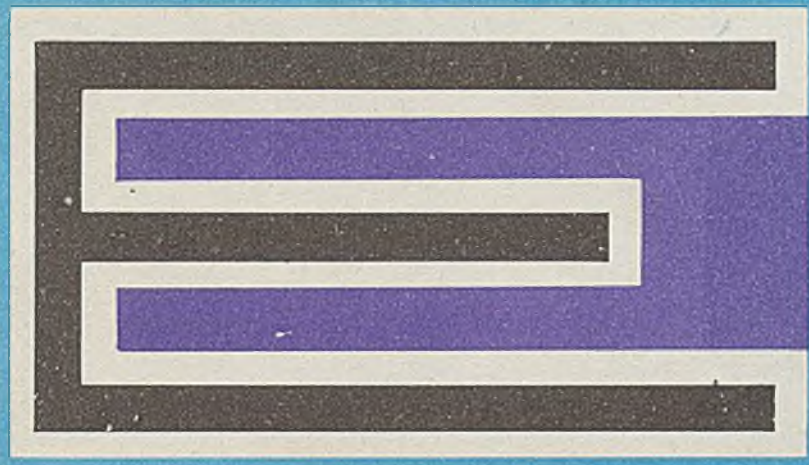
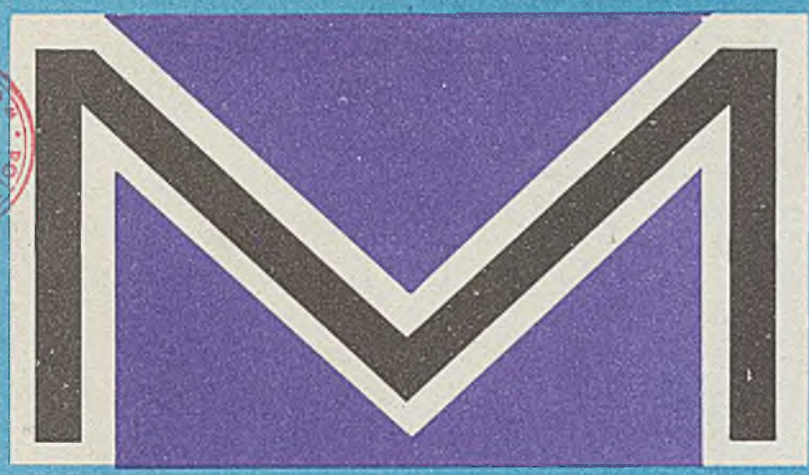


P. 2900 / 80



BIULETYN TECHNICZNY



11(225)
1980

Redakcja Kolegium w składzie:
mgr W. Borucki (redaktor działu „Ekonomika”),
mgr B. Drożak, mgr inż. J. Dziewięcki (redaktor naczelny), J. Esikowski,
mgr inż. R. Farfał, dr hab. M. Greniewski,
prof. dr hab. inż. A. Janicki (redaktor naukowy), inż. L. Kowalski,
mgr J. Kutrowska (sekretarz redakcji), mgr inż. L. Krzystalik, inż. R. Maciesowicz,
mgr E. Mańkiewicz-Cudny, red. T. Podwysocki, dr inż. R. Pregiel,
mgr inż. A. Teodorczuk, mgr inż. T. Ustaborowicz,
mgr inż. M. Wajcen (redaktor działu „Technika”)

Warunki prenumeraty

Jednostki gospodarki uspołecznionej, instytucje, organizacje i wszelkiego rodzaju zakłady pracy zamawiają prenumeratę w miejscowych Oddziałach RSW „Prasa-Książka-Ruch”, w miejscowościach zaś, w których nie ma Oddziałów RSW – w urzędach pocztowych. Czytelnicy indywidualni opłacają prenumeratę wyłącznie w urzędach pocztowych i u doręczycieli. Prenumeratę roczną w cenie 516 zł należy zamawiać do 25 listopada na rok następny, półroczną do 10 czerwca na II półroczu.

ZJEDNOCZENIE PRZEMYSŁU AUTOMATYKI
I APARATURY POMIAROWEJ „MERA”

9.2900 | 80



„MERA”

**BIULETYN PRZEMYSŁU
KOMPUTEROWYCH SYSTEMÓW
AUTOMATYZACJI I POMIARÓW**

WARSZAWA, LISTOPAD 1980

SPIS TREŚCI

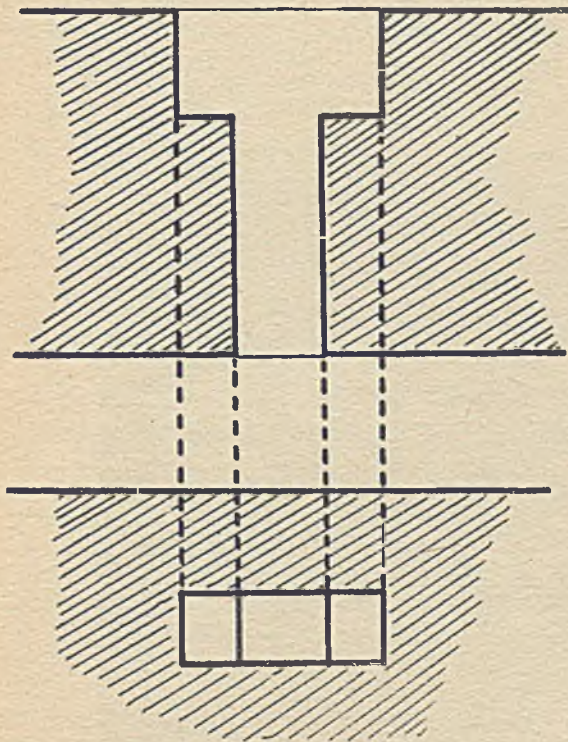
S. Stankiewicz	Wytwarzanie matryc do orientacji rdzeni ferrytowych pamięci maszyn cyfrowych	3
M. Grączewski	Matrycowe układy logiczne LSI	10
J. Kierkowski	Zależności wydajność - obciążenie w interaktywnych systemach operacyjnych	17
T. Müldner	Pewne uwagi o nowych językach programowania wysokiego poziomu: LOGLAN i ADA /część I /	23
A. Szewczyk	Modele symulacyjne w projektowaniu systemów informatycznych	28
T. Sinołęcki	Jugosłowska aparatura do automatyzacji procesów przemysłowych	33
<u>Informacje-Nowości</u>		
Komputerowy czytnik - sorter listów		36

Opracowanie Redakcyjne: Redakcja Biuletynu "Mera", ul. Patriotów 77, 04-950 Warszawa
 tel. 02-41-71/. Wydawca: Przedsiębiorstwo Automatyki Przemysłowej "Mera-Pnefal",
 ul. Poczni 19, 04-994 Warszawa. Zam.258/80. 2300 egz.

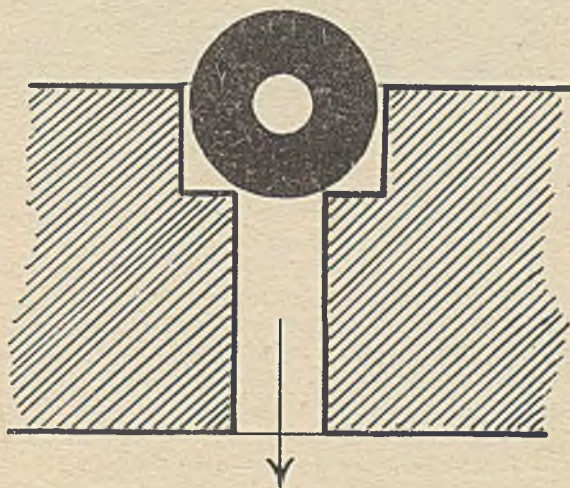
WYTWARZANIE MATRYC DO ORIENTACJI RDZENI FERRYTOWYCH PAMIĘCI MASZYN CYFROWYCH

Matryce do orientacji rdzeni ferrytowych

W realizacji pamięci ferrytowych występuje tzw. proces szycia polegający na wprowadzeniu przewodów w otwory rdzeni ferrytowych. W przypadku stosowanych obecnie rdzeni w kształcie cylindrycznych pierścieni o średnicy poniżej 1 mm, konieczne jest urządzenie pomocnicze do przestrzennego orientowania rdzeni przed szyciem. Głównym elementem takiego urządzenia jest tzw. matryca, pozwalająca na jednoczesne uszeregowanie i zorientowanie większych ilości rdzeni, co umożliwia przeszywanie ich prostymi przewodami [1]. Matrycę stanowi płytka metalowa posiadająca



Rys. 1. Otwór matrycy widziany w przekroju i z góry



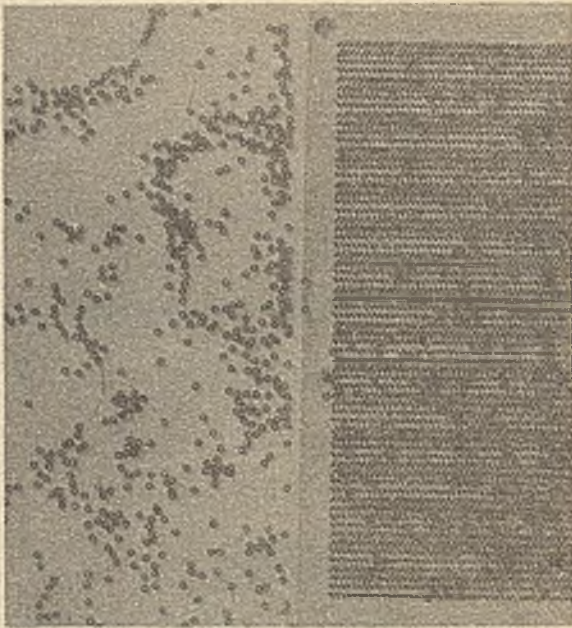
Rys. 2. Rdzeń wprowadzany do otworu matrycy

rozmieszczone rzędami otwory prostokątne, w których częściowo zagłębiamy się rdzenie.

Jedną z metod wprowadzania rdzeni do otworów jest stosowanie podciśnienia. W tym przypadku otwory w płytce wykonane są na wylot i posiadają różne wymiary z poszczególnych stron płytki /rys. 1/. Rdzenie nasypywane są na matrycę od strony większych otworów i zostają wprowadzone do nich przy pomocy powietrza zasysanego otworami z przeciwnej strony matrycy. Rdzenie wpadają do otworów matrycy w ten sposób, że ich osie są prostopadłe do osi otworów i zatrzymywane są w miejscu, w którym otwór matrycy zmienia wymiary /rys. 2 i 3/.

Tolerancje wymiarów otworów rdzeniowych muszą zapewnić swobodne wejście i wyjście rdzenia, właściwe jego zagłębienie się /do oparcia/ przy jednoczesnym ograniczeniu nadmiernego ruchu rdzenia w płaszczyźnie matrycy.

Stosuje się obecnie otwory rdzeniowe prostokątne o wymiarach 0,580x0,180 mm /dla rdze-



Rys. 3. Fragment matrycy i rdzenie

ni $\phi 0,54$ / oraz $0,480 \times 0,140$ mm /dla rdzeni $\phi 0,45$ / z tolerancją $+0,003$. Mniejsze otwory matrycy /do zasysania powietrza/ posiadają wymiary odpowiednio $0,26 \times 0,18$ mm i $0,25 \times 0,14$ mm. Całkowita grubość matrycy wynosi 1 mm, przy czym rdzeń wchodzi do otworu na głębokość $0,3$ lub $0,6$ mm - w zależności od stosowanej technologii szycia /szycie w matrycy lub na taśmie/. Format matrycy zależy również od technologii szycia oraz od możliwości technologicznych realizacji matryc odpowiednio dużych formatów. Dla rdzeni $\phi 0,54$ stosuje się obecnie matrycę o forma-

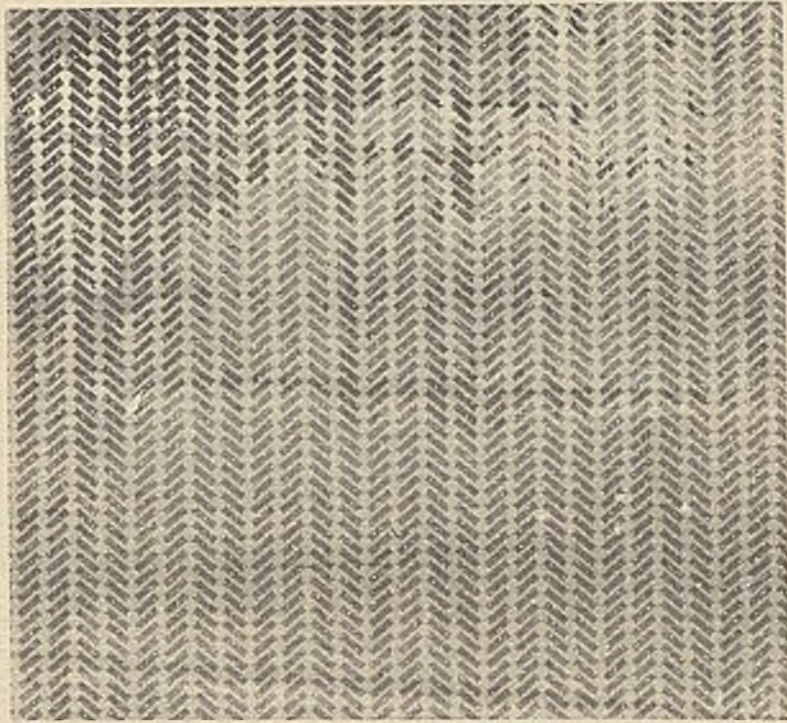
cie 43×60 mm, na której znajduje się 16 384 otworów /rys. 4/. Trwają opracowania oraz istnieją potrzeby na matryce większe. Dla rdzeni $\phi 0,45$ stosuje się matrycę o formacie 37×70 mm, która ma 15 876 otworów /rys. 5/.
Problemy realizacji matryc do orientacji rdzeni ferrytowych

Zgodnie z powyższym opisem matrycą ma być sztywna prostokątna płytka metalowa o powierzchni około 26 cm^2 , grubości 1 mm i zawierająca kilkanaście tysięcy otworów posiadających różne wymiary z obu stron płytki. Ze względu na dużą gęstość otworów i małe ich wymiary przy dużych wymaganiach w zakresie tolerancji tych wymiarów - wykonywanie matrycy jest zagadnieniem trudnym. Mechaniczna obróbka jest raczej niemożliwa. Pozostaje metoda określona jako trawienie kształtowe, polegająca na fotochemicznym selektywnym zabezpieczeniu powierzchni i wytrawianiu miejsc nie zabezpieczonych. Bezpośrednie zastosowanie tej metody do blach grubości 1 mm jest jednak niemożliwe, ze względu na dwuprzekrojowy kształt otworu oraz zbyt dużą grubość trawionej płytki, co prowadzi do znacznych podtrawień bocznych i związanego z tym powiększenia otworów.

Fotochemigrafia umożliwia natomiast realizację warstwy matrycy grubości $0,03 - 0,04$ mm mającą otwory o jednowymiarowym przekroju. Warstwy matrycowe uzyskiwane z odpowiedniej folii /taśmy/ metalowej mogą stanowić części składowe matrycy. W takim przypadku należy zastosować dwa rodzaje warstw składowych matrycy: z większymi i z mniejszymi otworami. Odpowiednie złożenie i trwałe połączenie około 30 takich warstw może utworzyć płytkę z otworami - matrycę.



Rys. 4. Powierzchnia matrycy dla rdzeni $\phi 0,54$



Rys. 5. Powierzchnia matrycy dla rdzeni \varnothing 0,45

Opisana metoda posłużyła do opracowania technologii otrzymywania matryc z warstw nazywanych dalej siatkami matrycowymi /podobieństwo do siatki/. Otrzymywanie siatek matrycowych z folii metalowej grubości 0,035 mm jest zagadnieniem fotochemigrafii precyzyjnej, która potrafi obecnie z powodzeniem sprostać takim zadaniom. Istotnym i trudnym problemem było natomiast opracowanie metody łączenia tych siatek i otrzymywania monolitycznej płytki przy zachowaniu założonych tolerancji wymiarów otworów rdzeniowych.

Wytwarzanie matrycy metodą łączenia poszczególnych jej warstw /siatek matrycowych/

Przy łączeniu siatek w matrycę należy rozpatrzyć dwa problemy:

- składanie i unieruchomienie kompletu siatek,
- trwałe łączenie siatek ze sobą.

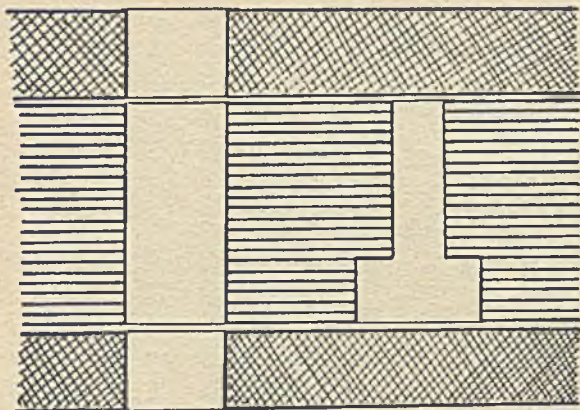
Do złożenia /ustawienia/ siatek potrzebny jest odpowiedni przyrząd dociskowy zaopatrzony w urządzenie do bazowania siatek. Nie wielkie przesunięcia siatek względem siebie spowodują zmniejszenie wymiarów otworów. Składanie siatek w przyrządzie dociskowym i ich bazowanie, odbywa się przy pomocy kołków bazujących umocowanych w tym przyrządzie oraz odpowiednich okrągłych otworów bazujących, znajdujących się na marginesach siatki. Konstrukcja przyrządu dociskowego jest uzależniona od metody łączenia siatek ze sobą. Materiałami łączącymi siatki mogą być: odpowiedni klej lub metal ulegający nadtopieniu, czy też wprowadzany elektrochemicznie z roztworu. Łączenie może odbywać się na wy-

branych punktach /np. punktowe zgrzewanie/, albo na całej powierzchni. W każdym wypadku materiał łączący nie może wchodzić w obszar otworów /np. przez wyciśnięcie/, co spowodowałoby zmniejszenie wymiarów otworów, a nawet ich zatkanie. Materiał łączący nie może też przekroczyć pewnych grubości, co spowodowałoby niedopuszczalne zwiększenie grubości całej matrycy. Materiał łączący powinien mieć dobrą przyczepność do siatek matrycowych oraz zapewniać dostatecznie mocne ich połączenia, nie zagrażające rozwarstwianiom w czasie pracy matrycy.

Metoda łączenia siatek matrycowych przy pomocy nitów galwanicznych

W początkowym okresie prac nad matrycami /lata 1971-72/ opracowano i stosowano metodę łączenia siatek polegającą na wytworzeniu szeregu nitów galwanicznych w systemie kanałów łączących z wprowadzonej tam elektrochemicznie miedzi /siatki matrycowe były wykonane również z miedzi/ [2]. Siatki oraz okładki przyrządu do łączenia miały otwory technologiczne, po złożeniu takich siatek w przyrządzie uzyskiwało się wspomniane kanałki łączące, w których osadzała się miedź przy zanurzeniu całego przyrządu do odpowiedniego elektrolitu, gdzie złożone siatki spełniały rolę katody /rys. 6/.

