematyką komputerowo wspomaganego nauczania zajął się Stanisław Waligórski w swoim wystąpieniu pt. "Informatyka szkolna i jej problemy".

Przez wprowadzenie informatyki do szkół autor referatu rozumie wsparcie komputerami jakichkolwiek zajęć prowadzonych w szkole, nie tylko zajęć z elementów informatyki. Także nie powinno być celem kształcenie z uczniów - programistów lecz przekazanie podstawowej wiedzy o użytkowaniu i stosowaniu informatyki.

Można tu wyróżnić trzy główne kategorie problemów:

- organizacyjno-techniczne i ekonomiczne problemy wprowadzania informatyki do szkół;
- prawidłowe korzystanie z posiadanych narzędzi informatycznych (sprzętu i oprogramowania),
- wytwarzanie i doskonalenie narzędzi informatycznych.

W pierwszej kategorii do najważniejszych zaliczyć należy:

- przygotowanie nauczycieli,
- przygotowanie pomocy naukowych, podręczników, oprogramowania,
- dostarczenie do szkół odpowiednich komputerów,

przy czym według autora decydujący jest element pierwszy a pozostałe dwa jedynie pogorszą sytuację, oczywista jest więc kolejność działań.

Głównym celem zajęć z przedmiotu "elementy informatyki" powinno być nauczanie metod rozwiązywania prostych problemów za pomocą komputera. Realizować się to ma w bezpośrednim kontakcie ucznia z komputerem - także po godzinach lekcyjnych.

Oczywiście komputer powinien być wykorzystywany i w innych przedmiotach. Nauczanie za pomocą komputera powinno być tak prowadzone, aby uczeń mógł wykazać się samodzielnością i aktywnością. Należy go uczyć prawidłowego podejścia do rozwiązywania problemów. Błędy powinny mobilizować do nowych prób.

Podano także minimalne wymagania dotyczące mikrokomputera szkolnego. Przemysł krajowy wyraża się z tym uporać. Profesjonaliści uporać się też z pewnością z innymi warunkami pracy oprogramowania dydaktycznego - na sprzęcie innym, o mniejszych możliwościach. Główne trudności leżą jednak w sferze dydaktycznej. Komputer szkolny powinien być taki, aby można było adaptować na niego oprogramowanie już istniejące.

Względy ilościowe nakazują aby zacząć informatyzację szkół - według autora - od średnich szkół ogólnokształcących i wybranych zawodowych.

Wytwarzanie oprogramowania jest w istocie wąskim gardłem w rozwoju komputerów. Uzyskiwanie poprawnych programów jest sprawą pierwszorzędnej wagi nie tylko z punktu widzenia samej sztuki programowania, lecz przede wszystkim z bardziej wymiernych powodów, takich jak ekonomiczne czy bezpieczeństwa pracy systemów opartych na komputerach. Różne też były podejścia do problemu np. testowanie programów, które przecieł dowodzi jedynie istnienia a nie braku błędów. Dowodzone poprawności algorytmów.

Wiele błędów powstaje we wczesnych fazach projektowania - niedostatecznie szybkie wykrycie ich podnosi koszty weryfikacji. Dobrą metodą byłoby więc systematyczne (staranne) sprawdzanie kolejnych faz. Zamiast kosztownego dowodzenia poprawności całego programu można by więc wprowadzić uzasadnienie poprawności w trakcie jego konstrukcji. Wymaga to opisanie - wyspecyfikowania - funkcji w formalnym języku specyfikacji.

W ostatnich latach rozwijają się następujące podejścia:

● weryfikacja specyfikacji:

specyfikacja jest pisana w języku specyfikacji natomiast projekt i ewentualnie kod w normalnym języku implementacyjnym; poprawność projektu jest określana w drodze rozłożenia dowodu na mniejsze kroki. To podejście reprezentują VDM,

● transformacja syntaktyczna:

specyfikacja jest klarowną, wykonywalną funkcją; najbardziej znanym przykładem tego podejścia jest projekt CIP,

● konstrukcyjna matematyka:

specyfikacja jest tu zdaniem twierdzenia z konstrukcyjnego dowodu, z którego może być wyekstrahowany program.

Konstruowanie poprawnych programów było tematem referatu C.B. Jonesa pt. "VDM and its application in program development".

Autor przeprowadził dyskusję modelowo ukierunkowanych specyfikacji VDM i specyfikacji własnościowo ukierunkowanych. Specyfikacje składają się z opisu typów danych oraz warunków wejściowych i wyjściowych. Pokazana została użyteczność modelowo ukierunkowanych specyfikacji w zakresie architektury systemu. Rozbudowanym przykładem zilustrowano aspekty systematycznego konstruowania metodą VDM.

Opracował:

Wiesław BABCZENKO

Samodzielna Pracownia

Rozwoju Komputerów IMM

Komputeryzacja w dydaktyce i badaniach pedagogicznych -
Ogólnopolskie seminarium technologii kształcenia

Kolejne, już XIII, Ogólnopolskie Seminarium Technologii Kształcenia odbyło się w Poznaniu w dniach 24-25 września 1986 r. Zostało ono zorganizowane przez obchodzący 20-lecie swego istnienia Zakład Technologii Kształcenia (poprzednio Zakład Nowych Technik Nauczania) Uniwersytetu im. Adama Mickiewicza w Poznaniu oraz Instytut Polityki Naukowej, Postępu Technicznego i Szkolnictwa Wyższego w Warszawie. Obradom seminarium przewodniczyli jego kierownicy naukowcy - dr Franciszek Januszkiewicz i doc. dr hab. Wacław Strykowski.

Otwarcia seminarium dokonał Rektor UAM w Poznaniu, prof. dr hab. Jacek Fisiak. W swoim wystąpieniu zwrócił uwagę na rolę i funkcję nowoczesnych metod i środków kształcenia w uczelniach wyższych w kontakcie doświadczeń, jakie w tym zakresie ma Uniwersytet Poznański. Szczegółowy program komputeryzacji UAM przedstawił prorektor tej uczelni, prof. dr hab. Waldemar Żukowski.

W pierwszym dniu obrad zostały wygłoszone dwa podstawowe referaty plenarne, w których problematyka komputeryzacji szkolnictwa została przedstawiona na szerszym tle społecznym i edukacyjnym. Prof. dr hab. H. Muszyński w referacie pt. "Spółczesna cywilizacja naukowo-techniczna i informatyczna - szanse, nadzieje i zagrożenia edukacyjne" w sposób bardzo interesujący wskazał na korelacje między rozwojem techniki, technologii, zmianami zachodzącymi w sferze tzw. obsługi masowej a celami, treściami i środkami kształcenia. W nowoczesnych społeczeństwach, powszechne stosowanie metod i technik komputerowych w różnych dziedzinach działalności, w sposób naturalny ujawnia potrzebę ciągłej modernizacji treści kształcenia, a szczególnie korelacji i integracji treści informatycznych z treściami przedmiotów ogólnokształcących i ogólnozawodowych. Aspekty dydaktyczne procesu komputeryzacji oraz możliwości zastosowania komputerów w badaniach pedagogicznych były tematem referatu prof. dr hab. T. Lewowickiego. Prof. T. Lewowicki wyeksponował znaczenie poprawnie sformułowanych celów kształcenia, które umożliwiają właściwy dobór treści, a także metod i środków kształcenia. W swoim wystąpieniu podzielił się także refleksjami na temat skutków nieuzasadnionego użycia sprzętu komputerowego oraz wskazał te dziedziny i obszary działalności edukacyjnej, w których sprzęt ten może twórczo inspirować uczniów (studentów), pełniąc funkcję nowoczesnego narzędzia pracy i środka dydaktycznego.

W programie pierwszego dnia seminarium przewidziano również pokaz mikrokomputera szkolnego "ELFRO-800 junior" oraz informację dotyczącą jego parametrów użytkowych, którą przekazał mgr inż. P. Frzysztofiak. Tego samego dnia wieczorem w ośrodku konferencyjnym UAM "Orlinek" odbyła się dyskusja na temat "Nowe technologie informacji w szkolnictwie wyższym". Wprowadzeniem do wieloaspektowej dyskusji był referat dr B. Ileożko dotyczący niektórych problemów pedagogicznych, technicznych i prawnych związanych ze stosowaniem nowych technik i technologii przekazywania informacji.

W drugim dniu seminarium wygłaszane były referaty i komunikaty z badań, odbył się też pokaz programów dydaktycznych. Dużo uwagi poświęcono:

- oprogramowaniu dydaktycznemu (doc. dr hab. W. Strykowski - "Procedura przygotowania kompleksowych programów dydaktycznych jako elementów nauczania-uczenia się multimedialnego", doc. dr hab. A. Szewczyk - "Metodyka badań komputerowych programów dydaktycznych"),
- wykorzystaniu mikrokomputerów w procesie kształcenia (dr inż. S. Kwiatkowski - "Metody i techniki komputerowe w kształceniu umiejętności projektowania", mgr J. Bejgerowski - "Zastosowanie mikrokomputerów w nauczaniu języków obcych", dr A. Puch, dr A. Zajac - "Trójkanałowy system kontroli i oceny wiedzy uczniów przy pomocy mikrokomputera", mgr inż. H. Maciąg-Sternik, dr inż. A. Mollrosz - "Nauczanie matematycznych metod planowania z zastosowaniem mikrokomputera"),
- wykorzystaniu mikrokomputerów w badaniach naukowych (dr J. Paluchowski - "Wykorzystanie mikrokomputerów w procesie badawczym w psychologii", mgr J. Morbitzer - "Mikrokomputerowa implementacja trójjęzycznego słownika z zakresu technologii kształcenia", mgr M. Jankowski - "Zastosowanie mikrokomputera w analizie metodyczno-statystycznej testów dydaktycznych"),
- infrastrukturze uczelni wyższych (dr K. Wenta - "Komputeryzacja dydaktyki w programie rozwoju Uniwersytetu Szczecińskiego"),
- przygotowaniu nauczycieli (dr H. Zaczęński - "Niektóre problemy mikrokomputeryzacji w zakresie doskonalenia i doskonalenia nauczycieli").

W bogatym programie seminarium nie zabrakło miejsca na prezentację najnowszego sprzętu mikrokomputerowego, pokazy profesjonalnych programów edukacyjnych oraz wystawę materiałów dydaktycznych i wydawnictw związanych tematycznie z technologią kształcenia.

Seminarium, w odczuciu uczestników spełniło istotną funkcję integrującą pracowników naukowo-dydaktycznych uczelni wyższych. Analiza dorobku poszczególnych ośrodków akademickich wskazuje na rosnące zainteresowanie celowym i planowym wykorzystaniem metod i technik komputerowych w procesie kształcenia i zarządzania w szkołach wyższych.

Opracował:

Stefan M. Kwiatkowski
Politechnika Warszawska

NOWOŚCI TECHNICZNE

Odpowiedniki komputerów osobistych IBM w USA

Na rynku amerykańskim pojawia się coraz więcej mikrokomputerów, które realizują programy pisane na IBM PC, a których parametry są konkurencyjne w stosunku do pierwowzorów.

W roku 1985 sprzedano w USA 7182 tysiące komputerów osobistych w tym 1493 IBM PC (zarówno XT jak i AT) oraz 1089 tys. kompatybilnych, co stanowiło odpowiednio 20,7 oraz 15,7% rynku tych komputerów. Przewidywania na rok 1986 mówiły o 8211 tys. mikrokomputerów, z czego 1837 tys. IBM PC, co oznacza utrzymaniu udziału tych komputerów na poziomie 21%. Natomiast liczba sprzedanych komputerów osobistych kompatybilnych z IBM PC, ale wytwarzanych przez inne firmy miała wzrosnąć do 2165 tys. Procentowy udział tych komputerów osiągnąłby więc 24,9%, to znaczy w wyniku szybkiego tempa wzrostu przekroczyłby poziom IBM.

Główną firmą specjalizującą się w mikrokomputerach IBM - podobnych jest Compaq Computer Corp. z Houston. To stosunkowo młode przedsiębiorstwo (nie istniało jeszcze w 1981 r.) sprzedało do końca 1985 r. 375 tys. komputerów osobistych, które zyskały sobie wysoką ocenę ekspertów. Jego udział w rynku mikrokomputerów poza IBM-owskich wynosił 22,5% podczas gdy druga firma na tej liście, Zenith Data Systems z Glenview miała tylko 14,7% (245 tys. komputerów). Firma ta dostarcza mikrokomputery dla instytucji rządowych, napotykając zresztą na rosnącą konkurencję. Trzecią z kolei firmą jest Tandy Corp. z Fort Worth w Teksasie. Udział jej wynosi 11,4% (150 tys. komputerów). Oferuje ona swe wyroby po konkurencyjnych cenach sprzedając je w sklepach firmy Radio Shack. Najtańszy model 1000 kosztuje tylko 699 dolarów z monitorem monochromatycznym i 999 dol. z monitorem kolorowym.

Na czwartym miejscu jest AT&T Corp. firma znana z opracowania systemu operacyjnego Unix, która oferuje maszyny 6300 i 6300 Plus. Te ostatnie działa 25% szybciej od AT IBM. Pracuje ona zarówno pod DOS-em, jak i Unixem. Udział AT&T wynosi 9% (150 tys. komputerów). Wyroby tej firmy są dodatkowo przystosowane do pracy w sieciach.

Zaczynają się też pojawiać pierwsze odpowiedniki IBM PC wytwarzane na Dalekim Wschodzie, głównie w Korei Płd i na Tajwanie. Przykładem może być też Model D sprzedawany przez firmę Leading Edge Inc. z Canton (w stanie Massachusetts) w cenie 1300-1400 dolarów, który produkowany jest w Korei przez Daewoo Ltd. W Korei też wytwarza się komputer Tele CATF-286, będący odpowiednikiem IBM PCAT z 20 Mbajtową pamięcią operacyjną i monitorem 640 x 400 (cena 2995 dolarów). Ogółem około 70 wytwórców azjatyckich oferuje mikrokomputery kompatybilne z IBM PC po cenach często 1000 lub więcej dolarów niższych od maszyn IBM.

Wśród droższych modeli przykładami konkurencyjnych opracowań mogą być Vectra firmy Hewlett-Packard Co. i Professional Computer firmy Texas Instruments, jednakże nowe wersje sprzętu IBM mają zwykle zalety tych opracowań. Czasami lepsze parametry uzyskuje się stosując własne, szybsze elementy pamięciowe, jak w maszynie Business-Pro firmy Texas Instruments, gdzie zastosowano kostki

dynamiczne o pojemności 256 kbitów i czasie dostępu 150ns na dodatkowym pakiecie o łącznej pojemności 3 Mбайты, dołączonym bezpośrednio do pakietu procesora, co zmniejsza opóźnienia istniejące na szynie. Komputer ten, nieco droższy od PC AT, jest wyposażony w 14 łączówek dodatkowych (o 6 więcej niż komputer IBM) i jest popularny wśród systemów dla wielu użytkowników np. wersja z ośmioma terminalami i systemami operacyjnymi Xenix.

Electronics nr 23/86

Komputer osobisty IBM RT

Na początku 1986 roku firma IBM Corp. ogłosiła parametry swego komputera osobistego nazwanego RT, od RISC (Reduced Instruction Set Computer - komputer o uproszczonej liście rozkazów, zob. "Techniki Komputerowe nr 4/86, str. 3). Jest to właściwie stanowisko robocze pracujące pod systemem operacyjnym AIX (Advanced Interactive Executive - rozwinięte zarządzanie interakcyjne), będącym ulepszoną wersją systemu Unix V.

Jest to komputer 32-bitowy, wobec czego RT może stanowić konkurencję dla znanych firm w tej dziedzinie jak Apollo Computer, DEC i Sun Microsystems. Firmy te jednak stwierdzają, że RT nie ma dostatecznie rozbudowanej grafiki i własności sieciowych. Rynek stanowisk roboczych rozwija się najszybciej w przemyśle komputerowym. Wg oceny Dataquest Inc. w roku 1985 sprzedano 21 tysięcy jednostek za około 735 mln dolarów, a w roku 1989 przewiduje się 191 tysięcy stacji za 2,7 mld. dolarów. Chcąc wejść na ten rynek IBM ustaliła dość niską cenę (8636 dolarów) za najprostsz system obejmujący 1 Mбайты pamięci wewnętrznej, 40 Mбайтыowy dysk sztywny i monochromatyczny monitor. Cena ta dotyczy zakupów obejmujących co najmniej 50 jednostek. Przy mniejszych ilościach system taki kosztuje 13085 dolarów. Natomiast modele rozbudowane są bardzo drogie, gdyż wolnostojący RT z 19-calowym monitorem kolorowym o dużej rozdzielczości, pamięcią wewnętrzną o pojemności 2 Mбайты i sztywnym dyskiem o pojemności 70 Mбайтыów kosztuje 40 tys. dolarów i nie stanowią konkurencji np. dla stacji Micro VAX II. Te rozbudowane modele mają system graficzny 5080, a ponadto są kompatybilne z PC AT - mają dodatkowe łączówki na pakiety 16-bitowe.

IBM ma nadzieję, że inne firmy przyjmą RT za podstawę swoich opracowań, nie wiadomo jednak czy opracowujący oprogramowanie łatwo przejdą z Unixa na AIX, który z nową kostką zarządzania pamięcią ma 40-bitową przestrzeń adresową pamięci wirtualnej. Na razie nieliczne firmy software'owe zajęły się RT. Można tu wymienić Silver-Lisco, Cadre Technologies Inc. Interleaf, Applix Inc. i Internece Corp. Przygotowywany jest symulacyjny program Laser do pakietów i obwodów scalonych stosowany przy weryfikacji projektowania i wytwarzania programów testujących.

Future Computing Inc. przewidywała sprzedaż od 20 do 25 tys. RT w 1986 roku. Inne prognozy (Forrester Research) mówiły o 12-15 tysiącach, z czego ok. 60% dla wielu użytkowników. Analogiczne przewidywanie dla Micro VAX II mówi o 9-12 tys. systemów, przy czym od maja 1985 roku sprzedano już 12 tys. tych stacji, mniej więcej tyle samo, co stanowisk Apollo w ciągu trzech lat.

Procesor RT zbudowany jest na układach n-MOS i wykonuje 84 spośród swych 118 rozkazów w pojedynczym cyklu o długości 170 ns. Daje to 1,6-2,1 mln rozkazów na sekundę - szybciej aniżeli wspomniane stanowisko firmy DEC.

Dostawy modeli RT przewidziane były na marzec 1986, lecz terminy opóźniły się gdyż na początku października podano, że w niecały miesiąc po dostarczeniu pewnych modeli wprowadzone są modyfikacje i obniżki cen. Obejmowały one dwukrotne zwiększenie pojemności pamięci wewnętrznej, trzykrotne przyspieszenie operacji zmiennego przedziału, oraz układ dwukrotnie zwiększający liczbę użytkowników do 16. Obniżka cen wyniosła 20-32% w zależności od modelu, ale i tak najtańsza konfiguracja kosztuje 7 tys. dolarów.