W czasie tej operacji przyrząd z siatkami wprowadzano w powolny ruch w celu wymiany elektrolitu w otworach. Proces prowadzono aż do całkowitego zamknięcia kanałów łączących



Rys. 6. Siatki między okładkami przyrządu do łączenia - przekrój przez otwór rdzeniowy i kanałek łączący

czyli do wytworzenia nitów galwanicznych [3]. Łączenie siatek w wyżej wymieniony sposób stosowano do matryc formatu 47x47 mm zawierających 4096 otworów do rdzeni ϕ 0,54 i 128 nitów galwanicznych utworzonych w kanałkach łączących ϕ 0,35. Ze względu na różne niedostatki tej metody /proces prowadziło się około 30 godzin/ oraz ograniczenie zagęszczenia otworów rdzeniowych spowodowane obecnością technologicznych otworów łączących - trzeba było szukać innej metody i technologii łączenia.

Metoda łączenia siatek matrycowych przez stapianie nałożonej galwanicznie warstwy stopu cyny i ołowiu

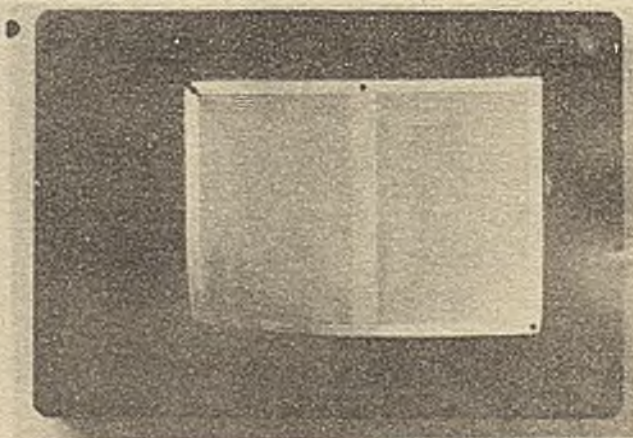
W trakcie doskonalenia metod wytwarzania matryc, opracowano i stosowano przez pewien czas technologię łączenia siatek matrycowych, polegającą na stapianiu nałożonej galwanicznie warstwy cyny i ołowiu [4]. Każdą siatkę matrycową pokrywano tym dwuskładnikowym spoiwem w elektrolicie zawierającym floroboran cynowy i ołowiany. Pokrycie zawierające oko-

ło 60% ołowiu i 40% cyny uzyskiwało grubość 4-5 μm . W następnej operacji pojedyncze siatki pokrywano topnikiem glicerynowym i zanurzano na około 3 s do oleju w temperaturze 230-250°C. Tak przygotowane siatki składano w przyrządzie do łączenia i zanurzano do oleju o temperaturze jak wyżej na okres kilku minut. Następowo połączenie siatek przez stopienie spoiwa. Istotą tej metody było stapianie spoiwa w temperaturze topnienia, co powodowało łączenie na zasadzie lutowania. Metoda ta odznaczała się również wieloma brakami i została zarzucona.

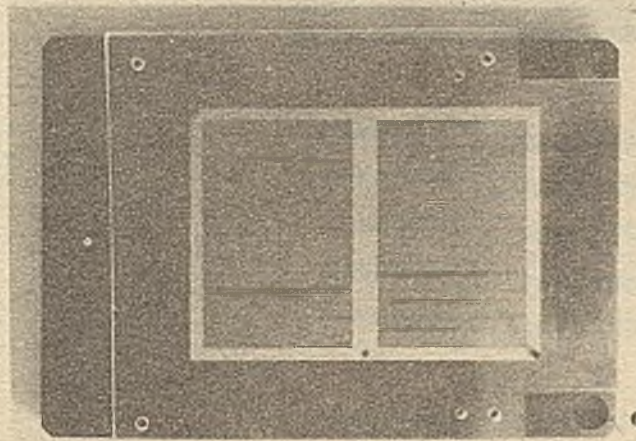
Metoda łączenia siatek matrycowych przy pomocy cyny nałożonej chemicznie

Rozwinięciem i udoskonaleniem metody poprzednio opisaney jest technologia posługująca się cyną nałożoną chemicznie i stapianie jej w piecu. Istotą tej metody jest dyfuzja cyny do miedzi w warunkach znacznego przekroczenia temperatury topnienia cyny oraz w warunkach powolnego stygnięcia, co powoduje znacznie trwalsze łączenie siatek w matrycę.

Praktyczne wykonanie wygląda następująco [5]: Siatki matrycowe po odpowiednim przygotowaniu ich powierzchni cynuje się chemicznie w roztworze opartym na chlorku cynawym i tiomoczniku. Uzyskuje się pokrycie cyną grubości 1-2 μm . Siatki ustawia się w przyrządzie do łączenia na jego kołkach bazujących /rys. 7 i 8/, a następnie rozprowadza się topnik na ostatniej siatce i skręca się przyrząd. Przyrząd zawierający komplet siatek ustawia się w piecu o temperaturze około 400°C. Po niecałej godzinie następuje wyłączenie pieca, który stygnie powoli do temperatury około 50°C, po osiągnięciu tej temperatury przyrząd z matrycą wyjmuje się. W okresie wysokich temperatur w piecu konieczne jest zapewnienie w komorze pieca atmosfery nie utleniającej. W metodzie tej istotne jest doświadczalne ustalenie



Rys. 7. Siatka włożona na kołki bazujące przyrządu do łączenia



Rys. 8. Komplet siatek w przyrządzie do łączenia

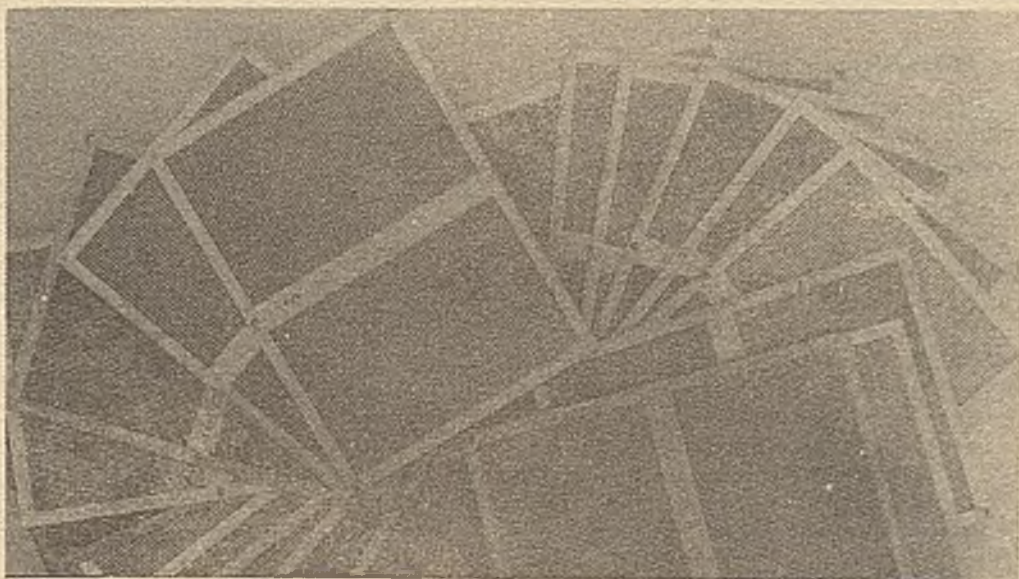
parametrów i warunków cynowania oraz łączenia, które muszą być ściśle zachowane w celu zapewnienia powtarzalności. Niedostateczna ilość nałożonej cyny powoduje rozwarstwienie się matrycy. Nadmierna ilość cyny powoduje jej wypływ do otworów rdzeniowych. Opisana metoda łączenia siatek w matrycy jest najmniej pracochłonna z wypróbowanych u nas i nadaje się do seryjnego wytwarzania matryc /rys. 9/.

Fotochemigrafia wytwarzania siatek matrycowych

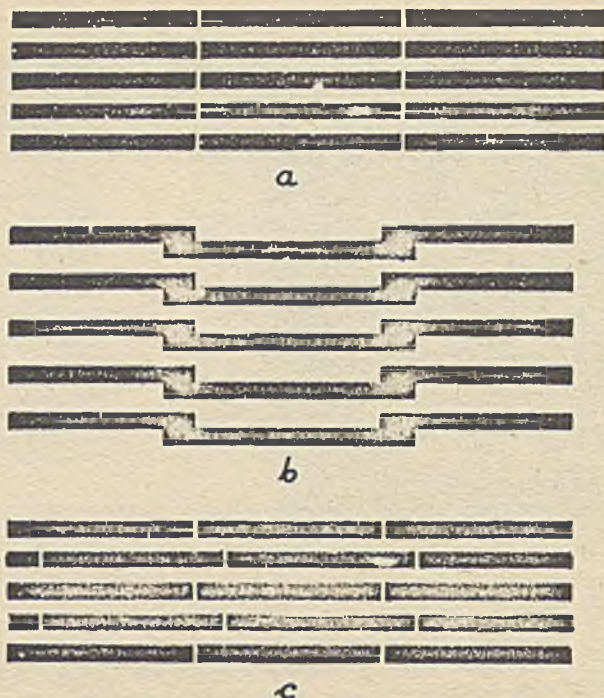
Części składowe matrycy, jej warstwy, zwane tutaj siatkami matrycowymi, otrzymuje się na drodze uzyskiwania fotochemigraficznej odbitki stykowej z fotoszablonu na folii miedzianej pokrytej fotorezistem oraz selektywnym trawieniu tej folii [5]. Na folię miedzianą odpowiednio oczyszczoną nanosi się przy pomo-

cy wirówki fotorezist Shipley AZ-1350. Jest to fotorezist pozytywowy, wymagający fotoszablonu, w którym miejsca otworów są jasne. Po naświetleniu w kopioramie wywołuje się w firmowym wywoływaczu. Przed trawieniem zabezpiecza się drugą stronę folii lakierem. Operacje z fotorezistem wymagają czystych i klimatyzowanych pomieszczeń. Trawienie prowadzi się w amoniakalnym roztworze chlorynu sodowego w temperaturze 30-40°C. Zalecane jest stosowanie odpowiedniej trawiarki pozwalającej na uzyskiwanie łatwo powtarzalnych parametrów trawienia.

Trawienie ręczne jest kłopotliwe i wymaga częstej kontroli mikroskopowej w trakcie prowadzenia operacji ze względu na przetrawienia lub niedotrawienia, które w przypadku trawienia ręcznego są rozłożone nierównomiernie na powierzchni siatki. W celu eliminacji lub zmniejszenia kształtowych błędów trawienia



Rys. 9. Matryce wykonane seryjnie



Rys. 10. Sposoby układania siatek: stykowo, na zakładkę, stykowo z przesunięciem marginesów.

/otwory prostokątne przy trawieniu ulegają zaokrągleniu/, przyjęto wprowadzanie określonych deformacji na narożach otworów fotoszablonów, co znacznie polepszyło odwzorowanie kształtów po trawieniu. Następną operacją jest szycie fotoszablonu oraz zabezpieczenie odwrotnej strony siatki. Dalej następuje mikroskopowa kontrola otworów w siatkach oraz ich selekcja. Siatki dzieli się na kilka grup. Te które spełniają całkowicie wymagania wymiarowe stosuje się jako pierwsze w górnej części matrycy, siatki nieco odbiegające od wymagań /o powiększonych otworach/ mogą być użyte w dolnej części matrycy. Siatki o zwężonych otworach rdzeniowych są odrzucane.

Seryjne wytwarzanie matryc

Zapotrzebowanie na matryce jako narzędzia pomocnicze przy operacji szycia pamięci jest uzależnione od wielkości produkcji pamięci, organizacji produkcji oraz w dużym stopniu od trwałości matryc. Matryce wymienia się w przypadku pojawienia się miejscowego rozwarstwienia, przy wystąpieniu uszkodzeń mechanicznych oraz na skutek jej naturalnego zużycia /rozkalibrowanie otworów rdzeniowych/. Biorąc pod uwagę wszystkie powody niesprawności matrycy, średni czas jej pracy wynosi w naszym przypadku 1250 operacji zasypania i ustawienia rdzeni /dotyczy matryc dla rdzeni $\phi 0,54$ mających ponad 16 tys. otworów/. Ze względu na małe zapotrzebowanie oraz specyfikę technologii wytwarzania - matryce produkuje się w skali i w warunkach laboratoryjnych. Koszt wytwarzania matryc zawiera się przede wszystkim w robociźnie. Koszt materiałów nie prze-

kracza 10% całkowitego kosztu wytwarzania matryc. Materiały z importu /fotorezist i wywoływacz firmy Shipley oraz folia miedziana/ stanowią około 50% całkowitych kosztów materiałowych, czyli około 5% całkowitych kosztów wytwarzania matryc.

Zasadnicze ciągi lub stanowiska wytwarzania matryc są następujące: fotochemigraficzne wykonywanie siatek matrycowych, selekcja siatek, łączenie siatek w matryce, kontrola i próba matryc, operacje i czynności pomocnicze. Najbardziej pracochłonne operacje to wytwarzanie siatek oraz ich selekcja. Odpad przy wytwarzaniu siatek wynosi 20-25%. Do złożenia jednej matrycy potrzeba ponad 30 siatek grubości 0,035mm. Odpad operacji łączenia siatek w matryce wynosi 20-40%. Uzyskiwanie małych wydajności wytwarzania siatek i matryc wiąże się z trudnością zachowania powtarzalności warunków prowadzenia niektórych procesów /np. ręcznego trawienia/.

Niektóre zagadnienia rozwojowe

Rozwój w wytwarzaniu matryc do orientacji rdzeni ferrytowych może postępować w następujących kierunkach: zwiększenie formatu matryc, zwiększenie gęstości otworów na matrycy, zmniejszenie otworów rdzeniowych /stosowanie mniejszych rdzeni/, zwiększenie wydajności wytwarzania matryc, zwiększenie żywotności matryc. Zmiany konstrukcyjne matryc są uzależnione od potrzeb operacji szycia pamięci, od zmian w technologii szycia. Konieczność zwiększenia formatu matryc może spowodować potrzebę zastosowania innego wyposażenia - odpowiadającego tym formatom, lub zastosowania innych technologii.

Zwiększenie formatu powoduje wystąpienie różnych trudności. Tak więc wydajność wytwarzania siatek obniża się, a łączenie siatek w matryce wymaga przyrządu, który zapewniłby równomierny docisk również na zwiększonej powierzchni. W celu uniknięcia niektórych trudności, można realizować większy format docelowy matrycy z siatek mniejszego formatu. Taką realizację można sobie różnie wyobrazić. A więc łączenie ze sobą fragmentów "na styk" lub "na zakładkę". Pierwszy sposób znacznie osłabiłby matrycę mechanicznie, drugi sposób spowodowałby powstanie niedopuszczalnych zgrubień na matrycy. Pozostaje więc metoda, którą można określić jako przesunięcie styków krawędzi siatek [7]. Polega ona na wykonaniu serii siatek składowych, które nie różniąc się obszarami czynnymi /obszary otworów rdzeniowych/ posiadają marginesy różnej szerokości /rys. 10/.

Układając takie siatki na kolkach bazujących przyrządu do łączenia uzyskuje się pokrycie otworów rdzeniowych, natomiast brzegi marginesów siatek tworzą w przekroju linię "układania cegieł". Ułożone w ten sposób i połączone siatki gwarantują dostateczną wytrzymałość mechaniczną powstałej matrycy. W ten

sposób można otrzymać matryce znacznych rozmiarów. Zmniejszenie odległości między otworami rdzeniowymi oraz zmniejszenie wymiarów tych otworów, to przede wszystkim precyzja fotochemigrafii, która w zakresie rozpatrywanych tu potrzeb i wymagań potrafi sprostać stawianym zadaniom. Dodatkowo można wprowadzić uzyskiwanie fotochemigraficznej odbitki z obu stron folii miedzianej i trawienie tej folii z obu stron, co prowadzi do zmniejszenia podtrawień, ewentualnie umożliwia stosowanie grubszej folii. Zwiększenie wydajności wytwarzania siatek i matryc oraz zwiększenie średniej żywotności matryc można osiągnąć na drodze zaostrzenia powtarzalności parametrów fizycznych procesów, wprowadzając dodatkową czy też ściślejszą ich kontrolę. Szczególnie istotna jest kontrola powtarzalności w procesach trawienia siatek, pokrywania siatek cyną, dociskania siatek w przyrządzie do łączenia oraz łączenia siatek w piecu.

L i t e r a t u r a :

[1]. S. Stankiewicz, Cz. Puła: Prace Ośrodka Badawczo-Rozwojowego "Mera-Elwro" 1. 36. 1974. .

[2]. M. Buzdygan, S. Witman, W. Łupiński: Patent PRL nr 713118.

[3]. W. Łupiński: Instrukcja technologiczna Ośrodka Badawczo-Rozwojowego "Mera-Elwro" JT-4-783, 1972.

[4]. J. Skopiec: Opis technologiczny TT-35 Laboratoryjne wykonywanie siatek matryc pamięci oraz ogniowe łączenie ich w matryce, 1974.

[5]. W. Łupiński: Instrukcja wewnętrzna Zakładu Nowych Technik Instytutu KSAiP - Wykonywanie matryc do ustawiania rdzeni ferrytowych, 1978.

[6]. W. Łupiński: Sprawozdanie Ośrodka Badawczo-Rozwojowego "Mera-Elwro" S-BKT-044/77 - Prace rozwojowe w zakresie trawienia kształtowego o dużej dokładności w warstwach metalowych /dokument wewnętrzny, nie publikowany/.

[7] M. Milejczak: Projekt wynalazczy "Mera-Elwro" WZE/865/75.



MATRYCOWE UKŁADY LOGICZNE LSI

Realizacja układowa sieci logicznych jest aktualnie możliwa w oparciu o wielką bazę podzespołową cyfrowych układów scalonych, przy dominującej roli układów o możliwie największej skali integracji [1]. Wśród tych układów można wyróżnić trzy główne klasy [2].

1. Klasa układów standardowych, zwanych również "układami z półki" /sklepowej/ - off-the-shelf, jest klasą układów najtańszych, sprzedawanych w dużych partiach. Z układami tymi, szczególnie w zakresie zaawansowanych technologii, takich jak ECL, wiąże się jednak poważne ryzyko gwałtownego ich braku na półkach sklepowych w przypadku bardzo prawdopodobnego przyjęcia wybranego typu lub rodziny układów do wielkoseryjnej produkcji sprzętu przez wielki przemysł komputerowy. Klasa tych układów do niedawna była klasą jedyną, jednak dynamiczny rozwój oraz osiągnięty stan dojrzałości technologii mikroelektronicznych umożliwił pojawienie się dwóch następnych klas.

2. Klasa układów na zamówienie odbiorcy - custom design IC - obejmuje układy stosunkowo drogie, opracowane i produkowane dla jednego odbiorcy. Czas opracowania wynoszący 1 do 2 lat oraz konieczność całkowitego pokrycia kosztów opracowania przez jednego odbiorcę nie sprzyja szybkiemu rozwojowi tej klasy układów.

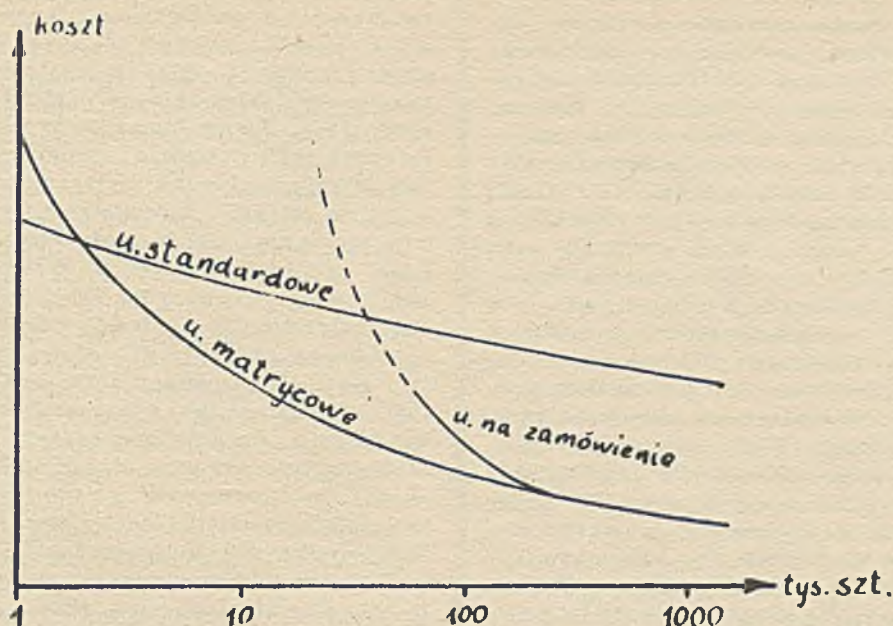
3. Klasa układów matrycowych obejmuje konstrukcje monolityczne zawierające w jednej strukturze wielokrotne zbiory elementów lub celi /komórek/ elementów wdyfundowanych w krzemową płytkę.

Cienkowarstwowa sieć połączeń na ogół od jedno do trzywarstwowych łączy elementy jednej struktury w sieć logiczną. W niniejszym artykule przyjęto nazwę matryca dla określonej topologii rozmieszczenia elementów w pojedynczej strukturze, a nazwę struktura matrycowa dla pojedynczej struktury układu ma-

trycowego, lecz bez sieci połączeń. Określenie struktura układu matrycowego dotyczy będzie natomiast monolitycznej struktury matrycowej z siecią połączeń. Na bazie jednej matrycy można po zaprojektowaniu wielu sieci połączeń uzyskać wiele typów matrycowych układów logicznych, w zależności od potrzeb odbiorców. Realizacja odrębnych układów matrycowych sprowadza się do realizacji sieci połączeń na strukturach matrycowych już uprzednio przygotowanych. Rozwiązanie takie cechuje się krótkim i tanim opracowaniem nowego układu, lecz jest ograniczone przez konieczność użycia przyjętej do struktury matrycowej bramki, jako cegiełki do budowy bloków LSI /co wydłuża czasy propagacji sygnałów w stosunku do poprzednich klas układów/ oraz przez zmniejszenie efektywności wykorzystania powierzchni struktury. Klasa układów matrycowych przejawia jednak dynamiczny rozwój, co świadczy o ich rosnącej atrakcyjności oraz minimalizacji wymienionych wad.

Wg danych firmowych [3] w klasie układów matrycowych opłacalność monolityzacji uzyskuje się już dla serii mniejszych niż 50 tys. sztuk układów, podczas gdy opłacalność tą dla układów poprzednich klas można uzyskać dopiero dla serii większych od 200 tys. sztuk układów. Dla serii od 50 tys. sztuk do 200 tys. sztuk opłacalność monolityzacji w danej klasie zależy jest od rodzaju układu. Wielkość kosztów jednostkowych dla trzech ww. klas układów, w zależności od długości serii produkcyjnej, przedstawia rys. 1. [4].

Omówione wyżej przewagi ekonomiczne układów matrycowych nad pozostałymi klasami układów w zakresie krótkich serii produkcyjnych, preferują je do zastosowań w sprzęcie profesjonalnym. Charakterystyczną cechą układów matrycowych jest możliwość wykonania przez odbiorcę sieci połączeń oraz mikro-montażu i hermetyzacji układów; na bazie pół-fabrykatów, jakimi mogą być niepocięte płytki



Rys. 1. Koszt jednostkowy układów w zależności od długości serii dla trzech klas układów [4].

krzemowe z wdyfundowanymi zespołami elementów, dostarczane przez producenta układów matrycowych. Możliwość przejęcia pewnych funkcji procesu projektowania i produkcji układów przez ich odbiorcę, dotyczy w przypadku układów matrycowych prac w zakresie projektowania sieci połączeń. Tacy producenci jak Exar [4], International Microcircuits, Inc., Ferranti [5], [6], [7] i inni dostarczają odbiorcy szczegółową dokumentację techniczną z rysunkami matryc niepołączonych elementów wdyfundowanych w płytkę krzemową.

Po naniesieniu na dostarczoną dokumentację szkicu sieci połączeń i sprawdzeniu działania zaprojektowanego układu na zestawie dostarczonych indywidualnych obudowanych układów scalonych, zawierających elementy logiczne odpowiadające parametrami elementom możliwym do uzyskania siecią połączeń na strukturze matrycowej - odbiorca przesyła szkice do producenta, który na przykład [8] oferuje możliwość dostawy serii prototypowej rzędu 100 sztuk po 9 tygodniach za cenę 20 tys. dolarów w przypadku matrycy 230-bramkowej/. Dalsze dostawy następują w cenie 20 dolarów za układ dla serii do 1,5 tys. sztuk oraz w cenie 4,8 dolara za układ dla serii powyżej 80 tys. sztuk. Osiągnięte dla układów matrycowych tak poważne zmniejszenie kosztów oraz czasu opracowania i wykonania krótkich serii, w stosunku do standardowych układów scalonych, jest wynikiem ograniczenia projektowania układu do projektowania sieci połączeń matrycy oraz możliwości dysponowania gotowymi zapasami struktur matrycowych.

Należy podkreślić, że organizacja taka umożliwiła odbiorcy zarówno projektowanie jak

i wykonawstwo sieci połączeń oraz - w przypadku zaawansowania w technikach mikroelektronicznych - wykonawstwo masek sieci połączeń. W efekcie odbiorca może uzyskać całkowitą ochronę tajemnicy przemysłowej dotyczącą rozwiązania układowego produkowanego sprzętu. Regularny charakter topologii matrycy stwarza korzystne warunki dla automatyzacji projektowania. Wg dostępnych informacji systemy CAD układów matrycowych posiadają Motorola [9], IBM [14], oraz Texas Instruments [16].

Należy przypuszczać, że połączenie systemów CAD z generatorami masek, jak to ma miejsce w IBM, pozwala uzyskać elastyczność systemu projektowo-produkcyjnego układów matrycowych stanowiącą nową jakość w mikroelektronice. Texas Instruments [16] przewiduje jeszcze w 1980 r. zmniejszenie czasu projektowania topologii do 4 dni, a w 1982 r. zmniejszenie czasu produkcji do ok. 5 dni !!! dzięki połączeniu systemów wspomagania komputerowego i litografii elektronicznej do bezpośredniej obróbki płytek krzemowych z monolitycznymi strukturami matrycowymi.

W opinii Texas Instruments w Huston [16] układy matrycowe są następną generacją głównego kierunku rozwojowego układów logicznych. W produkcję układów matrycowych zaangażowana jest aktualnie poważna ilość firm, zarówno z grupy kolosów przemysłu półprzewodnikowego, jak i drobnych przedsiębiorstw produkujących wyłącznie układy matrycowe, co ilustrują następujące przykłady. International Microcircuits Inc. /IMI/, USA, oferuje Masterslice Arrays 7000 Family w technologii CMOS w technice 6 μ m w czterech typach ma-

tryc od 240 do 2000 bramek o czasie opóźnienia 5 ns. Czas dostawy pierwszej zamówionej serii nowych układów wynosi 9 tygodni /Electronic Design, 6/ March 15, 1979/. Na koniec 1980 r. IMI zapowiada [16] struktury liczące 5000 bramek z czasem opóźnienia 3 ns realizowane techniką 3 μm , z dwuwarstwową metalizacją. American Microsystems, Inc. /AMI/ - USA oferuje Unconnited Logic Arrays w technologii CMOS w technice 5 μm w sześciu typach matryc od 300 do 1260 bramek o czasie opóźnienia w granicach od 5 do 9 ns. [20]. Siemens /RFN/ produkuje szereg układów matrycowych w technologii ECL, m. in. doprzewidywanego w produkcji pod koniec 1981 r. komputera 7300. Przykładowe układy matrycowe Siemens: SH100B/typ 0, 24-komórkowy w strukturze o powierzchni 28 mm^2 , z 500 równoważnikami bramek wydzielający 1,8W oraz SH100B/typ 1 36-komórkowy w strukturze o powierzchni 36 mm^2 z 800 równoważnikami bramek, wydzielający 2,3W. Obydwa te układy posiadają ponadto odpowiednio 30 i 42 komórki wejściowe oraz 38 komórek wyjściowych. Czas opóźnienia wynosi 0,5 ns/bramkę. Odpowiedniki tych układów z zamienialnymi z Siemensem maskami sieci połączeń produkuje La Radiotechnique-Compelec /RTC/ we Francji, oznaczane jako typy MLA24 i MLA36 [16]. Siemens [11] podaje interesujące wskaźniki topologiczne.

- Komórki logiczne zajmują 25% obszaru struktury matrycowej.
- Sieć połączeń międzykomórkowych: 25% obszaru struktury matrycowej.
- Powierzchnia wyprowadzeń: 16% obszaru struktury matrycowej.
- Sieć zasilania i układy dopasowujące: 24% obszaru struktury matrycowej.

Smiths Industries Ltd. /W. Brytania/ zapowiada [16] ofertę na koniec 1980 r. na układ matrycowy z 2000 równoważnikami bramek w wersjach o 48, 69 i 180 wyprowadzeniach, w technice 4-5 μm , informując równocześnie o zamierzeniach na 1982 r. produkcji układu 5000 bramkowego. Szereg firm japońskich /NEC, Hitachi, Mitsubishi/ produkuje i rozwija układy matrycowe. Firma Fujitsu [16] stosuje do układów matrycowych technologie CMOS TTL i ECL. W technologii CMOS produkuje m. in. 4 typy matryc: 770, 1275, 2000 i 3900 bramek w technice 3,6 μm . Największa 3900 bramkowa struktura ma wymiary 9,7 x 9,7 mm i wydziela maks. 100 mW.

W dalszym ciągu artykułu zostaną przedstawione niektóre szczególnie interesujące rozwiązania techniczne układów matrycowych.

Układy matrycowe dla sprzętu komputerowego

Współczesne komputery posiadające prędkość działań rzędu kilku milionów operacji na sekundę bazują na układach scalonych małej i średniej skali integracji, jak np. seria MS10000 o czasie opóźnienia 2 ns i mocy wydzielanej

rzędu 25 mW na bramkę. Dalsze możliwości zwiększania prędkości komputerów upatruje się w zwiększeniu prędkości bazy elementowej przez produkcję układów scalonych LSI "na zamówienie" oraz układów scalonych matrycowych [12]. Firma Motorola przewiduje [13], że rynek 1980 r. będzie wymagał 10 mln dolarów produkcji układów matrycowych, a w 1985 r. - 50 mln dolarów. Natomiast Ferranti przewiduje rynek dla układów matrycowych rzędu kilkuset milionów dolarów, nie tylko dla komputerów, lecz i dla sprzętu powszechnego użytku. Stwierdza, że wynika to z faktu, iż konwencjonalne układy małej i średniej skali integracji dają projektowaniu elastyczność i prędkość logiki lecz za wysoki koszt dużej złożoności konstrukcji, ograniczonej niezawodności i wielkiego zużycia obwodów drukowanych, natomiast układy "na zamówienie" są rzadko opłacalne, ze względu na wielką liczbę bramek i funkcji logicznych w sieciach logicznych komputerów.

Zalety układów matrycowych są w tym świetle szczególnie atrakcyjne. Stosuje się technologie ECL, TTL, I²L, CMOS u takich producentów jak: Ferranti, Motorola, RCA, RTC, Fairchild, Fujitsu, Philips, Plessey, Siemens i Signetics. Znanym odbiorcą układów matrycowych jest Amdahl - USA do komputerów 470 V/6 i V/7. Układy matrycowe produkują dla tej firmy [13] Fujitsu i Motorola z masek Amdahl'a. V/6 zawiera 2 tys. układów o 102 typach. Firma Control Data odbiera od Fairchilda układy matrycowe subnanosekundowe ECL, a International Computers Ltd stosuje 400-bramkowe matryce od Philipsa, Plessey'a i Motoroli.

Bazą podzespołową [12] jednostki centralnej V/6 jest prefabrykat 100-bramkowy ECL o czasie opóźnienia 650 ps i mocy wydzielonej 30 mW na bramkę. Całość układów jest realizowana /stan 1975 r. / 13 maskami, w tym 3 maskami dla warstwowej sieci połączeń. Struktura ma wymiary 4x4 mm i 84 wyprowadzenia. Obudowa ma radiator dla 4W mocy wydzielonej dla 85°C maks. temperatura złącza. W efekcie zastosowania układów matrycowych czas opóźnienia zmniejszył się 3-krotnie w stosunku do wersji V/6 na serii układów standardowych MS104 Motoroli, a intensywność uszkodzeń wyniosła 1. 10⁻⁶/tys. godz. dla średnio 70 bramek na układ, w porównaniu z 7. 10⁻⁶/tys. godz. dla średnio 4-8 bramek na układ w wersji MECL 10000. Stosunek szybkość /cena V/6 posiada 2 razy lepszy niż IBM-370 /158 i IBM/168. Szybkość wynosi 8 mln operacji na sekundę. W 1977 r. pojawił się V/7 o prędkości [12] 13,6 mln operacji na sekundę z układami matrycowymi "drugiego pokolenia" w ilości 130 typów dającymi dwukrotną poprawę stosunku szybkość/cenę. CDC przewiduje, że wdrożenie układów matrycowych pozwoli jej osiągnąć w systemie Star-100A szybkość 400-500 mln operacji na sekundę.

Wyróżniającą się konstrukcją są LSI Masterslice IBM, które są bazą podzespołową kon-

cepcji IBM obejmującej kompleksowe udoskonalenie całości konstrukcji komputerów w oparciu o istniejącą technikę mikroelektroniczną. Realizację tej koncepcji stanowi technika wielowarstwowych modułów ceramicznych /Multilayered Ceramic Modules-MCM/ wprowadzona w komputerach System/38 z General Systems division of IBM, który ma zastąpić stary system/3 do prac w zakresie finansów, oraz w komputerach 4300 z Data Processing divisions of IBM, z których 4331 ma zastąpić System/370 model 138. Konstrukcja nowych komputerów jest konsekwencją kompleksowej modernizacji rozwiązań konstrukcyjno-technologicznych od poziomu monolitycznych struktur krzemowych, aż do bloków sprzętowych.

Technologia nowych komputerów stanowi aplikację najnowocześniejszych technologii mikroelektronicznych:

- monolitycznych struktur matrycowych LSI z wyprowadzeniami poduszgowymi do lutowania rozpliwowego,
- wielowarstwowych modułów ceramicznych z dolutowanymi strukturami matrycowymi,
- pakietów wielowarstwowych obwodów drukowanych z modułami ceramicznymi,
- płyt bazowych wielowarstwowych obwodów drukowanych dla łączenia pakietów w bloki sprzętowe z zastosowaniem specjalnych złączy wielostykowych.

Wprowadzone przez IBM do MCM struktury układów matrycowych Master-Slice są dostosowane do montażu techniką lutowania rozpliwowego /reflow-soldering/ metodą "face-down" /twarzą w dół/ do podłoża ceramicznych z nadrukiem grubowarstwowym pierwszej warstwy sieci połączeń MCM. W tym celu struktury Masterslice są deplanaryzowane przez utworzenie na ich powierzchni wypukłych poduszek cynowo-olowiowych w rastrze wyprowadzeń struktury.

Wymiary przykładowej struktury, stosowanej do MCM w komputerach serii 4300 wynoszą 5,66x5,66 mm, a wymiary jej powierzchni aktywnej - 5,43x5,43 mm. Wymiar rastru wynosi 250 μ m, a maksymalna ilość wyprowadzeń poduszgowych jednej struktury od 11x11 do 17x17 szt. [17].

Warstwy metalizacji sieci połączeń struktury mają izolację SiO₂. Pierwsze dwie warstwy tworzą połączenia sieci logicznej matrycy. Trzecia warstwa stanowi połączenia wejścia-wyjścia i zasilania. Szerokość typowych połączeń wynosi 5 μ m przy minimalnej odległości środków 6,5 μ m. Łączna długość połączeń dla 1200 bramek jest rzędu 4 m dla pierwszej i drugiej warstwy. Zastosowano dla połączeń stop miedź-aluminium. Bipolarna struktura zawiera 1496 bramek logicznych, 88 układów wejściowych, 64 wyjściowe układy sterujące i dwa generatory referencyjne. Struktura posiada 94 we/wy do współpracy z układami TTL. Zasilanie struktury jest napięciami +5,0V -10% i 1,70V -5%. Wydzielenie mocy [17] wynosi 1,8W, a temperatura pracy złącza 25±

100°C. Struktura charakteryzuje się organizacją komórkową o 748 komórkach w 33 wierszach i 22 kolumnach. Elementy komórki mogą być połączone w dwie bramki, dając łącznie 1496 bramek. Standardowa bramka sprzętowa jest realizowana w klasie układów o sprzężeniu emiterowym ECL. Opóźnienie typowe wynosi 1,5 ns przy 0,85 nW/1,28 pJ/. Opóźnienie w wersji o zwiększonej mocy wynosi 0,8 ns przy 1,7 mW.

Cykl projektowania jest kompleksowo ujęty w Engineering Design System /EDS/, który obejmuje symulację logiczną, topologię sieci połączeń i generację testów. Wg informacji opublikowanych pod koniec 1979 r. [18], istniały wówczas cztery odmiany struktur matrycowych IBM wytwarzane w Data Systems division w East Fishkill, N.Y. [19], przeznaczone do komputerów System/38 i 4300. Pod koniec 1980 r. pojawiły się wzmianki [16] o matrycy 5000-bramkowej IBM. Blumberg i Brenner [17] podają informację o wykorzystaniu przez IBM w East Fishkill ekspozera elektronowego do generacji wzorów sieci połączeń na monolitycznych strukturach matrycowych metodą litografii elektronowej. Metoda ta umożliwia formowanie maski emulsyjnej na płytkach krzemowych przy pomocy strumienia elektronów sterowanego w systemie Engineering Design System /EDS/, stosowanym w IBM do projektowania układów matrycowych Masterslice.

Dostosowanie struktur matrycowych Masterslice do montażu metodą lutowania rozpliwowego z możliwością uformowania wyprowadzeń poduszgowych na wybranych polach rastru daje dodatkową istotną przewagę nad innymi konstrukcjami układów matrycowych, polegającą na powiększeniu maksymalnej ilości wyprowadzeń, oraz swobodzie ich doboru w rastrze. Wg IBM [16] stosunek ilości wyprowadzeń do ilości bramek układu winien wynosić 1:10.

Układy matrycowe Macrocell Motorola

Układy LSI Masterslice IBM zostały opracowane i są produkowane przez producenta komputerów na własny użytek. Inny przykład układów matrycowych stanowi Macrocell Array firmy Motorola - USA, pomyślany w pierwszym rzędzie jako rozszerzenie możliwości rodziny mikroprocesorów segmentowych M10800 tej firmy [14]. Specyfikę matrycy Macrocell realizowanej w technologii ECL stanowi jej podział na tzw. makrokomórki stanowiące bloki funkcjonalne - odpowiedniki układów scalonych małej lub średniej integracji i składające się z niepołączonych 52 szt. tranzystorów i 48 szt. rezystorów. Matryca jest złożona z 3 typów komórek:

- komórek głównych /48 szt./ składających się z ok. 50 tranzystorów i 50 rezystorów,
- komórek interfejsu /32 szt./,
- komórek wyjściowych /26 szt./.

Komórki główne /6x8/szt. zajmują wewnętrznym obszar matrycy, a komórki interfejsu oraz

wyjsciuowe usytuowane są na obrzeżach matrycy.