Electronics nr 4 i 32/86

Znikają różnice pomiędzy komputerami osobistymi Apple i IBM

W ostatnich latach potencjalny nabywca komputera osobistego stawał przed dylematem czy wybrać Macintosha firmy Apple z myszą do przesuwania wskaźnika po ekranie, fantazyjną grafiką i wyborem okien, czy IBM PC lub jego odpowiednika. Każdy wybór wprowadzał pewne ograniczenia. Mając Macintosha nie można było korzystać z ogromnego oprogramowania IBM, które stało się standardem przemysłowym, zaś dla osiągnięcia cech komputera Apple na PC trzeba było wprowadzać zbyt wiele modyfikacji, które czyniły go bardzo wolnym.

Ostatnie oferty tych firm idą w kierunku zmniejszenia tych różnic. Apple, dla której rok 1986 wypadł korzystnie, przygotowuje dwa nowe modele, których parametry mają być ogłoszone na początku marca 1987 r. Pierwszy z nich to Macintosh SE (od System Enhancement - usprawnienie systemu), który przypomina oferowany wcześniej Mac Plus, lecz posiada nową mysz, nową klawiaturę i mechanizm do dołączania dużych monitorów ekranowych. Drugi to Macintosh II, znacznie większy od swych poprzedników. Pozwala on dołączać różne układy, m. in. emulować IBM PC, mimo iż współzałożyciel Apple Steven Jobs niegdyś stwierdził, że Mac nigdy nie będzie tego robił.

Z kolei IBM planuje zastąpić dotychczasowe modele PC trzema nowymi modelami, z których wszystkie posiadać będą cechy jakie posiada Mac - sterowanie myszą, błyskotliwą grafikę i dyski elastyczne o średnicy $3\frac{1}{2}$ cala. Ma to na celu nie tylko emulację Apple, ale przede wszystkim walkę konkurencyjną z firmami wytwarzającymi odpowiedniki PC jak Compaq, Leading Edge, Epson i inne. Firmy te zmniejszyły udział IBM na rynku PC z 29 do 22%, a tym samym zyski czołowego wytwórcy, które spadły w czwartym kwartale 1986 o 48% w stosunku do roku ubiegłego. Nowe modele PC mają poprawić sytuację w tym zakresie. Nie wiadomo jednak jakie stanowisko zajmą użytkownicy nowych maszyn ze względu na pewną ich niekompatybilność z poprzednimi modelami.

Time z 1987.02.02

Uniezależnienie oprogramowania od rodzaju komputera

Użytkownicy komputerów mieli zawsze kłopoty z koniecznością zmiany oprogramowania przy zmianie komputera, na jakim pracowali. Obecnie podjęto inicjatywę, która powinna usunąć te problemy. Europejscy dostawcy komputerów: ICL, Bull, DEC, Ericsson, Hewlett-Packard, Nixdorf, Philips, Olivetti, Siemens i Unisys, opracowali w Amsterdamie tzw. X/OPEN system mający na celu standaryzację oprogramowania. Opiera się ona na systemie operacyjnym Unix firmy AT&T. Uzyskanie powszechnego zaakceptowania tego systemu zapewni lepszą kompatybilność języków programowania, kompilatorów, stanowisk roboczych i zarządzania bazami danych. Powinno to przynieść oszczędności sprzedawcom i użytkownikom oraz zwiększyć zakres i dostępność oprogramowania.

The Observer z 86.11.30

Mysz dla niewidomych

W jednym z ośrodków badawczych IBM w USA opracowano urządzenie podobne do konwencjonalnej "myszy" przesuwającej kursor na ekranie, służącej tu jednak do tłumaczenia znaków i symboli z ekranu na język Braille'a.

W doświadczalnym urządzeniu na powierzchni śledzonej znajduje się sześć białych tłoczków, które podnoszą się, gdy kursor przesuwa się po ekranie, wskazując na jakim symbolu się zatrzymał.

Niewidomy przesuwając kursor po ekranie odczytuje znaki, tak jakby palcami przesuwał po stronie z pismem Braille'a.

System wykorzystuje mysz, specjalny pakiet adapterowy do komputera osobistego i tablicę zapewniającą powierzchnię pracy dla myszy z oznaczonymi wierszami i kolumnami w alfabecie Braille'a dla lokalizacji położenia.

W przyszłości system będzie działał z syntetyzatorem głosu - wymawiając słowa napisane na ekranie.

The Times 1987.01.13

Intel opracowuje układy wyspecjalizowane

Wkrótce oczekiwane są pierwsze oferty firmy Intel Corp. w dziedzinie układów scalonych do określonych zastosowań (Application-Specific Integrated Circuits - ASIC), co było zapowiedziane od przeszło roku. Będą to prawdopodobnie dosyć standardowe układy, oparte na licznych wyrobach własnych. Mówi się również o zakupie licencji na pomoce projektowe od IBM.

Electronics nr 32/86

Brytyjska nagroda za najlepszy projekt 1987

Jedną z nagród w 1987 r. British Design Awards otrzymała firma Inmos za swój 32-bitowy mikroprocesor zwany transputerem, który można łączyć z sobą w niemal nieograniczonych ilościach zwiększając tym samym moc obliczeniową systemu. Układ 3-4 transputerów daje komputer osobisty o dużych możliwościach, zaś 16-tysięcy transputerów - superkomputer. Takie połączenie zapewnia szybką łączność, przy czym w przeciwieństwie do konwencjonalnych mikroprocesorów szybkość rośnie wraz z dodanymi procesorami. Na przykład stosując 100 transputerów można za około 100 tysięcy funtów uzyskać polioyjny system do badania odcisków palców, który w konwencjonalnym wykonaniu kosztowałby dziesięciokrotnie drożej.

The Times 1987.01.13

Prognozy na rok 1987 w dziedzinie półprzewodników

Znane ze swych optymistycznych prognoz Stowarzyszenie Przemysłu Półprzewodników (Semiconductor Industry Association - SIA) ogłosiło przewidywany wzrost światowej sprzedaży elementów półprzewodnikowych o 20% w 1987 roku. Wskaźnik ten uzyskano z danych dostarczonych przez amerykańskie, japońskie i europejskie firmy półprzewodnikowe.

Należy przypomnieć, że analogiczna prognoza dla roku 1986 z tego samego źródła mówiła o 26% wzroście, a aktualne dane każą zmniejszyć ten wskaźnik prawie czterokrotnie. Przewidywana wartość sprzedaży w 1986 roku ma wynosić 26,5 mld. dolarów, a według wspomianej prognozy w 1987 r. - 31,9 mln.

Electronics nr 32/86

Texas Instruments sprzedaje testery układów scalonych

Firma Texas Instruments Inc. zmieniła swą politykę dotyczącą utrzymywania w tajemnicy opracowanych przez siebie szczegółów testowania układów. Sprzedaje ona obecnie testery swym klientom, czyniąc z wytwarzania testerów duże przedsięwzięcie. Zapowiedziany jest nowy tester o nazwie Impact-1, tańszy od istniejących systemów, którego ceny na razie nie ujawniono, a który sprawdza parametry zmienna- i stałoprądowe przy jednorazowym załączeniu szybkiego układu scalonego. Sterowanie testera odbywa się z 32-bitowej jednostki centralnej.

Electronics nr 32/86

Szybki wzrost produkcji drukarek laserowych

Wszystkie większe firmy w przemyśle komputerowym, a także wytwarzające tylko drukarki i terminale drukujące oferują drukarki laserowe, które stały się istotnym i rozwijającym się działem przemysłu laserowego.

W roku 1985 wartość sprzedanych drukarek tego rodzaju oceniana jest na 1,6 mld dolarów, co stanowi ok. 30% obrotów przemysłu laserowego. Główną przyczyną tak szybkiego wzrostu jest rozwój automatyzacji biur i przewidywania na rok 1988 mówią o 4,4 mld dolarów.

Drukujące systemy laserowe wykorzystują lasery diodowe, helowo-neonowe, helowo-kadmowe i jonowe. Sprzedaż urządzeń laserowych, która osiągnęła 20 mln dolarów w 1984 r. wzrasta o 47% w r. 1985 i dalsze 35% w roku 1986, kiedy przekroczy 40 mln dolarów. Dla porównania cały handlowy rynek laserowy wzrasta o 28% w r. 1985.

Ze względu na obniżkę cen laserów stosowanych w drukarkach nieuderzeniowych, ilość sprzedanych drukarek rośnie szybciej aniżeli ich wartość. W 1985 r. o 54% do 97150. Do końca lat osiemdziesiątych najszybciej rozwijać się będą urządzenia na laserach diodowych. Są one stosowane w szybko rozwijających się osobistych drukarkach laserowych. Następnie idzie grupa urządzeń oparta na laserach He-Ne i na końcu drukarki na tzw. laserach niebieskich obejmujących lasery He-Cd i jonowe.

Automatyzacja biur spowodowała gwałtowny rozwój drukarek. Według danych Xerox Corp. w roku 1983 wydrukowano w USA $2,5 \times 10^{12}$ stron, a w r. 1989 wartość ta osiągnie 4×10^{12} , przy czym szybciej (w tempie ok. 40% rocznie) rośnie wydruk przez drukarki nieuderzeniowe. Drukarki laserowe mają szereg zalet w stosunku do tradycyjnych technik wydruku: ciszej pracują, są szybsze i bardziej elastyczne. Aktualnie drukowanie laserowe rozwija się najszybciej wśród technik nieuderzeniowych, gdzie występuje też nakładanie jonów, magnetografia i drukarki strumieniowe. Aby utrzymać jednak tę pozycję drukarki laserowe muszą poprawiać swe parametry, jak koszt, jakość wydruku, możliwość wydruku kolorowego.

W zależności od szybkości i ilości drukowanych materiałów opracowywane są i produkowane różne typy drukarek laserowych. Drukarki do komputerów osobistych wykorzystują zwykle lasery diodowe i mają małą prędkość wydruku (do 10 stron na minutę) oraz małe ilości drukowanych materiałów (do tysiąca kopii na miesiąc). Drukarki szybkie, dostarczające średnich ilości wydruków (ponad stronę na sekundę) wykorzystują zwykle lasery He-Ne, a szybkie i bardzo duże drukarki (ponad milion kopii na miesiąc) stosują lasery helowo-kadmowe i jonowe. Wyboru rodzaju lasera dokonuje się na zasadzie uzyskania najlepszych efektów przy minimalnych kosztach. Dlatego lasery tzw. niebieskie w cenie ok. 2 tysięcy dolarów opłacają się tylko przy największych drukarkach.

W 1985 roku najwięcej wyprodukowano drukarek małych z laserami diodowymi (54 tysiące), przy czym przyrost produkcji w stosunku do ubiegłego roku wyniósł 80%. Następną grupę (37,5 tysiąca) to drukarki o średniej prędkości na laserach helowo-neonowych, gdzie jednak było niższe tempo przyrostu (25%). Znacznie mniej liczną (4 tysiące) jest grupa szybkich drukarek o średniej wydajności, również na laserach He-Ne. Ze względu na późne pojawienie się tych drukarek na rynku tempo przyrostu było tu najwyższe i wynosiło 100%. Najmniej (1100) natomiast wyprodukowano największych drukarek na laserach niebieskich. Tutaj też było najniższe tempo rozwoju (22%). Takie drukarki bowiem powstały już w połowie lat siedemdziesiątych i były wykorzystywane w wielkich ośrodkach obliczeniowych. Dopiero w 1984 r. małe drukarki laserowe pojawiły się na skalę masową. Było to możliwe po wprowadzeniu przez firmę Canon w 1983 r. zespołu drukującego z prędkością poniżej 10 stron/min. Wówczas po raz pierwszy pojawiły się drukarki laserowe w cenie 3-20 tys. dolarów, które można było zastosować w komputerach osobistych. Zaczęły je wytwarzać takie firmy, jak Hewlett-Packard Co., Apple Computer Inc., Compugraphic Corp., Imagen Corp., Corona Data Systems, Quality Micro Systems i Concept Technologies.

Ten zespół drukujący firmy Canon wykorzystywał laser półprzewodnikowy, obracające się wielościanowe zwierciadło (skaner), optykę ogniskującą i wymienny pojemnik z bębniem i tonerem. Wykorzystując tani laser diodowy i masowo wytwarzany skaner firma obniżyła znacznie koszt zespołu, a wymienny pojemnik pozwolił uniknąć kosztownego serwisu. Jednocześnie firmy japońskie, które są głównymi dostawcami laserów diodowych, zwiększyły znacznie swą produkcję ze względu na rozwój płyt typu kompakt i drukarek laserowych, np. w firmie Sharp produkcja osiągnęła 400 tysięcy miesięcznie w końcu 1985 roku w stosunku do 200 tysięcy na początku tego roku. W roku 1984 zastosowano w drukarkach około 30 tysięcy laserów diodowych w stosunku do zaledwie kilkuset w r. 1983. Dla roku 1988 przewiduje się wykorzystanie 300 tysięcy takich laserów w drukarkach. Wpływy za lasery diodowe w drukarkach wzrosły z 2,7 mln dolarów w 1984 r. do 4,8 mln dolarów w 1985 r. a więc o 78%. Według przewidywań kolorowe drukarki laserowe powinny znacznie powiększyć zamówienia. Już obecnie (sierpień 1985) Battelle Columbus Laboratory pokazało drukarkę laserową z możliwościami wydruku kolorowego wykorzystującą trzy lasery diodowe dla trzech barw podstawowych.

Lasery diodowe zostaną zapewne wkrótce zastosowane do drukarek o średniej szybkości wydruku zastępując tam lasery He-Ne. Firma IBM ogłosiła, że zastępuje taki laser laserem diodowym z arsenku galu w swej drukarce o prędkości 20 stron na minutę. Zmniejsza to koszty i objętość, a także daje produkt bardziej niezawodny. Jednakże poprawa parametrów laserów He-Ne i wysoka jakość wydruku uzyskiwana z tych laserów powoduje, że liczba drukarek tego typu wzrosła w 1985 roku o ponad 30% w stosunku do roku ubiegłego do 42 tysięcy. Drukarki takie produkuje m. in. Xerox, Hewlett-Packard, Datagraphics, Fujitsu i Hitachi. Cena laserów He-Ne spadła o 8% w ciągu roku, tak że wartość sprzedanych drukarek wzrosła tylko o 21%. W latach 1988-90 przewiduje się produkcję drukarek-kopirek z laserami He-Ne rzędu 50-60 tysięcy rocznie.

Szybkich, dużych drukarek sprzedano w 1984 r. za 2,7 mln dolarów, a więc za tyle, co laserów diodowych w małych drukarkach. Podobne też było tempo wzrostu. Systemy kosztują tu rzędu 400 tysięcy dolarów, a więc koszt samego lasera stanowi około 0,5% tej wartości. Dominuje tu Xerox Corp., która w 1984 r. 30% swych 2 miliardowych wpływów uzyskiwała ze sprzedaży dużych drukarek laserowych.

Przewiduje się, że w przyszłości powstaną nowe grupy drukarek laserowych, związanych z jakością wydruku i wydrukiem kolorowym. Ceny, zwłaszcza najmniejszych drukarek, będą nadal spadać.

Na podstawie Laser Focus Nr 8/85

Laserowy sprzęt biurowy firmy Canon

Od roku 1975 udział sprzętu biurowego w asortymencie produkcji firmy Canon z Tokio wzrósł z 45 do 70%. W roku 1985 obroty przedsiębiorstwa osiągnęły prawie 5 mld dolarów, z czego 27-28% to dochody ze sprzedaży kamer.

Obecnie głównym wyrobem Canona są drukarki laserowe, a w przyszłości będą nimi systemy pamięci optycznej. Od roku 1973 do 1984 dochody netto wzrosły 20-krotnie, z czego 5-6% szło zawsze na badania rozwojowe. Obecnie na cele te wydaje się około 10%. W roku 1984 firma zgłosiła w USA 431 patentów (zajmując pod tym względem szóstą miejsce na świecie), a co pięć dni ogłaszano parametry nowego wyrobu. W roku 1975 opracowano pierwszą drukarkę laserową. Obecnie (październik 1986) Canon wytwarza 30 tysięcy drukarek z bębniem wykorzystującym OPC (organic photoconductor - fotoprzewodnik organiczny). W ciągu roku oczekiwane jest zwiększenie produkcji do 100.000 sztuk miesięcznie. Drukarka ta została opracowana w 1983 roku. Swój sukces rynkowy zawdzięcza częściowo temu, że stosowane tu są te same podzespoły co w popularnej kopiarce do komputerów osobistych, która produkowana jest przez firmę od 1984 roku.

Mimo istnienia około dwudziestu konkurentów w dziedzinie drukarek laserowych, Canon utrzymuje około 90% rynku, przy czym większość zamówień pochodzi z USA.

Inny rodzaj drukarki Canona posiada bęben z amorficznego krzemu. Drukarke taką opracowano już w 1979 roku. Może ona wykorzystywać podobnie jak OPC tanią diodę laserową o mocy 5 mW pracującą na długości fali 780 nm. Jednakże z jednego bębna można tu osiągnąć ponad milion kopii (tylko 3000 przy OPC) i ponad dwukrotne skrócenie czasu wydruku. Takie drukarki sterowane są w systemach rejestracji dokumentów, a drukarki OPC w szybkich układach faksymile (np. G4), jakie opracowano w 1985 roku.

Intensywnie opracowywane są podzespoły do pamięci optycznych, zarówno w zakresie środków przechowywania informacji, jak i głowicy odczytu. Prace badawcze w tym zakresie łączą odporność konstrukcji typu kompakt dysk z elektroniczną korektą błędów i precyzyjnym zapisem na ośrodkach wymazywalnych. Najbardziej zaawansowane są magnetyczno-optyczne systemy pamięciowe. Termomagnetyczna zasada zapisu pozwala na dużą odległość głowicy od warstwy aktywnej co z kolei umożliwia stosowanie grubych warstw ochronnych zabezpieczających przed zanieczyszczeniem dysku. Dzięki temu dyski optyczne mogą być wymieniane tak jak dyski typu kompakt. Wadą natomiast tych pamięci jest duża masa głowicy odczytu-zapisu, znacznie większa od współczesnych cienkowarstwowych głowic magnetycznych. Powoduje to wydłużenie czasu dostępu do ok. 100 ms, a więc pięciokrotnie więcej niż dla pamięci magnetycznych. Aby to poprawić rozważa się wprowadzenie elementów optyki zintegrowanej do projektu głowicy, które zastąpiłyby konwencjonalne pryzmaty i polaryzatory. Jedno z rozwiązań przewiduje zastosowanie falowodu z miedzią lub litu, w którym przykładane napięcie sterowałoby utrzymywanie lasera w ognisku. Wymaga to jednak dostosowania się do zmiany długości promieniowanego światła ze zmianą temperatury. Tańszym rozwiązaniem są cienkie elementy szklane z wytrawionymi siatkami, które zastępują pryzmaty i soczewki. Dierze się również pod uwagę elementy holograficzne, lecz mała wydajność nie rokuje im perspektyw.

Firma nie wytwarza laserów półprzewodnikowych, ale kupuje je od Hitachi i Mitsubishi i jest, poza wytwórcami adapterów płyt kompakt, głównym konsumentem tych laserów. Podejmowane są wspólne prace, w wyniku których powinno uzyskać się systemy z laserami półprzewodnikowymi o średnim okresie pracy rzędu 20 tysięcy godzin.