Przy 90% pracujących komórek układ matrycowy pobiera ok. 4W mocy. Struktura układu posiada 60 wyprowadzeń we/wy oraz do 1192 równoważników bramek, co daje 5 mW na bramkę. Czas opóźnienia zawiera się w granicach 0,9-1,3 ns. Macrocell są kompatybilne z rodziną 10k ECL. Układy I³L na bazie matryc Macrocell są projektowane w systemie CAD Motoroli. System zawierający podręczniki projektowania z biblioteką funkcji umożliwia projektowanie projektantom nie wyspecjalizowanym w logice ECL.

Biblioteka obejmuje:

- dla komórek głównych 54 funkcje,
- dla komórek interfejsu 14 funkcji,
- dla komórek wyjściowych 17 funkcji.

Odbiorca może korzystać z systemu CAD przez linie telefoniczne, jeżeli posiada wydzierżawiony lub nabyty terminal graficzny, ewentualnie dodatkowo plotter. Na ekranie terminala można uzyskać obraz topografii matrycy, a po wybraniu z biblioteki pożądaných funkcji wprowadzić je dla określonych makrokomórek matrycy, oraz określić wyprowadzenia we/wy i połączenia pomiędzy komórkami. System CAD uwzględnia możliwość weryfikacji logiki na drodze symulacji logicznej i eliminację przekroczeń reguł projektowania sieci połączeń. Po skompletowaniu danych w systemie CAD, zostają one przesłane do odbiorcy do końcowej weryfikacji. Po akceptacji odbiorcy system CAD umożliwia przekazanie danych do generacji masek. Czas nowego opracowania układu w systemie CAD Macrocell Motoroli jest oceniany na 12 tygodni [2].

Struktury matryc Macrocell montowane są do czworokątnych obudów ceramicznych typu QIL o powierzchni ok. 1 cala kwadratowego z radiatorem i 68 wyprowadzeniach. Rezystancja termiczna obudowy wynosi 15°C/W.

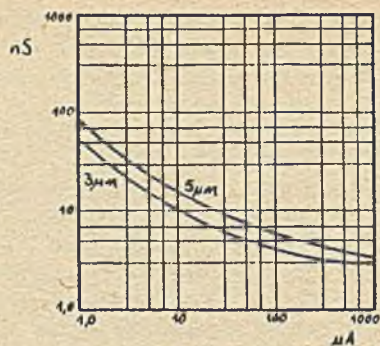
Układy matrycowe firm Fairchild i Ferranti

Firma Fairchild - USA, ma dwa programy [16] układów matrycowych: w technologiach ECL i I³L /Isoplanar Integrated Injection Logic/.

W technologii ECL Fairchild wprowadza szybką rodzinę F-300 o czasach opóźnienia mniejszych niż 0,5 ns. F-300 liczy 7 typów układów matrycowych mających w organizacji komórkowej orientacyjnie od 1000 do 5000 równoważników bramek wydzielających od 2W do 20W mocy w zależności od typu. Powyżej 6W zachodzi konieczność stosowania chłodzenia wodnego. Podkreśla się [16], że w trzech typach układów matrycowych z rodziny F-300 wprowadzono ponadto programowane matryce logiczne, a w największym z nich - liczącym ponad 5000 równoważników bramek - pamięć RAM.

W technologii I³L [15] Fairchild posiada matrycę 4000-bramkową typu F 9480. Do zalet technologii I³L firma zalicza:

- wielką gęstość upakowania,



Rys. 2. Zależność czasu opóźnienia /ns/ od prądu /µA/ bramki podstawowej I³L układu matrycowego F9480 [15].

- kompatybilność technologii I³L z technologią bipolarną,
- możliwość pracy przy małych napięciach i mocach,
- dobry współczynnik \tilde{U} . P,
- większą szybkość niż MOS.

Matryca F9480 została opracowana dla systemów μ P, jak np. 16-bitowego 9445. Zawiera 4000 bramek logicznych oraz 14⁰ układów pomocniczych, m.in. dopasowujących matrycę do standardów TTL. Sieć połączeń jest zbudowana w postaci 3-warstwowej metalizacji. Komórki matrycy składają się z 4 npn i 4 pnp tranzystorów. Są one zorganizowane w wewnętrznej części powierzchni matrycy w 16 wierszach - po 62 komórki w każdym wierszu - razem 992 komórki. Układy pomocnicze są usytuowane na obrzeżach matrycy. Zależność czasu opóźnienia od prądu 2-kolektorowej bramki podstawowej układu F9480 przedstawia rys. 2.

Rysunek 2 uwzględnia wpływ rozdzielczości techniki litografii /5 μ m i 3 μ m/. Aktualnie Fairchild stosuje technikę 5 μ m, osiągając 6 ns przy 100 μ A.

Ciąg prac projektowych Fairchilda dla matrycy F9480 jest analogiczny do systemu Macrocell Motoroli. Zawiera on niżej wymienione główne etapy:

1. Opracowanie schematu logicznego odbiorcy w logice I³L.
2. Przeprowadzenie symulacji logicznej i generacja programów testowych.
3. Opracowanie wstępne mozaiki komórek i wykonanie ręczne sieci jej połączeń, a następnie digitalizacja szkicu ręcznego.

Alternatywna możliwość realizacji punktu 3: wprowadzenie schematu logicznego do systemu CAD, który automatycznie opracowuje mozaikę komórek i sieć połączeń, współdziałając z projektantem.

4. Wykonanie masek sieci połączeń.
5. Wykonanie sieci połączeń.
6. Montaż i testowanie prototypów.

Firma Ferranti /W. Brytania/ opracowała rodzinę układów matrycowych Uncommitted Logic Array /ULA/, stanowiącą jedno z bar-

dziej dojrzałych opracowań w tej dziedzinie. Prace nad ULA, rozpoczęte w 1972 r. umożliwiły w 1979 r. przedstawienie katalogu [7], [4] z ofertą obejmującą 13 typów matrycowych układów bipolarnych LSI. W układach ULA stosuje się jednowarstwową sieć połączeń. Mają one możliwość współpracy z układami, zarówno bipolarnymi jak i MOS oraz CMOS. ULA są produkowane w trzech seriach: Seria 1000: zawiera matryce liczące od 150 do 286 bramek. Seria 2000: zawiera matryce liczące od 337 do 450 bramek. Seria 5000: zawiera matryce liczące od 726 do 880 bramek.

Matryce ULA mają organizację komórkową. Składają się one z komórek matrycowych oraz peryferyjnych, przy czym obydwa rodzaje komórek realizowane są w konfiguracjach RTL i CML /Resistor Transistor Logic i Current Mode/ w zależności od typu. Poszczególne komórki zawierają zestawy tranzystorów i rezystorów. Czasy opóźnienia na bramkę zawierają się w granicach od 450 ns do 8 ns, a prądy bramki od 4 μ A do 0,56 mA, w zależności od typu układu ULA. Stosuje się napięcie zasilania bramek 5V. Układy są dostarczane w obudowach DIL plastikowych i ceramicznych o ilości wyprowadzeń od 14 do 40. Mogą pracować w temperaturze otoczenia od -55°C do +125°C.

Ferranti wymienia następujące zalety układów matrycowych ULA:

- realizacja połączeń przy pomocy jednej maski,
- jedno napięcie zasilania,
- kompatybilność z TTL i CMOS,
- zalety układów Schottky TTL małej mocy,
- funkcje cyfrowe i liniowe,
- poprawa niezawodności systemów,
- redukcja kosztów montażu systemów,
- redukcja objętości i mocy systemów,
- redukcja kosztów testowania systemów,
- przyspieszenie rozwoju nowych systemów,
- łatwe reguły projektowania dla odbiorcy,
- możliwość baterijnego zasilania /dla typów niskoprądowych/,
- zawartość od 1000 do 6000 niepołączonych elementów.

L i t e r a t u r a :

- [1]. J. Kalisz: Cyfrowe układy scalone w technice systemowej, MON, Warszawa 1977.
- [2]. J. Prioste: A cost - effective approach to bipolar VLSI computer circuits. Computer Design, May 1979.
- [3]. Exar Integrated Systems, Inc., USA Semi-custom IC Design - Programs, Materiały firmowe, 1978.

Występujące w wyniku wprowadzenia układów matrycowych istotne efekty dotyczą:

- zwiększenia swobody konstruktora układów logicznych, nie ograniczonej zestawami układów standardowych,

- skrócenia czasu opracowania nowego typu układu /rzędu np. 12 tygodni/, a w wyniku skrócenia okresu wdrożenia nowego sprzętu o konkurencyjnych parametrach,

- przełamania bariery granicy opłacalności aplikacji nowego typu specjalizowanego układu scalonego i jej obniżenia do poziomu krótkich serii, a w wyniku rozszerzenia zakresu aplikacji układów LSI w sprzęcie elektroniki profesjonalnej,

- zachowania pozostałych efektów związanych z wdrażaniem układów scalonych LSI, takich jak oszczędności obwodów drukowanych, zmniejszenie kosztów montażu, oszczędności na złączach i połączeniach, poprawa niezawodności oraz ok. 10-krotne zwiększenie gęstości upakowania układów elektronicznych, - poprawy wskaźnika szybkości w stosunku do ceny w komputerach.

Z tych względów układy te stanowią przedmiot żywego zainteresowania m. in. producentów sprzętu komputerowego i wykazują silne tendencje do rozwijania produkcji zarówno w firmach przemysłu półprzewodnikowego, jak i w firmach komputerowych.

Szereg wymienionych wyżej źródeł podkreśla wysoką dynamikę rozwojową układów matrycowych w ostatnim okresie. Typowym przykładem jest firma International Microcircuits Incorporated /IMI/ - USA, która informuje, że w 1977 r. produkowała 500-bramkowe układy matrycowe. Natomiast w 1980 r. oferuje układy 2000-bramkowe i zapowiada wkrótce produkcję układów 4000-bramkowych, a na rok 1982 przewiduje produkcję układów 10 000-bramkowych. W warunkach krajowych również należy przewidywać, że układy matrycowe stanowiąby jedno z efektywniejszych rozwiązań w zakresie mikroelektronicznych układów specjalizowanych.

- [4]. P. Forshaw: Designing with the uncommitted logic array. Electronic Engineering, June 1979.
- [5]. MASTERMOS. Design Manual 1974. International Microcircuits, Inc., USA. Materiały firmowe.
- [6]. EXAR-CHIP I²L Design Kit. Instruction Manual Exar Integrated systems, Inc. USA, 1977. Materiały firmowe.

- [7]. ULA Technical Handbook Ferranti Electronics Limited 1979, W. Brytania, Materialy firmowe.
- [8]. Fast CMOS gate arrays match custom circuit speed. Electronic Design 6/March, 15, 79.
- [9]. J. Prioste, R. Rao, W. R. Blood: Functional array eases custom ECL design. Electronics, February 15, 1979.
- [10]. W. D. O'Neil: I^2L Gate Arrays make custom ICS economically feasible. Computer Design, September, 1979.
- [11]. Ultra-High-speed logic - more than just a promise in LSI Electronic Design 7, March 29, 77.
- [12]. W. A. Żukowski, J. K. Kusznew, A. N. Bubiennikow: Bipolarnyje matricznyje BIS - elementno-konstruktiwnaja baza wysokoproizwoditelnych CWM czetwiertogo pokolenia. Zarubieznaja Radioelektronika, 11/79.
- [13]. W. A. Arnold: Gate arrays have marketers raring to go. Electronics, April 27, 1978.
- [14]. J. Pomeranz, R. Nijhuis, Ch. Vicary: Customized metal layers vary standard gate-array chip. Electronics, March 15, 1979.
- [15]. R. Crippen, H. Hingark: Progress. Fairchild Journal of Semiconductor. VI, 8/Nr 2, March/April 1980. 4000 I^3L Gates in Array.
- [16]. J. G. Posa: Gate Arrays. Electronics, September, 1980.
- [17]. R. J. Blumberg, S. Brenrier: A 1500 Gate, Random Logic, Large Scale Integrated /LSI/ Masterslice. IEEE Journal of Solid State Circuits, Vol. SC-14, Nr 5, October, 1979.
- [18]. M. Grossman: Four LSI chips cram more flexibility into IBM computers. Electronic Design 26, December 20, 1979.
- [19]. IBM adopts new gate arrays. Electronics, February 15, 1979.
- [20]. 1980 MOS Product Catalog AMI, USA.



ZALEŻNOŚCI WYDAJNOŚĆ - OBCIĄŻENIE W INTERAKTYWNYCH SYSTEMACH OPERACYJNYCH

Wielka ilość funkcji, jaką muszą realizować interaktywne systemy operacyjne /tj. takie systemy operacyjne, które zostały zaprojektowane pod kątem obsługi dużej liczby użytkowników terminalowych/, powoduje że systemy te mają złożoną strukturę i są bardzo duże, składają się bowiem z od kilkuset tysięcy do kilku milionów instrukcji. Przy tak rozbudowanej i skomplikowanej strukturze zasadniczego znaczenia nabiera problem wydajności systemu. Wydajność systemu jest to jego zdolność do obsłużenia występującego obciążenia [1]. Wydajność jest zatem w przypadku interaktywnych systemów operacyjnych zdolnością do zrealizowania żądań użytkowników terminalowych /abonentów/.

W niniejszym opracowaniu, po przypomnieniu najważniejszych miar obciążenia i wydajności, zostaną przedstawione niektóre zachodzące między nimi zależności, określane ogólnie jako zależności wydajność-obciążenie.

Znajomość tych zależności umożliwia ocenę różnych, stosowanych alternatywnie algorytmów działania systemów operacyjnych, analizę porównawczą różnych systemów, a także śledzenie efektów zmian systemowych /modyfikacje algorytmów, zmiany parametrów/, na wydajność systemu operacyjnego.

Krótką charakterystyka miar wydajności i obciążenia

Celem przypomnienia, przed omówieniem zależności wydajność - obciążenie, zestawione zostały najczęściej stosowane w analizach systemów operacyjnych miary wydajności i obciążenia. Tabela 1 zawiera zestawienie miar wydajności, tabela 2 zestawienie miar obciążenia [2]. W obu tabelach podane są też definicje przytoczonych miar. Miary te występują w przedstawionych w jednym z następujących rozdziałów charakterystykach systemowych.

Tabela 1.

Zestawienie miar wydajności

Lp.	Miara wydajności	Oznaczenie	Opis
1	Przepustowość	-	Ilość pracy użytecznej /jednostek pracy/ zakończonej w jednostce czasu przy danym obciążeniu.
2	Czas odpowiedzi	R	Czas pomiędzy zainicjowaniem akcji, np. wprowadzeniem dyrektywy /komyndy/ terminalowej a uzyskaniem odpowiedzi /na tę dyrektywę/. Czas odpowiedzi nie obejmuje czasów transmisji z terminala i na terminal.
3	Procent czasu w stanie PROBLEM	P_V	Procent czasu przez jaki jednostka centralna wykonuje programy użytkowe.

Nasylenie systemu - poziom wieloprogramowości

Na wydajność systemu operacyjnego mają wpływ także inne czynniki, poza przedstawionymi w tabeli 2. Zaliczają się do nich parametry instalacyjne oraz systemowe parametry sterujące wydajnością [2]. Najważniejsze parametry instalacyjne systemu cyfrowego to wielkość pamięci operacyjnej, szybkość jednostki centralnej, szybkość kanałów. Najważniejszymi parametrami systemu operacyjnego mającymi wpływ na wydajność są: poziom wieloprogramowości oraz wielkość odcinków czasu procesora przydzielanych poszczególnym akcjom.

Poziom wieloprogramowości jest to ilość akcji znajdujących się równocześnie w pamięci operacyjnej i w związku z tym otrzymujących przydział czasu jednostki centralnej /otrzymujących sterowanie procesora/. Poziom wieloprogramowości /ang. multiprogramming level/ w skrócie MPL, bywa też określany jako zbiór wieloprogramowy /ang. multiprogramming set/, lub jako zbiór akcji mogących otrzymać sterowanie procesora /ang. dispatchable set/. MPL ustalany jest przez system operacyjny w funkcji obciążenia i dostępnych zasobów systemu cyfrowego. System operacyjny może dynamicznie zmieniać MPL, reagując na zmianę warunków pracy. Niektóre systemy mają ponadto określony, nieprzekraczalny górny poziom wieloprogramowości. O nasyceniu systemu mówimy wówczas, gdy wy-

korzystanie któregoś z zasobów systemu /np. pamięci operacyjnej, procesora/ osiągnie 100% [3].

O ile powiększenie poziomu wieloprogramowości, zanim system osiągnie stan nasycenia, powoduje wzrost wydajności, to po przekroczeniu progu nasycenia dalsze zwiększenie MPL nie daje już korzyści z punktu widzenia wydajności, a nawet powoduje pogorszenie wskaźników wydajnościowych systemu.

Charakterystyki systemowe

W tym rozdziale przedstawione zostaną najważniejsze /częściej używane/ charakterystyki systemowe. Są to zależności wydajność - obciążenie obserwowane w interaktywnych systemach operacyjnych. Podane też zostaną zwięzłe objaśnienia tych zależności.

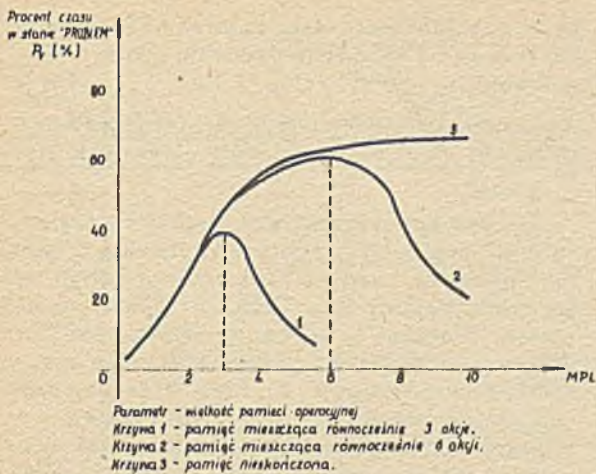
Zależność wykorzystania jednostki centralnej w stanie "problem" od poziomu wieloprogramowości.

Poziom wieloprogramowości /MPL/ jest to ilość akcji dopuszczonych jednocześnie do pamięci operacyjnej, a tym samym otrzymujących czas procesora. Na rys. 1 przedstawiony został wpływ MPL na udział stanu "problem" w pracy jednostki centralnej, przy czym parametrem jest wielkość pamięci operacyjnej. Zależność 1 odnosi się do systemu z pamięcią operacyjną, która może pomieścić równocześnie akcje trzech użytkowników, zależność 2 ilustruje zachowanie się systemu z pamięcią

Tabela 2.

Zestawienie miar obciążenia

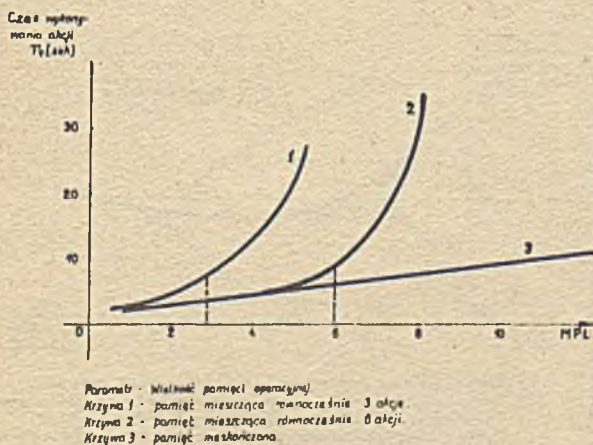
Lp.	Miara obciążenia	Oznaczenie	Opis
1	Liczba równoczesnych użytkowników	N_t	Liczba użytkowników interaktywnych, którzy nawiązali łączność z systemem /podłączonych równocześnie do systemu/
2	Liczba aktywnych użytkowników	N_A	Liczba interaktywnych użytkowników, podłączonych równocześnie do systemu, generujących żądania do systemu
3	Liczba akcji w systemie	N_s	Liczba akcji obsługiwanych oraz czekających w kolejkach na zasoby systemu $N_s = N_d + N_e$
4	Czas odpowiedzi użytkownika	R_u	Czas zużyty przez użytkownika przy terminalu na wygenerowanie nowego żądania
5	Wielkość roboczego zbioru stron	WS lub W	Liczba stron pamięci operacyjnej zarezerwowana dla pojedynczej akcji w celu zapewnienia efektywnego wykonania tej akcji



Rys. 1. Procent czasu jednostki centralnej w stanie PROBLEM w funkcji poziomu wieloprogramowości

wystarczającą na 6 akcji, krzywa 3 odnosi się do sytuacji, gdy pamięć ma wielkość nieskończoną. Widoczny jest szybki spadek wydajności systemu operacyjnego przy zwiększaniu MPI po osiągnięciu nasycenia, tj. w sytuacji gdy mimo braku wolnej pamięci podwyższony jest poziom wieloprogramowości. Przyczyną gwałtownego spadku wydajności jest wzmożony ruch stron pamięci /pomiędzy pamięcią operacyjną a pamięcią zewnętrzną/; zjawisko to określane jest w literaturze anglosaskiej jako thrashing /młócka/.

Jednakże powiększając pamięć operacyjną ponad wielkość mieszczącą równocześnie 6 akcji, uzyskujemy tylko stosunkowo niewielki wzrost wydajności. Zależność 3 obrazuje zachowanie się systemu w sytuacji, gdy nie zachodzi wymiana stron /pamięć nieskończona/. Przy $MPI=6$ obserwujemy zahamowanie wzrostu wydajności, a więc zjawisko odpowiadające nasyceniu systemu. Zahamowanie to uwarunkowane jest konstrukcją systemu operacyjnego i szybkością jednostki centralnej. Przedsta-



Rys. 2. Czas wykonywania akcji w funkcji poziomu wieloprogramowości

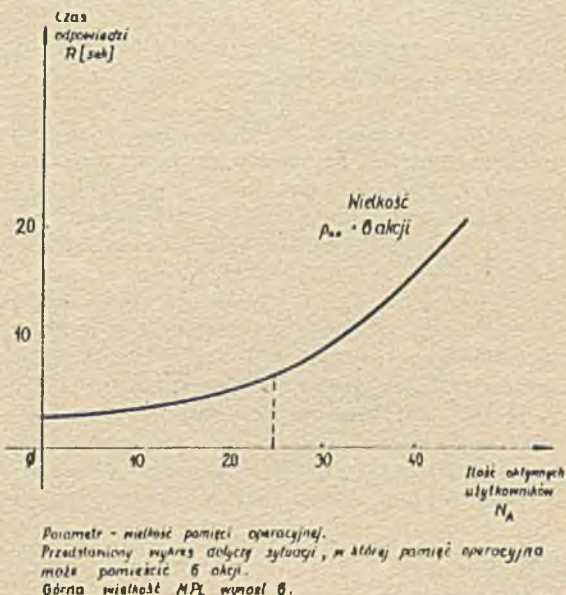
wiona ilustracja odnosi się do systemu operacyjnego VM/370[3]. Dla innych systemów opisane zjawisko może wystąpić przy innym poziomie wieloprogramowości.

Zależność czasu wykonywania akcji od poziomu wieloprogramowości

Wspomniana w tytule paragrafu zależność, przy wielkości pamięci operacyjnej jako parametrze pokazana została na rys. 2. Krzywe 1, 2 i 3 odnoszą się do systemów z pamięcią operacyjną, która może pomieścić odpowiednio 3, 6 i nieograniczoną liczbę akcji. Widoczny jest szybki wzrost czasów wykonywania /a więc spadek wydajności/, w przypadku zwiększania MPI ponad wartość, przy której występuje nasycenie, to znaczy powyżej 3 i 6 odpowiednio dla krzywych 1 i 2.

Zależność pomiędzy czasem odpowiedzi a ilością aktywnych użytkowników.

Badania wykazały [3], że w systemie VM/370 średnio jedna czwarta aktywnych użytkowników posiada jednocześnie akcje w realizacji /reszta użytkowników zajęta jest przygotowaniem dyrektyw na swoich termina-



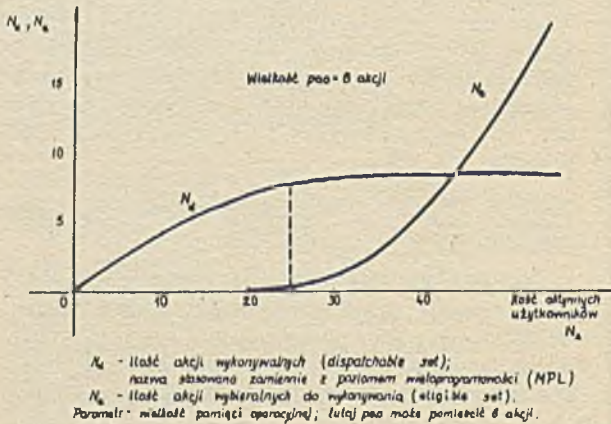
Rys. 3. Czas odpowiedzi w funkcji ilości aktywnych użytkowników

lach/. Zatem przy ilości aktywnych użytkowników nie przekraczającej 24, ilość akcji w stanie realizacji nie przekracza 6, co przy założeniu że pamięć może je pomieścić daje dobre czasy odpowiedzi. Po osiągnięciu nasycenia /6 akcji równocześnie w pamięci/, dodatkowi abonenci powodują znaczny wzrost czasów odpowiedzi w systemie. Wzrost ten spowodowany jest faktem, że akcje użytkowników nie są teraz bezpośrednio przyjmowane do pamięci operacyjnej /gdyż ta może przyjąć maksymalnie 6 akcji/, lecz muszą czekać na obsłużenie

w zbiorze akcji wybieralnych w pamięci zewnętrznej.

Zależność pomiędzy MPL oraz ilością akcji wybieralnych a ilością aktywnych użytkowników.

Wymienione zależności ilustruje rys. 4. Zgodnie z wcześniej przytoczoną regułą, średnio jedna czwarta aktywnych użytkowników posiada akcje w realizacji /dotyczy to systemu



Rys. 4. Ilość akcji wykonywalnych N_d / i ilość akcji wybieralnych N_e w funkcji ilości aktywnych użytkowników

VM/370/. Do momentu nasycenia akcje te przyjmowane są do zbioru akcji wykonywalnych /dispatchable set/. Po osiągnięciu nasycenia narasta kolejka akcji czekających na wybranie do wykonywania /eligible set/. Konsekwencją tego jest znaczny wzrost czasów odpowiedzi w systemie. W systemie operacyjnym, do którego ilustrację stanowi rys. 4 przyjęto zasadę ograniczenia MPL /tj. ilości akcji wykonywalnych/, stosując jako kryterium ilość dostępnej pamięci operacyjnej. Oczywiście można byłoby zwiększać poziom wieloprogramowości wraz z przyrostem ilości aktywnych użytkowników. Wtedy jednak należałoby liczyć

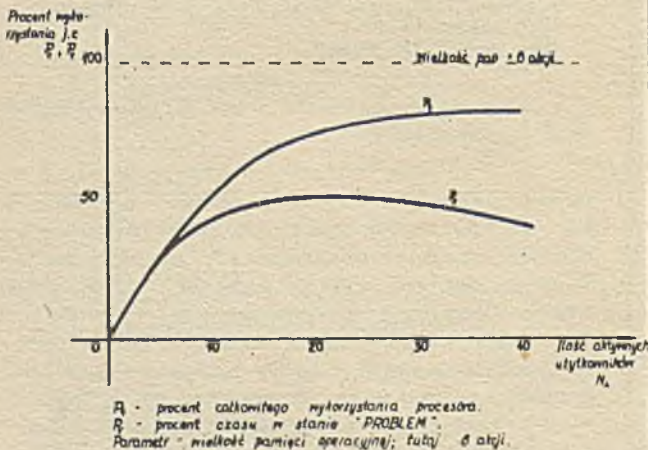
się z szybkim wzrostem czasów wykonywania akcji oraz ze spadkiem procentu czasu j. c. w stanie "problem" /rys. 1 i 2/.

Zależność pomiędzy stopniem wykorzystania jednostki centralnej a ilością aktywnych użytkowników.

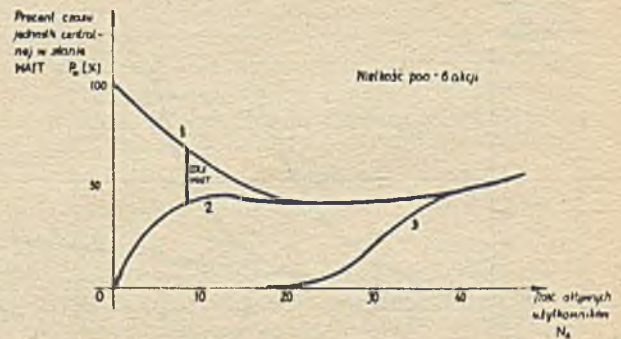
Stopień wykorzystania jednostki centralnej wyrazić można przez procent czasu j. c. w stanie "problem" oraz przez procent całkowitego wykorzystania j. c. /suma procentu czasu w stanie "problem" i w stanie "supervisor"/. Odpowiednie wykresy przedstawiono na rys. 5, dotycząc one systemu VM/370. Widoczny spadek udziału stanu "problem" jest efektem zilustrowanego później /patrz: rozdział "Zależność ilości przerw typu "brak strony" od ilości aktywnych użytkowników" - wzrostu ilości odwołań do podsystemu paginacji /tj. przerw typu "brak strony"/, po osiągnięciu nasycenia. Szczególnie duży wzrost ilości przerw "brak strony" i związany z tym spadek wykorzystania j. c. następuje w przypadku, gdy system operacyjny przewidzi zbyt małe robocze zbiory stron /working sets/ dla poszczególnych akcji, a więc dopuści zbyt wysoki MPL w stosunku do posiadanej pamięci operacyjnej. Narzut systemowy spowodowany stronicowaniem /ang. paging overhead/ drastycznie obniża wydajność systemu.

Zależność pomiędzy procentem czasu j. c. w stanie "wait" a ilością aktywnych użytkowników.

Odpowiednie wykresy widoczne są na rys. 6. Przy małym obciążeniu charakterystyczny jest wysoki procent stanu "wait" w czasie procesora, przy czym dominującą rolę odgrywa tu stan "idle wait" występujący, gdy brak akcji gotowej do przetwarzania oraz gdy nie ma akcji oczekującej na zakończenie operacji we/wy. Jest to spowodowane zdolnością procesora do szybkiego obsłużenia akcji pochodzących od niewielkiej liczby użytkowników. Przy

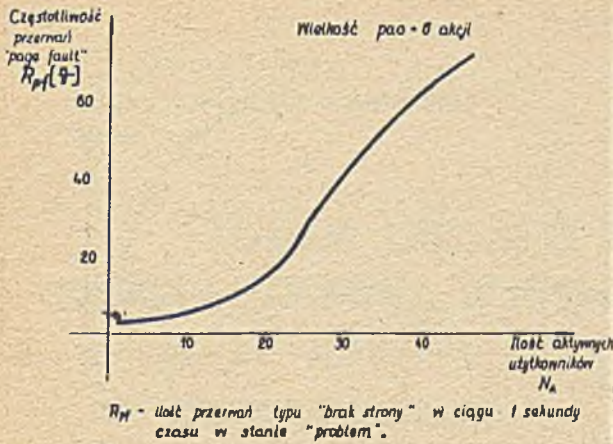


Rys. 5. Wykorzystanie jednostki centralnej w funkcji ilości aktywnych użytkowników



- 1 - Całkowity (sumaryczny) procent czasu j. c. w stanie WAIT (TOTAL WAIT)
- 2 - Procent czasu w stanie WAIT spowodowany oczekiwaniami na zakończenie operacji mycia/mycia (IO WAIT)
- 3 - Procent czasu w stanie WAIT spowodowany oczekiwaniami na sprawdzenie stron pamięci (PAGE WAIT).

Rys. 6. Procent czasu jednostki centralnej w stanie WAIT w funkcji obciążenia /ilości aktywnych użytkowników/.



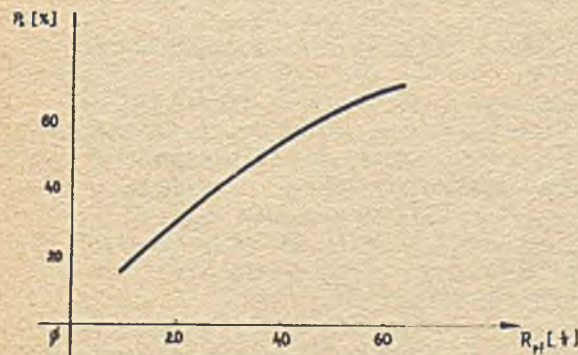
Rys. 7. Częstotliwość przerwań "brak strony" w funkcji ilości aktywnych użytkowników

wzroście obciążenia udział stanu "idle wait" maleje, dominuje zaś stan czekania na zakończenie operacji we/wy pochodzących od akcji użytkowników /I/O wait/.

Po przekroczeniu nasycenia rośnie szybko czas oczekiwania na sprowadzenie stron pamięci operacyjnej /page wait/. Wzrost ten może doprowadzić do poważnego zmniejszenia efektywności działania systemu cyfrowego. Dlatego, aby zmniejszyć czas dostępu do stron przechowywanych w pamięci zewnętrznej, w dużych systemach z pamięcią wirtualną powinny być stosowane urządzenia z głowicami stałymi, także jak bębny magnetyczne lub dyski stałe.

Zależność ilości przerwań typu "brak strony" od ilości aktywnych użytkowników.

Ilustrację stanowi rys. 7. Widoczny jest wyraźny wzrost ilości przerwań "brak strony" /"page fault"/ po osiągnięciu nasycenia. Powodem tego zjawiska może być przyjęcie zbyt wysokiego poziomu wieloprogramowości, wspomniane w rozdziale "Zależność pomiędzy stop-



P_s - narzut systemowy.
 R_{pf} - częstotliwość przerwań "brak strony".

Rys. 8. Procent czasu w stanie SUPERVISOR /narzut systemowy/ w funkcji częstotliwości przerwań "brak strony"

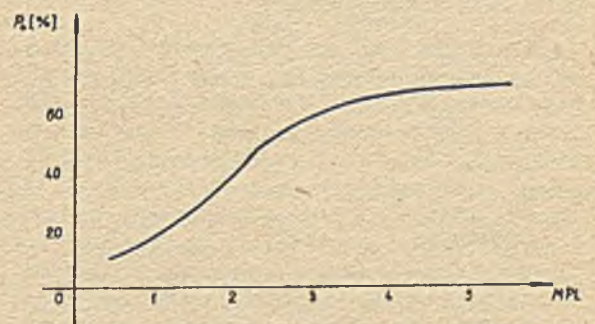
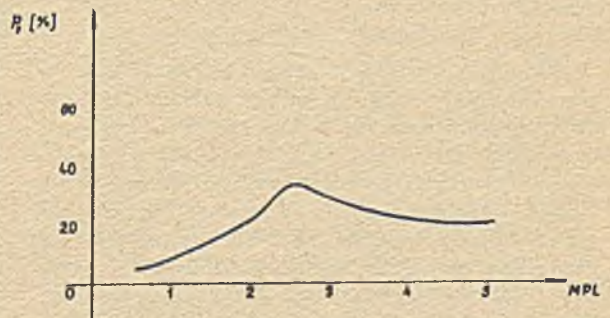
niem wykorzystania jednostki centralnej a ilością aktywnych użytkowników.

Zależność narzutu systemowego od ilości przerwań "brak strony".

Z rysunku 8 można ocenić znaczny wpływ ilości przerwań typu "brak strony" na efektywność wykorzystania jednostki centralnej. Już przy częstotliwości 40 przerwań na sekundę narzut systemowy osiąga 60% czasu jednostki centralnej. Podkreśla to zasadniczy wpływ właściwej oceny wielkości zbiorów roboczych poszczególnych akcji na wydajność systemu, a więc wagę problemu ustalania odpowiedniego poziomu wieloprogramowości.

Zależności narzutu systemowego oraz wykorzystania procesora w stanie "problem" od poziomu wieloprogramowości.

Przedstawione na rysunkach 9 i 10 wykresy odnoszą się do systemu TSS/360, w odróżnieniu od zbliżonych znaczeniowo zależności przedstawionych w rozdziale "Zależność pomiędzy stopniem wykorzystania jednostki centralnej a ilością aktywnych użytkowników, które ilustrują działanie systemu operacyjnego VM/370. Ze wzrostem poziomu wieloprogramowości procent czasu pracy j.c. w stanie "problem" wzrasta, lecz po osiągnięciu $MPL = 3$ zaczyna maleć. Natomiast stale rośnie narzut systemowy czyli procent czasu procesora w stanie "supervisor". Jest to spowodowane



P_p - procent czasu jednostki centralnej w stanie "PROBLEM".
 P_s - procent czasu jednostki centralnej w stanie "SUPERVISOR"
MPL - poziom wieloprogramowości.

Rys. 9 i 10. Wykorzystanie jednostki centralnej w stanie PROBLEM oraz wielkości narzutu systemowego w funkcji poziomu wieloprogramowości

wzrostem ilości odwołań do stron będących poza pamięcią operacyjną. Po przekroczeniu $MPL = 4$ narzut systemowy stabilizuje się na poziomie 60% przy niskim procencie wykorzystania procesora w stanie "problem". Przez resztę czasu procesor znajduje się w stanie oczekiwania. Ilustruje to przekroczenie stanu nasycenia pamięci, tj. sytuację w której wszystkie akcje oczekują na sprowadzenie stron do pamięci operacyjnej.

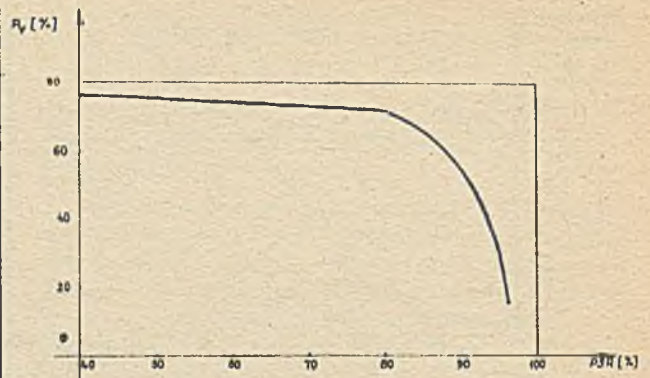
Zależność wykorzystania procesora w stanie "problem" od współczynnika odbierania stron /ang. page steal ratio/.

Zależność tę ilustruje rys. 11. Charakterystyczną cechą jest tutaj gwałtowny spadek wydajności systemu po przekroczeniu przez współczynnik odbierania stron wartości 80%. Powodem spadku wydajności jest oczywiście wzmoczona wymiana stron pamięci operacyjnej, zjawisko znane jako "thrashing". Na ogół systemy operacyjne starają się utrzymać pewien zapas wolnych stron pamięci operacyjnej przy określonym MPL. Gdy na skutek załadowania przez system szacunku wielkości zbiorów roboczych dla poszczególnych akcji, ilość dostępnych stron pamięci zmaleje w pobliżu zera, współczynnik odbierania stron znacznie wzrasta. Oczywiście przyjęcie przez system a priori zbyt wysokiego MPL da taki sam efekt. Rysunek 11 ilustruje pracę systemu operacyjnego CP 67, poprzednika systemu VM/370 [5].