Laser Focus nr 10/86

Lotus Development na rynku japońskim

Japonia jest krajem robotów, programowo sterowanych poślągów o kształtach aerodynamicznych, lecz komputeryzacja zarządzania i biur robi tu małe postępy. Japoński przemysł komputerowy niewiele zrobił w tym zakresie. Naśladuje on większość sprzętu amerykańskiego, lecz w dziedzinie oprogramowania pozostał w tyle.

Pojawienie się japońskiej wersji jednego z najbardziej znanych programów, typu arkuszy obrachunkowych (spreadsheet), które były dotychczas mało znane w Japonii, może zmienić tę sytuację.

Po wielu latach przygotowań, nowe przedstawicielstwo firmy Lotus Development Corp. z Cambridge w stanie Massachusetts, ogłosiło w październiku 1986 r. specjalną wersję swego programu planowania finansowego Lotus 1-2-3, która nie tylko pisana jest po japońsku, ale kompiluje dane w formatach odpowiadających zwyczajom tego kraju. Np. naciśnięcie klawiszy 1-9-8-6 powoduje pojawienie się rysunku w stylu japońskim i napisu "Showa 61", to znaczy 61 rok panowania cesarza Hirohito.

Firma Lotus nie jest pierwszą, która próbowała dostać się na rynek japoński. Jej największy rywal Microsoft Corp. od czterech lat stara się rozpowszechnić w Japonii swój pakiet Multiplan, lecz dotychczas sprzedała tylko 100 tysięcy egzemplarzy. Dlatego wszyscy z wielkim zainteresowaniem oczekują na wyniki akcji firmy Lotus, które dotychczas sprzedała już w USA i innych krajach ponad 2 miliony swych pakietów oprogramowania. Nawet sami przedstawiciele firmy nie są wielkimi optymistami w tym zakresie. Menadżerzy japońscy rzadko posługują się bezpośrednio komputerami. Jednakże w ostatnich latach rozpowszechniły się systemy przetwarzania tekstów dzięki usprawnieniom w programach zwanych "konwerterami kana-kenji", które pozwalają na posługiwanie się tylko około 50- przyciskową klawiaturą fonetycznego alfabetu kana, podczas gdy w kanji jest 7 tysięcy znaków.

Dlatego wiele takich systemów wykorzystywanych jest jako maszyny do pisania. Japończycy przyzwyczaili się pisać ręcznie i dlatego jest tam tak dużo urządzeń faksymile. Nawet określenie "automatyzacja biur" ma w Japonii nieco inne znaczenie. Przypomina automatyzację fabryk. Zwracają oni uwagę na poprawność przekazywania informacji, natomiast nie widać wysiłku na poprawienie wydajności za pomocą skomputeryzowanych narzędzi.

Lotus próbuje to zmienić i reklamy pokazuja samurajów walczących mieczami, a wojownik w środku ma w ręku dysk elastyczny. W ciągu pierwszych dwóch miesięcy kampanii sprzedano 6 tysięcy egzemplarzy pakietu w cenie około 500 dolarów, co nie jest złym początkiem.

Japoński arkusz znacznie różni się od oryginału. Kolumny i wiersze rozdzielane są siatką linii ciągłych i kreskowanych. Słowa sortowane są fonetycznie, a przedstawienie graficzne obejmuje japońskie grafy spiralne, gdzie dane mogą być kreślone wzdłuż trzech osi, przy czym jakość obróbki czy wydruków jest lepsza niż w USA - urządzenia ich mają wyższą rozdzielczość.

Ta przeróbka programu zajęła prawie 2 lata - więcej niż napisanie oryginału. Obecnie działa on na komputerze osobistym firmy NEC i biurowym komputerze IBM model 5550. /

International Herald Tribune 1986.12.09

Osiągnięcia japońskie

Japońscy wytwórcy układów półprzewodnikowych wyprzedzili swych konkurentów amerykańskich i jeśli zastosują te elementy we własnych urządzeniach będzie to stanowiło znaczne zwiększenie konkurencyjności japońskich komputerów, które dotychczas nie były zbyt powszechne na rynkach światowych.

Jest to również problem bezpieczeństwa, gdyż do celów obronnych potrzebne są najbardziej złożone układy, w dziedzinie których japończycy dokonali największego postępu. Niedawno w prasie japońskiej pojawiła się informacja o opracowaniu przez firmę Nippon Telegraph and Telephone kostki pamięci dynamicznej o pojemności 16 Mbitów. Będzie ona zawierać 35 milionów tranzystorów. Oficjalne podanie parametrów tego podzespołu oczekiwane jest w ciągu miesiąca. Rynek kostek pamięciowych o pojemności 1 Mbita należy prawie w 100% do wytwórców japońskich, a dwie firmy Hitachi i Toshiba przewidują uruchomienie produkcji tych elementów w USA w 1987 r. Przewidują one 10-krotny wzrost zapotrzebowania - z 5 mln sztuk w r. 1986 do 50 mln w roku bieżącym. Kostki te ma zamiar stosować w swych komputerach IBM.

Aby zabezpieczyć zmniejszające się zyski, szereg amerykańskich producentów półprzewodników podjęło w ostatnich miesiącach współpracę z firmami japońskimi.

The Times 1987.01.13

Nowe wielkie przedsiębiorstwo telekomunikacyjne

W grudniu 1986 r. francuska firma państwowa Compagnie Generale d' Electricité i ITT utworzyły przedsiębiorstwo telekomunikacyjne o nazwie Alcatel będące drugim w świecie wytwórcą urządzeń telekomunikacyjnych po AT&T. Oczekiwana w 1987 r. wartość sprzedanych wyrobów ma wynosić 9 mld funtów, a zysk 180 mln.

Celem utworzenia Alcatelu było stworzenie czołowego europejskiego koncernu łączności, zdolnego konkurować na rynkach światowych.

Polowa zarobków przejdzie do ITT, które wg umowy ma otrzymywać 625 mln funtów w gotówce za swe zakłady przekazane nowej firmie. Ponadto 240 mln funtów ma być przekazane na spłatę wewnętrznych długów ITT.

The Times 1987.01.13

Sprzedaż komputerowego działu Honeywella firmom francuskim i japońskim

Ze względu na straty, jakie ponosił ostatnio dział komputerowy firmy Honeywell Inc., zostanie on przejęty przez nowe przedsiębiorstwo o nieustalonej jeszcze nazwie, w którym udziały po 42,5% będą miały Honeywell i francuska firma Bull, a 15% japońska NEC. Honeywell otrzyma 500 mln dolarów, z czego połowa pójdzie na pokrycie strat. Udział Honeywella ma spaść do 19,9% do końca 1988 r. Wartość sprzedaży nowego przedsiębiorstwa jest oceniana na 1,85 mld dolarów rocznie. Będzie to pierwsza amerykańsko-francusko-japońska spółka komputerowa. Ostateczne zakończenie operacji przewidziane jest na koniec marca 1987 r. przy czym głównym partnerem będzie Bull. Honeywell przewiduje, że będzie największym klientem nowej firmy, jako dostawca systemów komputerowych.

Krok ten zmusza do zmiany struktury firmy Honeywell, która zmniejsza zatrudnienie o 4 tysiące pracowników a jednocześnie zakupiła udział kosmonautyczny Sperry od Unisys Corp. za ponad miliard dolarów.

Nowe przedsiębiorstwo przejmie produkcję dużych maszyn, mini- i mikrokomputerów wraz z oprogramowaniem i obsługą w USA i zagranicą.

Ponadto sprzedawać będzie wyroby firm Bull i NEC. Poprzednio Honeywell posiadał akcje firmy Bull, do jej nacjonalizacji w 1982 roku. Pomagał też w początku lat siedemdziesiątych firmie NEC, która później rozwinęła się szeroko w dziedzinie wielkich systemów informatycznych.

International Herald Tribune z 1986.12.03

Spadek zysków firmy Toshiba Corp.

Według oczekiwań w roku obliczeniowym kończącym się 31 marca 1987 roku czysty zysk firmy Toshiba Corp. spadnie o 46% w stosunku do roku ubiegłego osiągając wartość 200 mln dolarów. Jedną z przyczyn tego stanu rzeczy jest silna ryzyka jena, osłabiająca opłacalność eksportu.

Za półroczcie, skończone 30 września 1986 r., spadek ten wyniósł aż 54%. Sprzedaż ciężkiego sprzętu elektrycznego spadła o 24%, wyrobów elektroniki użytkowej o 10%, a elektroniki przemysłowej i elementów elektronicznych wzrosła tylko o 1%. Eksport spadł we wszystkich dziedzinach za wyjątkiem elektroniki.

Jednym ze środków zaradczych jest przenoszenie produkcji za granicę. W marcu 1987 r. rozpoczynają produkcję zakłady telefoniczne i sprzętu medycznego w Kalifornii, oraz kamer telewizyjnych w RFN. Przewidziana jest też umowa z Motorola Inc. na wspólne przedsięwzięcia. Prognozy analityków rynku nie są jednak korzystne dla firmy.

International Herald Tribune z 1986.12.03

IBM pozbywa się przedsięwzięć przynoszących straty

Na początku 1987 roku IBM ogłosiła zamknięcie International Marketnet - zbiorowej inicjatywy, której celem był marketing wyspecjalizowanych systemów komputerowych, opartych na skomplikowanej sieci satelitarnej dla maklerów giełdowych i innych inwestorów.

Kilka dni później podano wiadomość o likwidacji istniejącego od sześciu lat oddziału przyrządów IBM, który sprzedawał narzędzia do analizy chemicznej laboratoriom.