Rola modelowania

Przedstawione w poprzednim rozdziale charakterystyki są użyteczne zarówno na etapie projektowania jak i podczas dostrajania systemów operacyjnych. Na etapach projektowania i tworzenia systemów operacyjnych stosowane jest szeroko modelowanie zachowania się systemów w zależności od obciążenia, parametrów wewnętrznych oraz wielkości zasobów. Zarówno modelowanie symulacyjne jak i analityczne będą omówione w następnych opracowaniach tego cyklu.

Model - analityczny lub symulacyjny - dostarcza danych pozwalających na uzyskanie szeregu charakterystyk istotnych z punktu widzenia funkcjonalności i wydajności systemu operacyjnego. Zmiana parametrów wejściowych /danych wejściowych/ zadanych modelowi umożliwia szybkie i wygodne uzyskanie charakterystyk systemu w zadanych warunkach, bez potrzeby długotrwałych pomiarów. Podobnie ważną rolę spełniają modele systemów już istniejących, służące do długiego i często permanentnego procesu strojenia i poprawiania gotowego i używanego systemu. Zmiany w systemie rzeczywistym poprzedzane są wtedy zmianami w modelu, po czym następuje uruchomienie modelu dla uzyskania charakterystyk. Jeśli próby modelowe wykazują poprawę działania sys-



P_p - procent czasu jednostki centralnej w stanie "problem"
PSR - współczynnik odbierania stron (page steal ratio).

Rys. 11. Wykorzystanie jednostki centralnej w stanie "problem" w funkcji współczynnika odbierania stron /page steal ratio PSR/

temu, wówczas zmiany wprowadzane są do algorytmów i /lub/ parametrów systemu rzeczywistego.

Przedstawione zależności, chociaż oparte o systemy operacyjne i modele systemów firmy IBM /TSS360, CP67, VM370/, można odnieść ogólnie do wszystkich interaktywnych systemów operacyjnych. Opisane charakterystyki należą do najistotniejszych przy analizie wydajności systemu operacyjnego. Uwidaczniają one ograniczenia tak funkcjonalne jak i ilościowe, jakie są narzucane przez obciążenie i dostępne zasoby systemu. Wartości przedstawione na wykresach związane są z konkretnymi realizacjami systemów operacyjnych i nie mogą być przenoszone bezpośrednio na inne systemy, natomiast ogólne cechy zależności i ograniczeń są symptomatyczne dla wszystkich interaktywnych systemów operacyjnych.

L i t e r a t u r a :

- [1] I. Svobodova: "Computer Performance Measurement", Elsevier Publishing Company, New York, Oxford, Amsterdam 1976.
- [2] J. Kierkowski: "Miernictwo interaktywnych systemów operacyjnych - wprowadzenie", Biuletyn Techniczny "Mera" nr 10/1980.
- [3] Y. Bard: "Performance analysis of virtual memory time - sharing systems", IBM Systems Journal, nr 4, 1975.
- [4] H. A. Anderson jr., M. Reiser, G.L. Galati: "Tuning a virtual storage system", IBM Systems Journal, nr 3, 1975.
- [5] Y. Bard: "Application of the Page Survival Index to virtual memory system performance", IBM Journal of Research and Development, May, 1975.



PEWNE UWAGI O NOWYCH JĘZYKACH PROGRAMOWANIA WYSOKIEGO POZIOMU: LOGLAN I ADA

/Część II

Dynamiczny rozwój architektury komputerów i prace nad programowaniem spowodowały powstanie wielu ciekawych języków programowania wysokiego poziomu. Na drodze rozwoju języków programowania szczególnie ważne okazało się zaprojektowanie języków SIMULA 67 /Dahl, Myhrhaug, Nygaard/ i PASCAL /Wirth/.

W pierwszym z tych języków przedstawiono nowe podejście do jednostek dynamicznych bloków i wprowadzono klasy oraz klasy prefiksowane. Prefiksowanie jest bezsprzecznie bardzo cennym narzędziem programisty, ale trudności w jego implementacji powodują, że często projektanci języków programowania rezygnują z wprowadzania go do języka.

Język PASCAL ma rozbudowaną strukturę typów i instrukcji, przy jego projektowaniu /koniec lat sześćdziesiątych/ podstawowym kryterium była efektywność implementacji. W tym języku nie wprowadzono prefiksowania - nie znaczy to, że "moc" języka jest mniejsza niż np. SIMULI; algorytmy, które mogą być zapisane przy pomocy prefiksowania, mogą być również zapisane przy pomocy struktur języka PASCAL.

Oba języki zdobywają sobie rosnącą popularność, zarówno w Europie jak i w USA.

Strukturalność programowania, efektywność implementacji w nowych warunkach rozwiniętej architektury komputerów oraz potrzeba wprowadzenia obliczeń równoległych spowodowały, że w ostatnich latach powstało kilka nowych języków, np. CONCURRENT PASCAL /Brinch-Hansen/, MODULA /Wirth/. Oba wymienione języki są zaprojektowane na bazie PASCALA.

W niniejszym artykule chcemy zwrócić uwagę czytelnika na powstanie dwu nowych języków wysokiego poziomu - LOGLAN i ADA. Na początku podamy krótką historię powstawania tych języków. Prace nad językiem LOGLAN^{x/} zostały podjęte przez zespół pracowników Instytutu Maszyn Matematycznych "Mera"

i Instytutu Informatyki Uniwersytetu Warszawskiego około 1975 roku, zgodnie z planem badań naukowych Instytutu Maszyn Matematycznych. Prace analityczne i koncepcyjne były prowadzone pod kierownictwem A. Salwickiego /do którego należy idea stworzenia języka LOGLAN/ w Zakładzie Studiów i Analiz Instytutu Maszyn i podczas zajęć seminaryjnych i wykładów monograficznych w Uniwersytecie Warszawskim. Prace zakończyły się wspólnym raportem, w którym przedstawiono ogólną koncepcję języka /Archiwum Opracowań IMM, lipiec 1977, A. Salwicki, T. Müldner, H. Oktaba, W. Ratajczak/.

Od połowy 1976 roku do chwili obecnej prace nad projektowaniem i implementacją języka LOGLAN na minikomputer MERA 400 są prowadzone w Instytucie Informatyki UW przez zespół, od połowy 1978 roku kierowany przez A. Kreczmara, w składzie: P. Findelsen, P. Gburzyński, M. Lao, A. Litwiniuk, T. Müldner /IMM "Mera"/, H. Oktaba, W. Ratajczak, A. Salwicki /IMPAN/, D. Szczepańska-Wasersztrum. Współpraca między Zjednoczeniem "Mera" a Instytutem Informatyki UW rozpoczęła się w 1977 roku i jest kontynuowana zgodnie z umową między tymi instytucjami.

Rozwój języka LOGLAN był możliwy również dzięki życzliwości i zainteresowaniu wielu osób i instytucji, które nie brały bezpośredniego udziału w pracach nad samym językiem, w tym Zjednoczenia "Mera" i jego Dyrektora Naukowego, prof. A. Janickiego, który od początku przewidywał przyszły rozwój obliczeń wieloprocessorowych i stymulował badania w tym kierunku. Zjednoczenie "Mera" dostarczyło środków obliczeniowych i finansowych na prowadzenie badań dotyczących tematu: "Bazowy język programowania LOGLAN".

Język ADA został zaprojektowany na zlecenie Departamentu Obrony Stanów Zjednoczonych. Prace rozpoczęły się w 1974 roku sformułowaniem zbioru wymagań, jakie powinien

^{x/} Ustęp dotyczy historii LOGLAN-u; jest cytatem z raportu tego języka.

spełniać język programowania wysokiego poziomu. Kolejne wersje tego zbioru wymagały powstawały średnio co rok, a ostateczna/pod nazwą Steelman/ była podstawą do rozpisania pewnego rodzaju konkursu na zaprojektowanie języka. Przy pomocy kolejnych eliminacji wybrano dwie wersje - Red /propozycja firmy Intermetrics/ oraz Green /propozycja firmy CII Honeywell Bull/. Na podstawie opinii wielu znanych naukowców i ośrodków z USA i innych krajów NATO zdecydowano wybrać język Green, któremu nadano nazwę ADA. W połowie 1979 roku ogłoszono drukiem pierwszą wersję języka. Przyjęto, że ta wersja będzie ustalona przez rok, a w tym czasie zbierano uwagi z wielu ośrodków, którym zlecono programowanie i implementowanie tego języka. W połowie 1980 roku opublikowano kolejną wersję, która zawiera wiele drobnych zmian w stosunku do poprzedniej i która będzie znowu ustalona przez najbliższy rok.

LOGLAN jest zaprojektowany na bazie SIMULI 67, z uogólnieniem na prefiksowanie wielopoziomowe, obliczenia równoległe i oddzielną kompilację. ADA jest zaprojektowana na bazie języka PASCAL, z różnymi modyfikacjami, m.in. obliczeniami równoległymi i oddzielną kompilacją.

Oczywiście, w tak krótkim opracowaniu, pominięto wiele istotnych i ciekawych cech obu języków /m.in. oddzielną kompilację/. Mamy jednak nadzieję, że czytelnik zorientuje się w podstawowych konstrukcjach języków LOGLAN i ADA, pełne opisy znajdują się w następujących pozycjach:

- Język programowania LOGLAN, wewnętrzny raport.
- Sigplan Notices, Vol. 14, N. 6, June 1979
Part A: Preliminary ADA Reference Manual
Part B: Rationale for the Design of the ADA Programming Language.
- Reference Manual for the ADA Programming Language.
Proposed Standard Document. United States Department of Defense, July 1980.

JEDNOSTKI PROGRAMOWE

Wprowadzenie jednostek programowych umożliwia strukturalizację programu. Zależności statyczne /tekstowe/ i dynamiczne między nimi określają wykonanie programu.

LOGLAN: Rozróżnia się podprogramy, bloki, klasy, współprogramy oraz procesy.

• Podprogramy

Podprogram tworzy opis pewnego zadania. Wprowadza się dwa rodzaje podprogramów: funkcje i procedury. Podprogram składa się z lokalnej struktury danych /i operacji/, oraz listy instrukcji. W instrukcjach, oprócz lokalnych danych, mogą występować globalne dane, tj. dane zdefiniowane w otoczeniu, zgodnie z regułami widoczności /por. część II/. Interakcje między podprogramem a jego otoczeniem są realizowane przy pomocy paramet-
 trów /argumentów podprogramu/ oraz zmiennych globalnych. Procedury opisują wykonanie pewnych akcji. funkcje dostarczają zawsze pewnej wartości. Wykonanie podprogramu /tzn. listy jego instrukcji/ następuje wskutek jego wywołania, w którym podaje się nazwę podprogramu i argumenty, na których ma operować - parametry aktualne.

Przykład:
 type dzielenie z reszta: procedure
 (a, b: integer; output q, r: integer);
 r:=a;
 while r>=b do q:=q+1; r:=r-b od
 fin

• Bloki

Blok /rozumiany jak w języku ALGOL/ jest opisem lokalnej struktury danych i operacji oraz pewnych akcji. Blok jest instrukcją, tzn. jego wystąpienie powoduje utworzenie jednostki dynamicznej bloku i jej wykonanie. Obiekty bloków i podprogramów są anonimowe, tzn. programista nie ma dostępu do struktury tych obiektów, spoza obiektów. Po zakończeniu wykonywania instrukcji, obiekty są automatycznie dealokowane /usuwane z pamięci/.

• Klasy

Pojęcie klasy jest najważniejszym pojęciem LOGLANu /zapożyczonym z SIMULI 67/. Klasa jest, podobnie jak podprogram, wzorcem wg którego tworzy się obiekty. Utworzenie obiektu następuje wskutek wykonania instrukcji generacji. Do obiektów można się odwoływać przy pomocy zmiennych referencyjnych. Raz rozpoczęte wykonywanie instrukcji klasy kończy się /po wykonaniu wszystkich instrukcji/ wygenerowaniem obiektu i nie może być wznowione. Jednakże obiekty klasy nie są w takiej sytuacji automatycznie dealokowane i istnieją dopóty, dopóki nie nastąpi bądź systemowa dealokacja /przez program garbage collector, w sytuacji gdy nie ma referencji do tych obiektów/, bądź programowa dealokacja /przy pomocy instrukcji kill/. Jednocześnie może istnieć wiele obiektów jednej klasy.

Przykład:

```
type zespolone: class (re, im: real);
type iloczyn: function (z: zespolone): zespolone;
    result := new zespolone (re+z.re,
    im+z.im)
    fin,
var d: real; d:= re*re + im*im;
    if d>0 then d:= sqrt(d) } re:=re/d;
    im:=im/d fi
    fin
```

Pojęcia prefiksowania klas /i bloków/ a nawet funkcji i procedur, wielkości wirtualnych oraz ograniczenia widoczności /przez listy wielkości schowanych, chronionych i dziedziczonych z prefiksu/ są uogólnieniem /m.in. na prefiksowanie wielopoziomowe/ pojęć wprowadzonych w języku SIMULA 67 i pozwalają na szerokie zastosowania języka.

Niech B będzie klasą postaci

```
type B: class;  
    lista deklaracji B;  
    lista instrukcji B  
fin
```

Wówczas nazwa B może być użyta jako prefiks dla klasy C:

```
type C: B class;  
    lista deklaracji C;  
    lista instrukcji C  
fin
```

Klasa C jest w istocie rozszerzeniem klasy B, tzn. obiekt klasy C powstaje przez połączenie w zwartą całość pól danych klasy B i klasy C, natomiast instrukcje klasy C powstają z połączenia listy instrukcji I klasy B oraz listy J klasy C /przez wstawienie listy J w miejsce wyróżnionej instrukcji inner występującej na liście I/.

Zauważmy, że prefiksowanie jest narzędziem istotnie różnym od zagnieżdżenia tekstowego. Z drugiej strony prefiksowanie może być zaimplementowane w taki sposób, że nie wymaga wstępnego przetworzenia /nie jest makro-definicją/.

Współprogramy /coroutines/.

Współprogram jest szczególnym przypadkiem klasy i może być w trzech stanach - przyłączony /aktywny, jego instrukcje są bieżąco wykonywane/, odłączony /wstrzymany/ i wygenerowany. Dzięki temu wykonywanie zadań może być wielokrotnie wstrzymywane i wznowiane.

● Procesy

Procesy są również odmianą klasy - ale mogą być wykonywane równolegle. Synchronizację procesów umożliwiają pojęcia pierwotne, oraz zbudowane przy ich pomocy pojęcia złożone, np. monitory.

ADA: Wyróżnia się podprogramy, bloki i moduły.

Podprogramy i bloki.

Znaczenie jak w LOGLANie.

Przykład:

```
function dot_product (X, Y: vector) return  
real is  
    sum:real:=0.0;  
    begin  
        for J in X'range loop  
            sum:=sum+X(J)*Y(J); end loop;  
        return sum;  
    end dot_product
```

● Moduły

Wyróżnia się dwa rodzaje modułów: pakiety i zadania. Pakiet definiuje zbiór jednostek - danych, typów i podprogramów. Widoczność i dostęp do, oraz z, pakietu może być ograniczona, co umożliwia np. ograniczenie dostępu do własności logicznych wyrażonych przez dany pakiet. W pakiecie mogą również występować instrukcje, które służą inicjalizacji danych.

Nagłówek podprogramu i pakietu może być deklarowany oddzielnie. Programista ma dostęp wyłącznie do nazw zadeklarowanych w nagłówku.

Przykład:

```
package rational_numbers is  
    type rational is  
        record  
            numerator: integer;  
            denominator:  
                integer range 1..integer'last;  
        end record;  
    function equal (X, Y:rational) return boolean;  
end
```

```
package body rational_number is  
    procedure same_denominator (X, Y:in out  
rational) is  
        ....  
    end;  
    function equal (X, Y:rational) return boolean  
is  
    U, V:rational;  
    begin  
        U:=X; V:=Y; same_denominator (U, V);  
        return U.numerator = V.numerator;  
    end;  
end rational_numbers
```

Zadania mogą być wykonywane równolegle. Synchronizację zadań umożliwia pojęcie instrukcji wejściowych /entries/, przypominających procedury i tzw. rendezvous, w czasie którego wykonuje się treść instrukcji we wzajemnym wykluczaniu z innymi zadaniami. W "życiorysie" zadania wyróżnia się trzy fazy: aktywne, pasywne i wstrzymane. Po aktywacji zadania wykonywane są jego instrukcje. Pasywacja następuje wskutek zakończenia, natomiast wstrzymanie zadania jest skutkiem synchronizacji.

TYPY DANYCH

Typ danych T wyznacza zbiór wartości |T|, jakie te dane /stałe lub zmienne/ mogą przyjmować. Elementy zbioru |T| będziemy nazywać obiektami typu T.

LOGLAN: Rozróżnia się dwa rodzaje typów:

● Typy pierwotne /skalarne/

Systemowo definiuje się typy boolean, integer, real, character i string.

● Typy złożone

Obiekty typów złożonych są obiektami strukturalnymi, złożonymi ze składowych. Składowe tablice /= obiektu typu tablicowego/ są zmiennymi tego samego typu i można się do nich odwoływać przez indeksowanie.

ADA: Rozróżnia się cztery rodzaje typów danych:

● Typy skalarne

Dla obliczeń numerycznych wprowadza się: typy stałopozycyjny /z absolutnym ograniczeniem błędów/ i zmiennopozycyjny /ze względnym ograniczeniem błędów/.

Systemowo definiuje się typy integer i duration /ten ostatni dla operacji związanych z czasem/. Oprócz tego wprowadza się typ wyliczeniowy /jak w PASCALu/ wyznaczający uporządkowane zbiory dyskretnych wartości. Systemowo definiuje się typy character i boolean.

● Typy złożone

Wprowadza się typ tablicowy i rekordu /dopuszcza się rekordy z wariantami/.

• Typy dostępu /access type/.
Wartości typu dostępu są referencjami do dynamicznie tworzonych obiektów strukturalnych.

• Typy prywatne.

Typ prywatny jest to typ zdefiniowany w pakiecie, użytkownik ma dostęp tylko do jego nazwy i pewnych operacji na nim, natomiast szczegóły implementacyjne są schowane.

Dla każdego typu można wprowadzić jego podtyp. Zbiór wartości $\{T_i\}$ podtypu T , typu T jest zawarty w zbiorze $\{T\}$, tak więc pojęcie podtypu pozwala na wprowadzenie podzbiorów obiektów skalarnych, tablic z ograniczonymi wartościami indeksów i rekordów z szczególnym wariantem.

DANE /STAŁE I ZMIENNE/.

LOGLAN:

Stałe są typu pierwotnego, ich wartość początkowa jest wyznaczona /w czasie kompilacji/ przez wartość wyrażenia występującego w deklaracji. Wartość stałej nie może ulec zmianie. Zmienne mogą być dowolnego typu, ich wartości początkowe są standardowo wyznaczone przez typ, np. 0.0 dla typu real.

Przykłady:

```
var x:real, const pi=3.14.
```

ADA:

Wprowadza się stałe liczbowe i zmienne o znaczeniu jak podano wyżej. Wartość początkowa zmiennej może być wyznaczona przez wyrażenie występujące w deklaracji. Dodatkowo wprowadza się stałe dowolnego typu, których wartość początkowa może nie być statycznie określona. Zauważmy, że stałe są w różny sposób traktowane przez kompilator: stałe liczbowe mogą być zastąpione przez odpowiednie wartości, natomiast dowolne stałe są traktowane jako zmienne, których wartość może być tylko odczytywana /jednakże ich istnienie ma znaczenie przy projektowaniu programu/.