Ponadto firma obniżyła o 15% ceny swych głównych modeli komputerów osobistych w USA, co traktowane jest jako wstęp do nowej generacji tych komputerów. Oczekuje się, że w najbliższych miesiącach ogłoszone zostaną parametry tej generacji. Ceny hurtowe zostały obniżone w grudniu.

Mimo tych kroków jest pewne, że wyniki ostatniego kwartału 1986 roku przyniosą kolejne obniżenie zysków firmy.

Według ostatnich doniesień wstrzymana zostanie produkcja podstawowego IBM PC, a więc PC-XT jest najprostszym produkowanym mikrokomputerem z tej serii.

Wysuwane są różne przypuszczenia dotyczące nowych modeli. Mówi się o takim urządzeniu, przede wszystkim dla potrzeb oświatowych, ale sama firma stwierdza, że nie jest zainteresowana małymi systemami ze względu na konkurencyjne ceny. Dlatego większość prognoz mówi o większych komputerach, przekraczających parametry PC AT.

The Times 1987.01.13

Utworzenie działu zastosowań militarnych DEC w W. Brytanii

Firma Digital Equipment od ponad 20 lat zaopatruje brytyjskie Ministerstwo Obrony w odpowiedni sprzęt. Jej rodzina komputerów VAX stała się faktycznie standardem wojskowym. Dla realizacji nowych zamówień ocenianych na ponad 3 mln funtów powołano specjalny oddział zajmujący się dowodzeniem, sterowaniem i łącznością, a także systemami ekspertowymi, które powinny przyspieszyć podejmowanie decyzji.

Ogólna wartość obrotu sprzętem komputerowym w tych dziedzinach wynosi 200 mln funtów, z czego w zastosowaniach wojskowych 45 mln. DEC ma nadzieję zwiększyć ten udział.

Przedsiębiorstwo zatrudnia około 6 tysięcy osób w 30 różnych jednostkach w Wielkiej Brytanii. W ciągu 1987 r. przewiduje się zwiększenie personelu o 1400 pracowników, z czego 160 w ośrodku systemów rządowych i obronnych w Basingstoke.

Macierzysta firma DEC w USA ujawniła również szereg kontraktów rządowych obejmujących m.in. instalację komputerów VAX w W. Brytanii. Obejmuje to modernizację światowej cyfrowej sieci automatycznej (AUTODIN) Departamentu Obrony USA za 75 mln dolarów, w ramach której zainstalowane zostaną nowe komputery w Croughton, oraz kontrakt na wyposażenie Marynarki USA w system obejmujący 125 komputerów VAX.

The Times 1987.01.13

Informatycy poszukiwani

W okresie gdy bezrobocie w Wielkiej Brytanii przekracza rozmiary kryzysu lat trzydziestych, nie omijając nawet przemysłu elektronicznego, specjaliści komputerowi są bardzo poszukiwani, zwłaszcza na południu kraju i nawet perspektywa wysokich zarobków nie rozwiązuje tego problemu. Dzieje się tak dlatego, że począwszy od 1979 roku dwie trzecie nowo utworzonych miejsc pracy w tej dziedzinie powstała w południowo-wschodniej części W. Brytanii. Wykwalifikowanym sprzedawcom komputerów w Londynie oferuje się 30 tysięcy funtów rocznie plus dodatkowe korzyści, a mimo to nie wszystkie miejsca są obsadzone. Uposażenia tej grupy pracowników rosną szybciej niż w innych działach gospodarki. Od października do grudnia 1986 roku wzrosły one o 11,9%, co jest nieco niższym wskaźnikiem aniżeli w roku poprzednim (12,6%).

Regionalne braki personelu wynikają też z niechęci pracowników do zmiany miejsca zatrudnienia i zamieszkania, o ile mają stabilną pracę nawet na nieco gorszych warunkach. Ankiety przeprowadzone wśród firm usługowych wykazują, że brak specjalistów jest głównym czynnikiem ograniczającym zastosowanie mikrokomputerów w przemyśle południowo - wschodniej Anglii.

The Times 1987.01.13

Magnetofony cyfrowe

W drugiej połowie lutego 1987 r. japońskie firmy Aiwa, Matsushita, Sony i Sharp zademonstrowały swe pierwsze magnetofony cyfrowe (DAT - Data Audio Tapeplayer), które w ciągu następných tygodni miały być sprzedawane w Japonii i jeszcze w tym roku miały dotrzeć do USA i Europy Zachodniej. Inne firmy jak JVC, Hitachi, Toshiba wkrótce również wejdą na ten rynek.

Jest to jak gdyby nowy krok w tej dziedzinie po dyskach typu kompakt. Tu również stosowana jest cyfrowa technika zapisu, która wykorzystuje komputery do zamiany dźwięku na miliardy bitów informacji przed umieszczeniem ich na taśmie magnetycznej. O ile jednak adaptory typu kompakt odtwarzają tylko muzykę, urządzenia taśmy cyfrowej mogą być wykorzystane również do zapisu i kopiowania.

Początkowo ceny zarówno kaset jak i urządzeń są dosyć wysokie i tylko prawdziwi wielbiciele dobrej muzyki będą mogli sobie na nie pozwolić, ale wkrótce napewno będą one spadać. Koszt kasety, której wymiary odpowiadają mniej więcej dwóm trzecim wymiarów zwykłej kasety analogowej, wynosi obecnie około 12 dolarów, a pierwszy magnetofon cyfrowy firmy Matsushita kosztować będzie 1285 dolarów /adaptory typu kompakt kosztują 200-600 dolarów/. Produkcję rozpoczyna się od 2 tysięcy urządzeń miesięcznie, by osiągnąć 50 tysięcy w styczniu 1988 roku. Produkcję w roku 1990 ocenia się na 3 miliony sztuk, choć firma Aiwa mówi o dwukrotnie większej liczbie uwzględniając stereofoniczne magnetofony samochodowe i magnetofony z odbiornikami radiowymi.

Oczywiście dominującym czynnikiem jest tu jakość odtwarzanej muzyki, wolnej od zakłóceń typowych dla zwykłych magnetofonów. W celu uniemożliwienia nieuprawnionego przegrywania wytwórcy japońscy zainstalowali specjalne układy zabezpieczające przed bezpośrednim cyfrowym kopiowaniem taśm i płyt odpowiednio kodowanych. Można natomiast przegrywać w postaci analogowej.

Standardy na urządzenie i taśmy ustalone były już w czerwcu 1985 r. Przemysł potrzebował jednak ponad półtora roku do pełnego opracowania technologii. Przedstawiciel firmy Sony w USA powiedział, że codziennie otrzymują oni setki zapytań o te produkty.

Time z 1987.03.02

Mikroinżynieria

Wydaje się to nieprawdopodobne, ale fachowcy twierdzą, że jesteśmy o krok od uzyskania superminiaturowych silników elektrycznych niemal niedostrzegalnych gołym okiem. Wytwarzane byłyby one tak jak elementy mikroelektryczne w krysztale krzemu z wykorzystaniem tych samych technologii wytrawiania, zmniejszania fotograficznego i rozpuszczenia podkładu. Powstają wówczas ciągi mikromechanizmów obracających się wokół ustalonych punktów. Przy tych wymiarach powietrze działa łożyskująco i nie wymagane jest smarowanie.

Łącząc je z elektronicznym sterowaniem można osiągnąć efekty przekraczające wyobrażenie. Niektórzy badacze przewidują prędkość obrotową rzędu miliona obrotów na minutę i dodają, że Bell Laboratories w USA ma na ukończeniu prototyp takiego silnika o średnicy mniejszej od jednej dziesiątej milimetra.

Utrzymanie w tajemnicy tych prac związane jest z ich wojskowymi zastosowaniami. Mówi się np. o sztucznych owadach, które umieszczone w terenie czekałyby na nadejście nieprzyjaciela, by potem zdalnie sterowane zaatakować go. Wiadomo też, że stabilność żyroskopu zależy od jego prędkości obrotowej, a więc idealne zastosowanie dla rakiet przeciwczołgowych.

Również inne dziedziny jak np. mikrochirurgia są tu cennym zastosowaniem. Opracowano koncepcyjnie nożyce o długości ostrzy 1 mm, które miałyby czujniki pozwalające chirurgowi odczuwać proces cięcia. Oddałoby to duże usługi np. przy nakładaniu nerwu ocznego przy transplantacji oka.

Praktyczne wykorzystanie zaś sztucznych owadów mogłoby być przez zatrudnienie ich do mycia szyb lub wyjadania brudu z zakamarków mieszkania.

The Times 1987.01.13

Projektowanie jachtów za pomocą superkomputerów

Po utracie tytułu mistrzowskiego w "America's Cup" w 1983 r. USA wyasygnowały 60 mln dolarów na poprawę konkurencyjności jachtów amerykańskich. Do projektowania nowych jednostek zostały zaangażowane takie instytucje jak NASA, Boeing, MIT, Cray Research. Do symulacji działania wiatru i fal użyto potężnych superkomputerów, jak Cray XMP-48 i specjalistów, jak H. Meldner, który projektuje broń nuklearną.

Projektowane jachty odbiegają od klasycznych, np. jacht nazwany USA z klubu w San Francisco ma balast w kształcie torpedy osadzony na wąskiej podporze w tyle kadłuba. Wyposażony jest również w dwastery, jeden jak zwykle na rufie, a drugi blisko dziobu. Użyte jednocześnie pozwalają na bardzo sprawne zwroty, co jest ważne przy ściganiu się.

Cykl obejmuje projektowanie kadłuba, analizowanie go na komputerze, dokonanie zmian, ponowne sfinalizowanie i tak wielokrotnie aż projekt wydaje się zadawalający. Następnie projektanci sprawdzają swą koncepcję w specjalnym zbiorniku. W jednym przypadku sprawdzono czterdzieści różnych kombinacji kadłuba, balastu i skrzydeł do niego przytwierdzonych, wszystko na modelach w skali 1:10. Modele te holuje się w zbiorniku, w którym specjalne urządzenie wytwarza fale symulujące warunki wyścigu.

The Observer z 86.11.30

Komputeryzacja departamentu paszportów

Brytyjska firma Software Sciences należąca do koncernu Thorn-EMI wygrała przetarg na komputeryzację departamentu paszportów Home Office (brytyjskiego ministerstwa spraw wewnętrznych) o wartości 10 milionów funtów szterlingów. Departament ten wydaje 2 miliony paszportów rocznie.

Zgodnie z wcześniejszymi uzgodnieniami w krajach Wspólnoty Europejskiej w ciągu najbliższych dwóch lat mają być wprowadzone paszporty, które mogą być kontrolowane maszynowo. Omawiany system m.in. stosowany będzie do sterowania wytwarzaniem tych paszportów.

Przetarg wzbudził zainteresowanie, gdyż brała w nim udział wyspecjalizowana firma amerykańska Electronic Data Systems.

The Times 1987.01.13

Pokaz elektroniki użytkowej w Las Vegas

W sprawozdaniu z tego pokazu po raz pierwszy od czterech lat nie mówiło się na pierwszym miejscu o mikrokomputerach ale o urządzeniach do odtwarzania dysków optycznych lub cyfrowej taśmy dźwiękowej.

Nie zabrakło oczywiście i mikrokomputerów. Po raz pierwszy pokazano w USA Amstrada 1512, którego wyceniono na 800 dolarów. Pojawiły się też kompatybilne z IBM PC komputery osobiste, firm Apple i Atari. W pierwszym przypadku szereg innych firm oferowało dodatkowe pakiety, które wetknięte do Apple IIo lub IIGS pozwalają na prace tych komputerów z większością oprogramowania pisanego na IBM PC.

Atari natomiast oferuje nowe wersje swej wersji serii ST o pojemności pamięci operacyjnej od 1 do 4 Mbajtów. W planach jest rozmieszczenie pamięci do 16 Mbajtów, opracowanie szybkiego dysku sztywnego i sieci tych komputerów, a także drukarki laserowej, która ma kosztować nieco ponad tysiąc dolarów (pełny biurkowy system wydawniczy - 2 tysiące).

Na skutek ogłoszenia tych nowych wyrobów ceny komputerów Atarii zostaną od lutego 1987 obniżone o ok. 20%.

The Times 1987.01.13

Wystawa robotów w Bostonie

W lutym 1987 roku otwarto w Bostońskim Muzeum Nauki wystawę "Roboty i co się za nimi kryje", która w ciągu najbliższych trzech lat odwiedzi siedem innych miast USA. Koszt wystawianych eksponatów wynosi 4 miliony dolarów, z czego połowa pochodzi z firmy Digital Equipment Corporation z Maynard w Massachusetts.

W dwóch obszernych pomieszczeniach znajduje się kilkadziesiąt inteligentnych maszyn reprezentujących wszystkie dziedziny tej najnowszej gałęzi techniki. Już przy wejściu wita nas robot o wysmukłych palcach noszący nazwę Jorel, który metalicznym głosem powtarza zdanie: "Ludzie, jesteście świadkami rozpoczęcia nowej, wielkiej ery".

Zwiedzający mogą zagrać z komputerowym oszustem karcianym lub z klawiatury sterować robotami-zabawkami w fabryce zbudowanej z klocków Lego. Mogą też przyglądać się jak mechaniczne ramiona układają stos płytek krzemowych lub przelewają chemikalia z jednej próbówki badawczej do drugiej. Obserwując cierpliwie roboty zamknięte w pomieszczeniu można usłyszeć ich okrzyki bólu przy każdym uderzeniu o ścianę.

Obserwujemy np. półprzezroczystego robota okrytego sztuczną skórą z warstwy piezoelektrycznej. Lekkie dotknięcie go powoduje okrzyk "Czuję to!", a gwałtowne uderzenie "Przestań, ty dzieciolo!" Inny robot maluje obrazy wg programu Aaron, opracowywanego od 13 lat przez H. Cohena, profesora na uniwersytecie w San Diego.

Jeszcze inny rodzaj robotów to te, które widzą. Kamery sprzężone są z komputerami, z których niektóre reagują na obiekty poruszające się, inne na nie zmieniające położenia. Roboty przemysłowe tego rodzaju identyfikują pewne przedmioty, jak grzebień, klucze czy monety.

Najbardziej złożone są systemy ekspertowe, emulujące rozumowanie i postępowanie ekspertów w określonej dziedzinie. Mogą np. doradzać postępowanie w jakiejś grze lub oceniać je.

Time z 87.02.12

Postęp w dziedzinie nadprzewodnictwa

Z materiałami nadprzewodzącymi od dawna wiązano nadzieję na opracowanie ultraszybkich komputerów, układów przetwarzania sygnałów i zapisu danych, jednakże problemy związane z koniecznością utrzymywania bardzo niskich temperatur opóźniały praktyczne ich zastosowanie.

Ostatnio pojawiło się kilka doniesień o znalezieniu materiałów nie wymagających ekstremalnie niskich temperatur co może znacznie zaawansować te badania.

W kwietniu 1986 r. laboratorium IBM w Zurychu doniosło o osiągnięciu zjawiska nadprzewodnictwa w temperaturze 30° Kelwina -243°C w mieszaninie baru, lantanu, miedzi i tlenu. Później odkrycie to potwierdził Uniwersytet w Tokio.

Ostatnio natomiast naukowcy z Instytutu Fizyki Chińskiej Akademii Nauk w Pekinie podali, że uzyskali to zjawisko w materiale o tym samym składzie, ale przy temperaturze 70 stopni Kelwina, a więc blisko tej, jaką osiąga się za pomocą ciekłego azotu.

Dr R. Care z firmy AT&T, który w końcu 1986 roku ze swoim zespołem uzyskał podobny materiał, oświadczył, iż jeśli będzie można powtórzyć eksperyment z Pekinu będzie to oznaczać przełom w dziedzinie zastosowań nadprzewodników.

The Times 1987.01.13

Transoceaniczne kable światłowodowe

Amerykańska firma AT&T Co. rozpoczęła w stanie New Jersey łączenie początkowych odcinków pierwszego transatlantyckiego podwodnego kabla na światłowodach. Ma on być ukończony w 1988 roku za cenę 335 mln dolarów i będzie miał długość ok. 6650 km z Tuckerton w stanie New Jersey do Tidemouth w Wielkiej Brytanii i Penmarch we Francji. Ten system kabla światłowodowego nazwany TAT8 będzie mógł przenosić 40 tysięcy rozmów telefonicznych jednocześnie, ponad dwukrotnie więcej od obecnie dostępnych systemów podwodnych. Dla porównania najnowszy podmorski kabel miedziany TAT7, zbudowany w 1983 r. może przenosić tylko około 9 tysięcy rozmów.

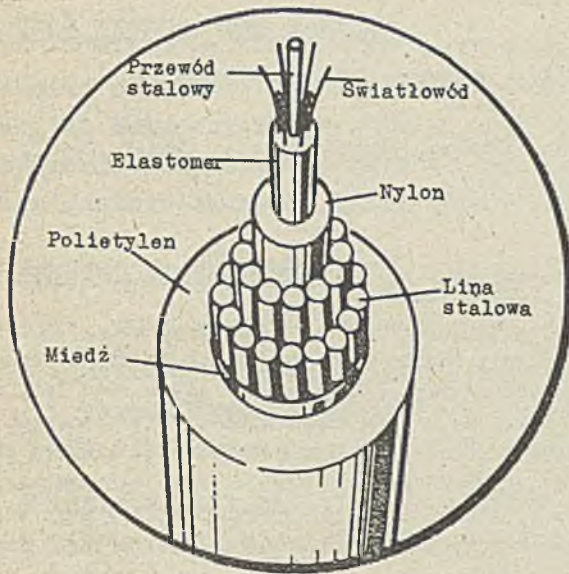
Podstawowym elementem nowego systemu są same światłowody, będące miniaturowymi żyłkami szklanymi mogącymi przenosić głos i informację komputerową za pomocą strumieni światła laserowego. Ponieważ wiązki tego światła mogą być bardzo skupione, światłowody mogą przenosić więcej informacji aniżeli konwencjonalne przewody miedziane.

TAT8 pozwoli przedsiębiorstwom telefonicznym rozszerzyć ich usługi transatlantyckie i oferuje szybkie kanały danych, poprzez które klienci będą mogli przesyłać ogromne ilości informacji z większym zabezpieczeniem aniżeli to jest możliwe przy transmisjach satelitarnych.

Kabel ten finansowało 29 koncernów telefonicznych z Europy i USA, wśród których są AT&T, ITT World Communications Inc, RCA Global Communications, MCI International Inc i Western Union Telegraph Co. Trzy firmy: AT&T, Standard Telephones & Cables PLC z Wielkiej Brytanii i CIT Alcatel-Les Cables de Lyon SA z Francji budują ten kabel.

Do AT&T należy największa część (37% przedsięwzięcia). Zainstaluje ona pierwsze 5850 km przez Atlantyk do wzmacniacza rozdzielającego. Kabel od tego wzmacniacza do punktu końcowego w Tidemouth instalować będzie przedsiębiorstwo brytyjskie. Konsorcjum francuskie natomiast zapewni 307 km kabla od wzmacniacza do Penmarch.