Przykłady:

```
x:real:=3; pi:constant := 3.14; limit:constant  
coefficient:=disp (1.1)
```

TABLICE:

LOGLAN:

W LOGLANie tablica jest rodzajem klasy, tzn. jednostką syntaktyczną z programowalną alokacją i dealokacją. Typu tablicowego nie deklaruje się *explicitie*, może jedynie występować w deklaracjach zmiennej, np.

```
var X : array of array of T.
```

Deklaracja zmiennej typu tablicowego ustala wymiar tablicy i typ elementów, natomiast pary graniczne mogą pozostać nie ustalone. Ustala je dopiero instrukcja generacji tablicy, np. array X dim (a:b, c:d).

Pary graniczne są dynamiczne w tym sensie, że wyrażenia je określające mogą nie być określone w czasie kompilacji.

ADA:

Typ tablicowy deklaruje się *explicitie* ustalając wymiar, typ indeksów oraz typ elementów.

Pary graniczne mogą pozostać nie ustalone, np.

```
type schedule is array (day) of boolean;  
type day is array (1..10) of integer;  
type matrix is array (integer range< >)  
of real.
```

REKORDY

ADA:

Rozróżnia się trzy rodzaje rekordów: zwykłe, z deskryminantem, oraz z wariantami. Elaboracja deklaracji rekordu pierwszego rodzaju powoduje utworzenie pola danych dla składowych. Wszystkie zmienne tego typu /rekord/ mają te same składowe.

Przykład:

```
type date is record  
    day: integer range 1..31;  
    month: integer range 1..12;  
end record
```

W przypadku rekordu z deskryminantem różne zmienne tego typu mogą mieć różne wartości, zależnie od wartości deskryminantów. Deskryminant jest składową /stałą/ typu dyskretnego. W deklaracji

```
type buffer (size:integer range 0..max  
:=100) is  
record  
    pos:integer range 0..max:=0;  
    value:string (1..size);  
end record
```

size jest deskryminantem z "proponowaną" wartością 100. Przy deklaracji zmiennej, postaci

```
message:buffer  
deskryminant size jest równy 100, ale przy  
deklaracji large:buffer (200)  
wartość size jest równa 200.  
Deskryminant występuje w rekordzie z wariantami, jego wartość wyznacza postać rekordu.  
Na przykład dla deklaracji  
type device is (printer, disk);  
type periph (unit:device) is  
record  
    case unit is  
        when printer => line_count:integer  
            range 1..page_size;  
        when disk => cylinder:cylinder_index;  
    end case  
end record
```

składowa unit jest deskryminantem. W deklaracji zmiennych typu periph, wartość tego deskryminantu wyznacza bieżący wariant rekordu:
writer:periph (unit => printer)
reader :periph (unit => disk).

TRANSMISJA PARAMETRÓW.

Zmienne parametry.

LOGLAN:

Wprowadza się dwa sposoby transmisji - wejściowy /odpowiadający podstawieniu przez wartość z ALGOLu/ oraz wyjściowy. W pierwszym przypadku na wejściu do treści jednostki wartość parametru aktualnego wyznacza wartość początkową parametru formalnego, w drugim przypadku na wyjściu z jednostki wartość parametru formalnego wyznacza wartość parametru aktualnego.

ADA:

Oprócz powyżej podanych sposobów transmisji wprowadza się trzeci - wejściowo/wyjściowy. Podprogramy i typy jako parametry.

LOGLAN:

Dopuszcza się podprogramy i typy jako parametry.

ADA:

Typy i podprogramy nie mogą być parametrami formalnymi. Natomiast w podprogramie i pakiecie można użyć tzw. generycznych parametrów formalnych - podprogramów lub typów. Moduły z generycznymi parametrami formalnymi nazywają się jednostkami generycznymi i są wzorcem do budowy jednostek programowych. Jako takie nie mogą być bezpośrednio użyte. Instancje wzorca tworzy się przy pomocy tzw. inicjalizacji generycznej, otrzymane podprogramy i pakiety są zwykłymi jednostkami programowymi i mogą być używane w zwykły sposób.

INSTRUKCJE

Wystąpienie instrukcji powoduje wykonanie pewnych akcji i zmianę wartościowania /stanu pamięci/.

Instrukcje atomowe.

● LOGLAN:

Wyróżnia się instrukcję podstawienia nadającą nową wartość ustalonej zmiennej, /ma ona jednolitą postać dla "zwykłych" zmiennych oraz dla zmiennych referencyjnych, w tym ostatnim przypadku, wykonanie instrukcji powoduje podstawienie referencji/. Wprowadza się instrukcję kopiowania dla zmiennych referencyjnych /wykonanie powoduje podstawienie kopii wartości, danej przez referencję/, in-

strukcje wywołania procedury, generacji /tablicy i obiektu/, sterowania /wyjścia z pętli, zakończenia - bloku, podprogramu lub obiektu/. Ponadto wprowadza się instrukcję likwidacji /dealokacji/ obiektu. Dalej wyróżnia się instrukcje przyłączenia /aktywacji/ i odłączenia /pasywacji/ współprogramu, oraz instrukcje dla procesów: aktywacji, zatrzymania, oczekiwania /na zakończenie syna/ oraz wejścia/wyjścia do/z rejonu krytycznego.

● ADA:

Wyróżnia się instrukcję podstawienia /nadającą nową wartość ustalonej zmiennej/, wywołanie podprogramu, sterowania /skoku, wyjścia z pętli, zakończenia wywołania podprogramu, z ewentualnym obliczeniem wartości funkcji/. Ponadto wyróżnia się instrukcje synchronizacji dla zadań.

Instrukcje złożone.

● LOGLAN:

Wprowadza się instrukcje warunkową i case /umożliwiające wykonywanie pewnych akcji zależnie od spełnienia odpowiednich warunków/, pętli z wieloma wyjściami /umożliwiające iteracyjne wykonywanie pewnych akcji dopóki nie będzie spełnił on jeden z warunków wyjścia, tzn zakończenia wykonywania/, oraz bloku i bloku prefiksowanego.

● ADA:

Wprowadza się instrukcje warunkowe, pętli, case i bloku /o znaczeniu jak podano powyżej/.

Zarówno w języku ADA jak i LOGLAN nie wprowadza się klasycznej instrukcji złożonej begin...end z ALGOLu.

W II części pracy podamy bardziej szczegółowy opis konstrukcji programowych obujętych.



MODELE SYMULACYJNE W PROJEKTOWANIU SYSTEMÓW INFORMATYCZNYCH

Artykuł jest kontynuacją opracowania p.t. "Wykorzystanie symulacji w projektowaniu zbiorów danych" zamieszczonego w nr 5/1978 Biuletynu Technicznego "Mera" autorów: K. Chwesiuk, B. Królikowska, A. Szewczyk.

Cechy modeli symulacyjnych

Zastosowanie badań empirycznych przy analizie systemów /procesów, zjawisk/ jest często praktycznie ograniczone, gdyż eksperymentowanie na obiektach realnie funkcjonujących dla określonych celów jest niepożądane z co najmniej dwóch powodów [26]:

- zakłócenie eksperymentami funkcjonowania systemu ogranicza lub uniemożliwia bieżące realizowanie celów,
- eksperymentowanie na konkretnym systemie prowadzi do suboptymalizacji rozwiązań definiowanych lokalnymi własnościami systemu i otoczenia.

Zupełnie niemożliwe jest natomiast prowadzenie eksperymentów na dopiero projektowanym systemie, ze względu na fragmentaryczność uzyskiwanych wyników i niemożność przewidzenia na tej podstawie efektu końcowego. Przyjmuje się więc, że właściwą metodą badania, a zwłaszcza projektowania i usprawniania systemu jest jego modelowanie a następnie symulowanie jego działania. Model symulacyjny jest jednym z rodzajów modeli matematycznych. Charakteryzuje się on następującymi cechami [15], [6], [14]:

- obiektywną odpowiedzialnością z modelowanym systemem,
- zdolnością zastępowania go pod określonymi względami,
- zdolnością dodawania w trakcie badania modelu pewnej dopuszczalnej kontroli informacji,
- istnieniem dostatecznie ścisłych prawideł przejścia od informacji modelowej do informacji o samym modelowanym obiekcie,
- zawieraniem wielu zmiennych, parametrów i funkcji,
- zmienne mogą być losowe, z określonym rozkładem i ograniczeniami,
- występowaniem wielu różnorodnych powiązań między elementami modelu,
- relacje nie zawsze mogą mieć określony czas trwania,

- umożliwianiem obserwacji zachowania się /działania/ systemu w trakcie realizacji celu, do którego ów system został lub ma zostać utworzony /charakter dynamiczny/.

Model symulacji /MS/ /w tym także komputerowej/ tworzy się z matematycznego modelu systemu /MM/ i algorytmu symulacji tego modelu A^S [3], [22], [25]:

$$M^S = \langle M^M, A^S \rangle = \langle E, S^M, T \times S, F \rangle$$

gdzie:

- E_M - zbiór elementów modelu /systemu/,
- S^M - matematyczna struktura modelu systemu, zbudowana z funkcji lub innych struktur /np. rozkładów prawdopodobieństw/ pozwalających na wykonywanie obliczeń numerycznych w miejsce relacji,
- $T \times S$ - iloczyn kartezjański zbioru chwil czasu i zbiorów stanów modelu systemu w tych chwilach,
- F - funkcja przejścia, posiadająca właściwość określania stanu modelu systemu w chwili następczej, na podstawie znajomości stanu modelu systemu w chwili poprzedniej.

Warianty wykorzystania modelu symulacyjnego systemu

Wyróżnia się następujące warianty wykorzystania modelu symulacyjnego systemu [16]:

- **B a d a n i e s y s t e m u w s p o s ó b b e z p o ś r e d n i** - jeśli znana jest struktura systemu /badanie nie jest celem działania/ oraz informacje wejściowe charakteryzujące system, to zbudowany model symulacyjny umożliwia takie badanie. Stanowi on ilustrację działania systemu, będącą efektywnym narzędziem lepszego poznania jego charakteru i występujących w nim prawidłowości. Może również służyć do sprawdzania efektów podejmowanych decyzji i procedur przed ich zastosowaniem w praktyce,
- **I d e n t y f i k a c j a** /modelowanie/ systemu - mając do dyspozycji zestawy wejściowych informacji empirycznych oraz odpowia-

dające im relacje systemu rzeczywistego, szuka się struktury modelu. Porównuje się reakcje systemu rzeczywistego i modelu na identyczne wymuszenia wejść. Sprawdza się hipotezy o systemie, którego się dobrze nie zna i którego zachowanie stara się wyjaśnić, zakładając istnienie pewnych składników strukturalnych i modyfikując model w trakcie poznania prawidłowości. Poznanie to pozwala na modyfikację systemu rzeczywistego, w kierunku wzrostu jego efektywności,

- **S t e r o w a n i e s y s t e m e m** - gdy znana jest struktura i pożądane reakcje wyjściowe, a znajduje się informacje wejściowe. Kolejno są one przybliżane przez hipotezy,

- **D o b i e r a n i e m o d e l u s y s t e m u** jest sposobem użycia modelu symulacyjnego w badaniach socjologicznych, ekonomicznych, politycznych i medycznych, tam, gdzie znane jest zachowanie się systemu, natomiast nie są znane wywołujące je procesy. Stawia się więc hipotezy o przypuszczalnych przyczynach zdarzeń i porównuje z zachowaniem się systemu rzeczywistego.

- **P r o j e k t o w a n i e s y s t e m u** - poszukiwanie takiej struktury systemu, która możliwie najlepiej realizuje założony /znany/ układ wejścia/wyjścia. Teoretycznie - rozwiązań idealnych nie ma. Są ograniczenia co do możliwości stosowania określonych wariantów strukturalnych modelu, charakteryzujących się tym samym układem wejścia/wyjścia. W przypadku tym buduje się model hipotetyczny w celu uzyskania informacji o systemie rzeczywistym, który analityk /człowiek lub procedura/ sam tworzy i który jest mu znany. W miarę uzyskania informacji o prawidłowościach systemu, modyfikuje się jego strukturę. Symulacja daje tu szczegółowy, dynamiczny i kompleksowy obraz charakterystyk działania przyszłego systemu. Symulacja jest bowiem jednym z kierunków szeroko pojmowanego usprawnienia procesu projektowania systemów informatycznych. Proces projektowania składa się z określonego ciągu działań, w trakcie których projektant wykorzystuje odpowiednie środki techniczne i informacyjne. Sprawność elementarnego działania zależy między innymi od środków wykorzystywanych w trakcie jego realizacji.

Działania realizowane na początku procesu projektowania charakteryzują się na ogół dużym udziałem projektanta. Jest to wynikiem niewielkiego zasobu informacji o projektowanym rozwiązaniu; w przeważającej części są to więc czynności twórcze. Na tym etapie wykorzystywane są przede wszystkim metody heurystyczne i szeroko rozumiane modelowanie [23]. W projektowaniu szczegółowym, wraz ze wzrostem informacji o nowym rozwiązaniu projektowym, występuje rosnący udział prac powtarzalnych, co stawia dogodne warunki do stosowania instrumentalizacji projektowania [4]. Można więc wyodrębnić dwie grupy narzędzi umożliwiających wspomaganie procesu projektowania informatycznego systemu zarządzania:

- umożliwiające zbudowanie modelu systemu w oparciu o narzędzia informatyki, w celu wszechstronnego przebadania i oceny użyteczności rozwiązania dla przyszłych użytkowników systemu,

- wspomagające dokumentowanie prac projektowych za pomocą komputera^{x/}.

Zastosowanie pierwszej grupy narzędzi polega na skonstruowaniu modelu przyszłego rzeczywistego systemu informatycznego - już we wstępnych etapach projektowania. Tak skonstruowany model musi być poddany wszechstronnym badaniom z punktu widzenia:

- Użytkownika; rozwiązanie takie umożliwi zaprezentowanie konkretnych możliwości projektowanego rozwiązania. Użytkownik, definiując w sposób ścisły swoje potrzeby może więc na tym etapie w znaczny sposób wpłynąć na kształt przyszłego systemu,

- Projektanta systemu informatycznego; badanie to umożliwia przeanalizowanie licznych wariantów i przyjęcie rozwiązań zapewniających optymalną - w danych warunkach i według określonych kryteriów - postać systemu informatycznego.

Obszary systemu informatycznego wspomagane symulacją

Symulacja w przypadku projektowania ma więc na celu sprawdzenie w efektywny sposób działania projektowanego systemu bądź udoskonalenie istniejących już systemów /tzw. projektowanie kroczące/.

Obszary systemu informatycznego wspomagane metodami modelowania symulacyjnego wyznaczone są przez:

- proces projektowania systemów,
- system projektujący,
- gotowy już system informatyczny.

W ostatnim przypadku występuje dość specyficzne zjawisko: systemy liczące, które dotychczas służyły jako narzędzie wykorzystywane do rozwiązywania problemów w innych dziedzinach, same stały się przedmiotem badań naukowych. Spowodowane to zostało głównie burzliwym i dość często niekontrolowanym z różnych względów rozwojem zarówno techniki wykorzystania systemów jak i technologii ich budowy. Symulację stosuje się tu do oceny systemu obliczeniowego na różnych poziomach jego hierarchii strukturalnej, np. symulacja podsystemów: pamięci zewnętrznej o bezpośrednim dostępie, wewnętrzne działanie jednostki centralnej na poziomie międzyrejestrów itp. [5].

Symulacja systemu projektującego może być natomiast narzędziem ułatwiającym podjęcie decyzji o organizacji pracy i podziale środków w ramach systemu projektującego. Oczywiście może to mieć miejsce w ramach dużych, wielozadaniowych zespołów. W modelu symulacyj-

^{x/} Ten sposób usprawnienia procesu projektowania nie będzie w artykule omawiany i dlatego pomijamy jego charakterystykę.

nym systemu projektującego poszczególne podzespoły czy stanowiska mogą być traktowane jako "stanowiska obsługi", przez które przechodzi projekt systemu. W trakcie procesu symulacji generowane są zadania projektowe napływające z zewnątrz do systemu projektującego. Zadania te poddawane są obróbce, której efektem jest projekt i jego oprogramowanie. Przebieg poszczególnych zadań zależy od parametrów zadania, a także od charakterystyk poszczególnych zespołów. Przeanalizowanie wielu wariantów realizacji procesu projektowania pozwala w tym przypadku na wybór właściwej organizacji tego procesu.

Przy projektowaniu systemu informatycznego natomiast symulacja pomaga projektantowi w definiowaniu problemu, określeniu ilościowego wpływu poszczególnych czynników na projektowany system, w wyborze ostatecznego wariantu systemu z szeregu alternatywnych rozwiązań; wskazuje ponadto na przeciążenie elementów systemu, na krytyczne miejsca w projektowanym systemie oraz stanowi podstawę do oszacowania wydajności projektowanego systemu. Przy pomocy symulacji projektant uczy się działania systemu i na odwrót - symulacja wymaga od projektanta znajomości działania systemu i modelu [18].

Dekompozycja systemu informatycznego według jego struktur

Projektowany system informatyczny rozpatrywany dalej będzie w układzie jego struktur. Jest to wskazane ze względu na dużą różnorodność składników, charakteryzujących się różnymi cechami. Ta niejednorodność cech poszczególnych składników skłania do rozpatrywania systemu informatycznego w aspekcie jego struktur, z których każda charakteryzuje się tym, że składa się z elementów o cechach jednorodnych. Poszczególne elementy mogą być wówczas rozróżniane za pomocą określonych miar. Zasadniczą przesłanką strukturalizacji projektowanego systemu informatycznego jest więc wysoka złożoność takiego systemu, co utrudnia budowę jego modelu i powoduje niemożność zastosowania metod symulacyjnych przy użyciu maszyny cyfrowej do odwzorowania jego działania jako całości. Możliwe jest natomiast śledzenie stanów jego struktur, których funkcjonowanie można opisać za pomocą charakterystyk /parametrów/ wyrażonych ilościowo [20].

Dekompozycję systemu informatycznego przeprowadza się według następujących struktur:

- informacyjnej, rozumianej jako zbiór elementów przechowujących informacje w systemie wraz z relacjami przyporządkowującymi informacje do poszczególnych zbiorów,
- technicznej, którą stanowią urządzenia techniczne niezbędne do realizacji zadań systemu oraz wzajemne powiązania między tymi urządzeniami,

- przestrzennej, określanej jako zbiór zlokalizowanych punktów nadawania, gromadzenia, przetwarzania i komutacji oraz dróg komunikacyjnych między tymi punktami,
- funkcjonalnej, będącej zbiorem celów i funkcji systemu oraz wzajemnych ich współzależności.

Wykorzystanie symulacji w projektowaniu poszczególnych struktur systemu informatycznego

Modelowanie symulacyjne struktur systemu obejmuje zarówno opis elementów systemu składającego się na poszczególne struktury jak i optymalizację poszczególnych struktur, będącą optymalizacją modelu systemu - w aspekcie celów i funkcji. W systemach informatycznych z reguły występują znacznie większe możliwości wyboru struktury niż np. w systemach technicznych. Obecnie projektowanie systemów nawet o dużym stopniu złożoności przeprowadza się w drodze opracowania jednego z możliwych wariantów - w niemałym stopniu uzależnionego od indywidualnych umiejętności i doświadczenia projektantów.