Pierwszy miedziany telefoniczny kabel TAT1 położono między Nową Fundlandią a Szkocją 30 lat temu. Zapewnił on połączenie dla 36 rozmów. Poprzednio rozmowy telefonowane były przekazywane przez Atlantyk drogą radiową.



Przekrój kabla

Kabel światłowodowy różni się znacząco od konwencjonalnego kabla miedzianego. Jest on znacznie cieńszy, mniej elastyczny i bardziej delikatny, wymagający specjalnej techniki łączenia i manipulowania. Po zakończeniu lądowej części kabla, statek kanadyjski położył odcinek do końca szelfu kontynentalnego o długości około 160 km. Ten odcinek jest szczególnie zabezpieczony w stalowe i nylonowe warstwy ochraniające przed ostrymi skałami, kotwicznymi i statkami rybackimi. Jako dodatkowy środek zapobiegawczy, część kabla leżąca na szelfie kontynentalnym do głębokości ok. 1000 m będzie zakopana przy pomocy robota-ługacza morskiego. Ługacz ten wykopie rów o głębokości ok. 60 cm, do którego zostanie położony kabel, a dyuze wodne z tyłu ługacza zasypywać będą rów gruntem dennym.

Robot używany do zakopywania kabla nazywa się Soars (Submersible Craft Assisting Repair and Burial - zanurzalny statek pomocniczy przy naprawach i zakopywaniu), posiada dwa długie ramiona, którymi można manewrować ze statku. Przy pracy tych urządzeń pracują też zwykle nurkowie.

Jesienią 1987 roku statek kablówkowy firmy AT&T o nazwie Long Lines wydobędzie koniec kabla na skraju szelfu kontynentalnego i połączy go z kablem znajdującym się na pokładzie statku, a następnie będzie kładł kabel przez Atlantyk. Część głębinowa kabla nie będzie miała dodatkowych osłon i będzie leżała na dnie oceanu.

Przed położeniem kabla przestudiowana będzie szczególnie mapa dna oceanicznego, aby wybrać najbardziej płaską trasę unikając ostrych szczytów i obszarów wulkanicznych, gdzie kabel mógłby ulec zniszczeniu w czasie wybuchu.

Największym jednak niebezpieczeństwem mogą być mikroskopijne zanieczyszczenia w samym kablu. Dlatego segmenty światłowodowe o średnicy ok. 0,13 mm będą stapiane z sobą przy pomocy płomienia w specjalnej komorze hermetycznej ("clean room") znajdującej się na statku.

Zwyczajna prędkość kładzenia kabla wynosi ponad 9 km/godz. Z tą prędkością potrzeba około 30 dni na przeprowadzenie się przez Atlantyk. Ponieważ kabel światłowodowy jest dwukrotnie mniejszy od kabla miedzianego, statek może zebrać jednorazowo cały ładunek. Przy kładzeniu kabla koncentrycznego Long Lines musi być czterokrotnie załadowany.

Sam kabel składa się z sześciu światłowodów tworzących trzy skrętki owinięte dookoła rdzenia stalowego otoczonych warstwą z plastiku, która z kolei otoczona jest grubymi drutami. Na zewnątrz znajduje się warstwa miedziana, przewodząca prąd zasilający regeneratory - urządzenia przetwarzające sygnały świetlne. Warstwa miedziana jest wreszcie izolowana zewnętrzną warstwą plastikową.

Dwie z par światłowodów będą wykorzystywane, z których każda przenosi sygnały w jednym kierunku. Trzecia skrętka stanowi rezerwę.

Regeneratory, umieszczone są w ok. 57 km, reprodukuja impulsy światła laserowego, co zabezpiecza informację wzdłuż całej trasy.

Przedsiębiorstwa realizujące TAT-8 podpisały już umowę na budowę drugiego transatlantyckiego kabla światłowodowego o nazwie TAT-9 o dwukrotnie większych możliwościach, który kosztować będzie 400 mln. dolarów.

Zgodnie z harmonogramem tego projektu ma on być ukończony w 1991 roku.

Istnieją również plany budowy światłowodowego kabla przez Pacyfik w 1989 roku. Koszt tego przedsięwzięcia szacuje się na ponad 700 mln. dolarów. Zapewni on podstawowe połączenia telefoniczne i informatyczne z Japonią i Filipinami.

C. Jeffcoat, główny inżynier do spraw kablowych firmy AT&T, powiedział, że wkrótce informatyk w Tokio będzie miał połączenie światłowodowe z systemem cyfrowym w Londynie.

International Herald Tribune, 1986.12.06-07.

Komputer przy wykrywaniu mordercy dzieci

Ministerstwo Spraw Wewnętrznych Wielkiej Brytanii nabyło duży komputer o nazwie Holmes. Chciano wykorzystać go do analizy wypadków zaginięcia trzech dziewczynek w wieku 5-11 lat, co wydarzyło się w latach 1982-86. Istnieje bowiem hipoteza, że we wszystkich tych przypadkach działał ten sam morderca. Wstępna analiza wykazuje istotne podobieństwa. Jednakże okazało się, że zebrane dotychczas materiały są w nieodpowiednich formatach do przesłania na ten komputer, a ręczne indeksowanie około 70 tysięcy zapisów dla jednego tylko przypadku zajęłoby zbyt wiele czasu. Dlatego Ministerstwo wystąpiło o dodatkowe fundusze dla standaryzacji policyjnych informacji komputerowych.

The Times 1986.12.12

Opracował: mgr inż. Jan RYŻKO

Informacja o cenach i warunkach prenumeraty na 1987 r.
- dla czasopism Instytutu Maszyn Matematycznych

● Cena prenumeraty rocznej

Techniki Komputerowe - Biuletyn Informacyjny	2280.- dwum.
Przegląd Dokumentacyjny - Nauki i Techniki Komputerowe	1860.- dwum.
Informacja Ekspresowa - Nauki i Techniki Komputerowe	4200.- mies.
Prace naukowo-badawcze Instytutu Maszyn Matematycznych	1800.- 3x w roku

● Warunki prenumeraty

1/ dla osób prawnych - instytucji i zakładów pracy:

- instytucje i zakłady pracy zlokalizowane w miastach wojewódzkich i pozostałych miastach, w których znajdują się siedziby oddziałów RSW "Prasa-Książka-Ruch" zamawiają prenumeratę w tych oddziałach;
- instytucje i zakłady pracy zlokalizowane w miejscowościach, gdzie nie ma oddziałów RSW "Prasa-Książka-Ruch" i na terenach wiejskich opłacają prenumeratę w urzędach pocztowych i u doręczycieli;

2/ dla osób fizycznych - prenumeratorów indywidualnych:

- osoby fizyczne zamieszkałe na wsi i w miejscowościach, gdzie nie ma oddziałów RSW "Prasa-Książka-Ruch" opłacają prenumeratę w urzędach pocztowych i u doręczycieli;
- osoby fizyczne zamieszkałe w miastach - siedzibach oddziałów RSW "Prasa-Książka-Ruch" opłacają prenumeratę wyłącznie w urzędach pocztowych nadawczo-oddawczych właściwych dla miejsca zamieszkania prenumeratora. Wpłaty dokonują używając "blankietu wpłaty" na rachunek bankowy miejscowego oddziału RSW "Prasa-Książka-Ruch";

3/ Prenumeratę ze zleceniem wysyłki za granicę przyjmuje RSW "Prasa-Książka-Ruch", Centrala Kolportażu Prasy i Wydawnictw, ul. Towarowa 28, 00-958 Warszawa, konto NBP XV Oddział w Warszawie nr 1153-201045-139-11. Prenumerata ze zleceniem wysyłki za granicę pocztą zwykłą jest droższa od prenumeraty krajowej o 50% dla zleciodawców indywidualnych i o 100% dla zlecających instytucji i zakładów pracy.

● Terminy przyjmowania prenumeraty na kraj i za granicę:

- do dnia 10 listopada na I kwartał, I półrozie roku następnego oraz na cały rok następny,
- do dnia 1 - każdego miesiąca poprzedzającego okres prenumeraty roku bieżącego.

Zamówienia na prenumeratę "Prac naukowo-badawczych Instytutu Maszyn Matematycznych" przyjmuje Dział Sprzedaży Wysyłkowej Ośrodka Rozpowszechniania Wydawnictw Naukowych PAN, Warszawa, Pałac Kultury i Nauki, tel. tel. 20-02-11 w. 2516. Egzemplarze pojedyncze Prac są do nabycia w księgniarni ORWN PAN, Warszawa, Pałac Kultury i Nauki, tel. 20-02-11 w.2105.

INSTYTUT MASZYN MATEMATYCZNYCH
BRANŻOWY OŚRODEK INFORMACJI
NAUKOWEJ, TECHNICZNEJ I EKONOMICZNEJ
WARSZAWA

CENY UJEDNOLICZONE

CENNIK OPRACOWAŃ I USŁUG

CENY OPRACOWAŃ

1. Tematyczne Zestawienie Dokumentacyjne
- koszt wg poniesionej robocizny przy cenie 1 r/godz. 1.200,-
2. Badania patentowe
- koszt wg poniesionej robocizny przy cenie 1 r/godz. 1.200,-
3. Kopie kserograficzne ze zbiorów biblioteki 1 strona A-4 26,-

CENY USŁUG INTROLIGATORSKO-REPROGRAFICZNYCH

1. Kopie kserograficzne A-4 1 strona 22,-
A-3 1 strona 33,-
2. Rozłożenie 100 str. 150,-
3. Oprawa introligatorska, miękka /o max. formacie A-4/
1 oprawa 300,-
4. Powielanie offsetowe oraz prace inne nie wymienione
wyżej - koszt wg poniesionej robocizny przy cenie
1 r/godz. oraz koszt materiałów 1.200,-