Zastosowanie metody symulacji komputerowej umożliwia zbadanie większej liczby wariantów, a tym samym stwarza możliwość lepszego wyboru. Struktura funkcjonalna systemu informatycznego jest stosunkowo najmniej podatna na modelowanie symulacyjne, głównie ze względu na jakościowe a nie ilościowe kryteria jej rozpatrywania. Dopiero jej dekompozycja w:

- przekroju struktury procesów,
 - przekroju struktury zasobów,
 - przekroju struktury komórek
- a więc rozpatrywanie jej w aspekcie pozostałych struktur /technicznej, informacyjnej, przestrzennej/ stwarza możliwość budowy modeli symulacyjnych.

Symulacja struktury technicznej wchodzi w skład grup metod oceny i wyboru zestawu komputerowego zwanych metodami komparatyki komputerów [7]. Metoda symulacji umożliwia dobór środków technicznych informatyki do realizacji określonego systemu, zestawu systemów lub kilku niezależnych systemów. Tego typu badania symulacyjne poprzedzają zakup nowego lub rozbudowę eksploatowanego zestawu komputerowego. Metodę symulacji można stosować poza tym do analizy przepustowości posiadanego sprzętu, w celu jego lepszego wykorzystania. W zakresie tego typu problematyki istnieją już konkretne rozwiązania np. pakiet programów SCERT [9], [19].

Symulacja struktury przestrzennej systemu może ułatwić rozwiązanie takich problemów jak: właściwy obieg informacji, lokalizacji punktów koncentracji danych, miejsca tworzenia maszynowych nośników informacji, lokalizacji i rozmiarów banków danych. Wydaje się, iż kierunek ten będzie miał istotne znaczenie przy tworzeniu dużych hierarchicznych systemów, gdzie ilość możliwych wariantów jest szczególnie duża.

Symulacja struktury informacyjnej, ze względu na jej szczególną złożoność, może obejmować jedynie jej poszczególne wycinki. Te szczegółowe rozważania mogą dotyczyć np.:

- symulacji bazy indeksowo-normatywnej pod kątem np. niezawodności kodów, badanie efektywności różnych metod budowy cyfr kontrolnych, wpływu zmian norm na efekty systemu itp.,

- symulacji projektu oprogramowania systemu w zakresie projektowania równoległego i weryfikacji założeń [24], [17]. Przy projektowaniu równoległym z zastosowaniem symulacji dzieli się program na segmenty i niezależnie uruchamia każdy segment przy użyciu symulacyjnych fantomów pozostałych części programów. Testowanie z kolei może przebiegać bez rzeczywistych zbiorów na tzw. sztucznej bazie danych.

Weryfikacja założeń ma natomiast na celu zintegrowanie oddzielnych w aktualnym systemie projektowania faz, jakimi są: opracowywanie algorytmów i pisanie programów. Łatwość otrzymania sztucznej bazy danych /za pomocą generatorów sztucznych danych/ jak również stosowalność metod teorii masowej obsługi, zwłaszcza do systemów zapytaniowych, które mogą być traktowane jako układ zdarzeń, pozwala na doświadczalną weryfikację programów i założeń dotyczących np. efektywności systemu. Prócz tego weryfikacji symulacyjnej podlegać mogą inne założenia przyjęte w projektowaniu np. podział systemu na jednostki przetwarzania, co stwarza możliwość badań optymalizacyjnych dotyczących np. ilości danych w systemie, częstości ich wykorzystywania, czasochłonności przebiegów wyceny kosztów systemu itp. [21].

- symulacji zbiorów danych^{x/} w aspekcie ich elastyczności, organizacji, struktury danych, możliwości otrzymania informacji /w odpowiednim czasie i żądanym przekroju/ czyli zaspokojenia potrzeb użytkowników itp. [8], [2].

Symulację zbiorów danych można stosować już od pierwszego etapu, tzn. od ustalenia założeń odnośnie ich zawartości. W przyjętym zazwyczaj układzie projektowania, założenia dotyczące zbiorów ustalane są przez projektanta i użytkownika na początku procesu projektowania i nie są weryfikowane ze względu na wielką pracochłonność związaną z taką weryfikacją. Jednakże, jak wskazują doświadczenia z eksploatacji systemów informatycznych, najbardziej zawodnym elementem są obecnie dane, powodujące /ze względu na ich niekompletność, nadmiar czy przekłamanie/, konieczność wielokrotnych korekt [11].

Po określeniu zawartości zbiorów ważną sprawą jest właściwe określenie struktury zapisu danych; od tej cechy bowiem zależy efektywność przetwarzania danych. W tradycyjnie przyjętym systemie projektowania jakość struk-

tur danych trudno zweryfikować ze względu na wspomnianą wyżej wielką pracochłonność związaną ze zmianami układu danych. Dlatego też optymalizacja struktur danych ze względu na kryterium efektywności systemu nie jest praktycznie nigdy przeprowadzana. Metody symulacji znacznie upraszczają to zagadnienie optymalizacji [1], [13].

Zalety zastosowania metod symulacji przy projektowaniu systemu informatycznego

Wydaje się więc celowe podkreślenie korzyści wynikających z zastosowania symulacji właśnie na tym odcinku procesu projektowania [12], [10], [2]:

1. Przy pomocy symulacji łatwo mogą być podejmowane decyzje dotyczące zawartości i struktury zbiorów danych już w fazie opracowywania koncepcji, bez konieczności przeprowadzania następujących faz projektowania i realizacji: tym samym kosztowne metody prób i błędów mogą być częściowo lub całkowicie wyeliminowane.

2. Działania na zbiorach danych mogą być symulowane i obserwowane we wszystkich możliwych warunkach; parametry systemowe i parametry otoczenia mogą być łatwo zmieniane w celu symulacji każdego żądanego warunków, nie jest więc konieczne poleganie wyłącznie na badaniach eksploatacyjnych.

3. Czas trwania prób działań na zbiorach może być skrócony o dużą liczbę rzędów wielkości; może on też być kondensowany lub rozciągany, co pozwala na obserwację procesów w dowolnej skali.

4. Uzyskiwane sztuczne, jednakże prawdziwe dane mogą być otrzymane szybko i w dużej ilości przy nieporównywalnie niższych kosztach niż manipulowanie rzeczywistymi zbiorami.

5. Model symulacyjny zbiorów danych systemu informatycznego o maksymalnej elastyczności może być adaptowany do prawie wszystkich rozwiązań rzeczywistych jak również hipotetycznych. Symulacja może być więc użyta do eksperymentów na nowych sytuacjach, o których ma się niewiele lub wcale nie ma się informacji. Modelowanie symulacyjne może tu więc służyć jako "test ochronny" do wypróbowywania nowych rozwiązań, zanim ryzyko przejdzie z systemu eksperymentalnego do rzeczywistego.

6. Symulacja zbiorów jest całkowicie powtarzalna, ponieważ użytkownik w pełni steruje rozwojem modelu i ilością przebiegów symulacyjnych.

7. Model symulacyjny jest wolny od ograniczeń fizycznych właściwych dla danego systemu, ponieważ jest on przedstawiony w czysto symbolicznej i logicznej formie.

^{x/} Zbiory są tu rozumiane jako kolekcje danych na wejściu i wyjściu systemu oraz kolekcje danych przechowywanych w systemie.

8. Doświadczenia uzyskane przy projektowaniu modeli symulacyjnych często sugerują zmiany w symulowanym systemie. Skutki tych zmian mogą być potem sprawdzane przez symulację - przed ich realizacją w ramach rzeczywistego systemu informatycznego.

9. Ułatwiona jest kontrola wyboru najwłaściwszego rozwiązania np. optymalnej struktury czy organizacji zbiorów, gdyż wyniki przebiegów symulacyjnych mogą być rozwiązaniem funkcji kryterium.

Z przedstawionych rozważań wynika, że technika symulacyjna jest wygodnym i korzystnym sposobem umożliwiającym kolejne 'podejścia' w projektowaniu systemu informatycznego. Wprowadzając modelowanie symulacyjne na poszczególne wycinkowe etapy projektowania struktur systemu można otrzymywać symulacyjną drogą prób i błędów - optymalne rozwiązanie.

L i t e r a t u r a :

[1] K. Bauknecht, J. Kohlas, C. A. Zehnder: Simulationstechnik - Entwurf und Simulation von Systemen auf digitalen Rechenautomaten, Springer-Verlag, Berlin Heidelberg, New York, 1976.

[2] K. Chwesiuk, B. Królikowska, A. Szewczyk: Wykorzystanie symulacji w projektowaniu zbiorów danych, Biuletyn Techniczny "Mera", 5/1978.

[3] E. Dec, S. Pabis: Modelowanie systemów dla symulacji komputerowej, w: Teoretyczno-praktyczne problemy projektowania informatycznych systemów zarządzania, INFOGRYF'78, TNOiK, Szczecin-Kołobrzeg, 1978.

[4] W. C. Dorosiński: Wybrane aspekty wspomaganie projektowania, w: Współczesne problemy projektowanie systemów informatycznych, TNOiK, Szczecin, 1977.

[5] M. E. Drummond jr: Analiza i ocena eksploatacji systemów komputerowych, WNT, Warszawa, 1979.

[6] G. W. Evans, G. F. Wallace, G. L. Sutherland: Symulacja na maszynach cyfrowych, WNT, Warszawa, 1973.

[7] G. Friederich: Simulation von EDV - Anlagen, Data Report, 11/1976, Zeszyt 1.

[8] Z. Głodek, J. Perenc, A. Szewczyk: Symulacja w wyborze struktury zbiorów danych, Informatyka, 5/1979.

[9] J. Ilczuk: SCERT - metoda racjonalnego doboru zestawu komputerowego, Informatyka, 2/1978.

[10] L. Kondratowicz: Symulacja cyfrowa w języku CSL, PWN, Warszawa, 1974.

[11] P. Lorenz, F. Stuchlik: Die Simulation als Instrument der Projektierung einer Datenbank, Rechentechnik - Datenverarbeitung, 6/1977.

[12] F. Martin: Wstęp do modelowania cyfrowego, PWN, Warszawa, 1976.

[13] H. F. Meuche, H. J. Schinköthe, G. Ullmann: Model und Methodenbanken, Rechentechnik-Datenverarbeitung, 4/1977.

[14] J. H. Naylor: Modelowanie cyfrowe systemów ekonomicznych, PWN, Warszawa, 1975.

[15] S. Pabis: Metodologiczne podstawy modelowania systemów informatycznych, w: Materiały na IV Krajowe Sympozjum Eksploatacji Urzędów Technicznych, tom 2, OPT, Katowice, 1977.

[16] Praca zbiorowa: Metody symulacji komputerowej, OBRI, Warszawa, 1975.

[17] E. Rasek: Über ein Simulationsmodell für den Entwurf der servosysteme in Magnet-Handgeräten, Angewandte Informatik, 7/1973.

[18] J. Reitman: Computer Simulation Applications - Discrete - event Simulation for synthesis and analysis of complex system, Wiley, New York, 1971.

[19] H. D. Saur: Ein Simulationsmodell für das Band-Platte - Operating an einem Großrechner, Angewandte Informatik, 5/1973.

[20] J. Skibiński: Zastosowanie podejścia systemowego w badaniach sprawności formalnych struktur systemów społecznych, Wojskowy Przegląd Organizacji i Informatyki, 1/1975.

[21] J. Sowiński: Uwagi na marginesie symulacji systemów oprogramowania, Wojskowy Przegląd Organizacji i Informatyki 4/1976.

[22] H. Strohe: Experimentelle Systemanalyse. Akademie - Verlag, Berlin, 1975.

[23] A. Szewczyk: Heurystyka pomoże, Informatyka, 3/1979.

[24] B. Walke: Entwicklung optimaler Zuteilungsstrategien für Rechner-Modelle durch Simulation und Rechnung, Elektronische Rechenanlagen, 9/1974, Zeszyt 1.

[25] P. M. Williams: Indeterminate Probabilities in Formal Methods in the Methodology of Empirical Sciences, Ossolineum, Wrocław, 1976.

[26] H. Wysocki: Modelowanie informatycznych systemów zarządzania, w: Teoretyczno-praktyczne problemy projektowania informatycznych systemów zarządzania, INFOGRYF'78, TNOiK, Szczecin - Kołobrzeg, 1978.

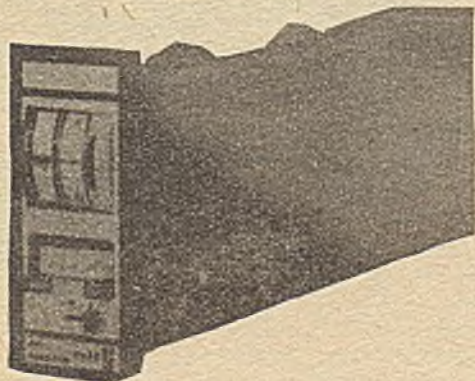


inż. TADEUSZ SINOŁĘCKI
OBR Automatyki Przemysłowej
"Mera - Pnefal"

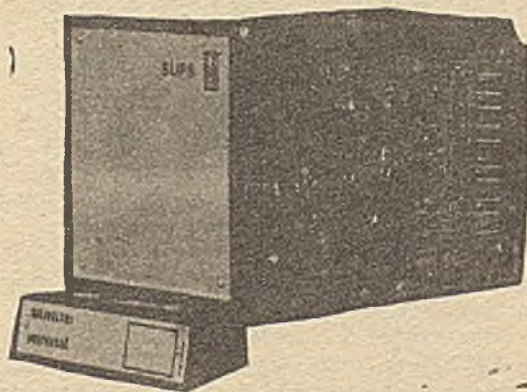
JUGOSŁOWIAŃSKA APARATURA DO AUTOMATYZACJI PROCESÓW PRZEMYSŁOWYCH

Zacieśniająca się współpraca pomiędzy PRL i SFRJ oraz przewidywane rozszerzenie wymiany towarowej również w dziedzinie tzw. środków automatyzacji stwarza możliwości uzyskania na korzystniejszych niejednokrotnie warunkach niż z krajów zachodnich, importowanego sprzętu z tej dziedziny. Jedną z przeszkód stanowi fakt, że jugosłowiańskie systemy automatycznego sterowania i ich elementy nie są na ogół znane w naszym kraju. Niniejsza, oparta na materiale katalogowym, informacja ma za zadanie zasygnalizować niektóre pozycje z tej dziedziny.

Produkcją urządzeń automatyki oraz automatyzacją obiektów przemysłowych zajmują się w SFRJ około 12 firm - kombinatów stowarzyszonych już od 17 lat w Jugosłowiańskim Zrzeszeniu Automatyki - UNIMATIK, którego siedzibą jest Belgrad. Jednym z czołowych kombinatów tego Zrzeszenia jest Energoinvest - zatrudniający około 33 tys. pracowników w 46 zakładach. Obok działalności obejmującej automatyzację kompletnych obiektów szczególnie w dziedzinie energetyki i hydroenergetyki, przemysłu spożywczego /m. in. cukrownictwo/, chemicznego itp., kombinat jest producentem kompletnych rozbudowanych systemów automatycznego sterowania oraz szerokiego asortymentu aparatury pomiarowej.



Fot. 1. Regulator PID systemu SUPS



Fot. 2. Przetwornik małych napięć /np. sygnałów z termoelementów/ typu TEM 100

Poniżej zostaną przedstawione najbardziej interesujące systemy automatyki i wybrane przyrządy kontrolno-pomiarowe produkowane przez Energoinvest.

Blokowy system automatycznej regulacji SUPS jest klasycznym analogowym systemem automatyki elektronicznej o standardowym sygnale prądu stałego 4...20mA /lub 0...20mA/. Część przetwornikowa zapewnia przetwarzanie:

- różnicy ciśnień /przetworniki DEM100, DEM106, DEM200/ odpowiednio w zakresach: od 0...5kPa do 0...500kPa /przetwornik DEM100/; od -10mmH₂O do 15mmH₂O, 25...50mmH₂O, 50...250mmH₂O, 250...1200mmH₂O /przetwornik DEM106/; od 0...500mmH₂O do 0...5000mmH₂O, od 0...4000mmH₂O do 0...20000mmH₂O /przetwornik DEM200/.
- ciśnienia przetwornik PEM100 w zakresie od 0...0,25MPa do 0...45MPa
- sygnałów z termoelementów /5...100mV/ oraz czujników termooporowych /20...200Ω, Pt100/ - przetwornik TEM10, przy czym nieliniowość przetwarzania przetworników jest rzędu 0,5% a zakres temperatur pracy wynosi -20°C...+50°C.

Do zasilania przetworników przewidziano zasilacz NEJ100, o napięciu wyjściowym 42V -1V. Przetworniki mają budowę iskrobezpieczną, przy czym elementami pośredniczącymi między przetwornikami a przyrządami części centralnej systemu są bariery ochronne:

- typ SB44 /dla współpracy z przetwornikami pomiarowymi/,
- typ SB22 /dla współpracy z przetwornikami wykonawczymi/.

Część centralna systemu oprócz regulatorów tablicowych P, PI, PID, regulatora adaptacyjnego RAD100 i stacyjki sterowniczej DIS100 składa się z bogatego zestawu modułowego i elementów obróbki informacji, takich jak:

- blok mnożenia MAS100,
- blok dzielenia DAS100,
- blok podnoszenia do kwadratu KVD100,
- blok pierwiastkowania KORU100,
- blok sumowania algebraicznego 3 sygnałów ALG100,
- ogranicznik sygnału PIU100,
- blok różniczkujący NPD100,
- blok całkujący z /pamięcią analogową/MIA100,
- wybierak maksymalnego lub minimalnego z czterech sygnałów SAS100,
- oddzielnik galwaniczny sygnałów GIS1009/ o oporności wejście/wyjście ponad 200M Ω /
- blok sygnalizacji dwugranicznej KPS100
- przetwornik analogowo-częstotliwościowy z licznikiem impulsów /integrator/ o charakterystyce liniowej PIF100 lub pierwiastkującej IKF100,
- ręcznie sterowany blok mnożenia przez współczynnik 0,5...2 typ RA100.

Dokładność przetwarzania bloków analogowych części centralnej wynosi -0,2% zakresu sygnałów, a temperatura pracy od +5°C do +50°C. Dla części centralnej przewidziano zasilacz TR100 /220V -10% 50Hz/ 24 -10% 50Hz/.

Część centralną systemu uzupełniają:

- przetwornik elektropneumatyczny EPP100S o nieliniowości 1%,
- blok sterowania silnikiem elektrycznym PDM100 o histerezie 1%.

Elementy wykonawcze to szeroki wachlarz serwomechanizmów elektrohydraulicznych liniowych serii ESHL o skokach 18...118mm i obciążeniu 5...50kN, katowych serii EHSU /0...70°, 0,25...4kNm/, oraz silowników pneumatycznych z ustawnikami pozycyjnymi elektropneumatycznymi /EPS3, EPS5/ i pneumatycznymi /PS3, PS5/ o zakresie 0...25mm i 0...40mm lub 0...20° i 0...30°.

System elektronicznych dyskretnych elementów sterowania SPRINT LOGIC zawiera 30 podstawowych modułów zrealizowanych na bazie obwodów scalonych /TTL/.

W skład systemu wchodzi elementy realizujące podstawowe funkcje logiczne i działania

sekwencyjne, dekodery, komparatory sygnałów, liczniki, czujniki wielkości elektrycznych, elementy dopasowujące wyjście, oddzielnice galwaniczne, wzmacniacze wyjściowe, lampki sygnalizujące itp. Łatwa wymiana modułów, wygrawerowane na płytkach czołowych schematy i oznaczenia zacisków ułatwiają eksploatację i ewentualne korekty struktury układów na obiekcie. System przystosowany jest do pracy w warunkach istnienia silnych zakłóceń elektrycznych z myślą o stosowaniu na obiektach energetycznych.

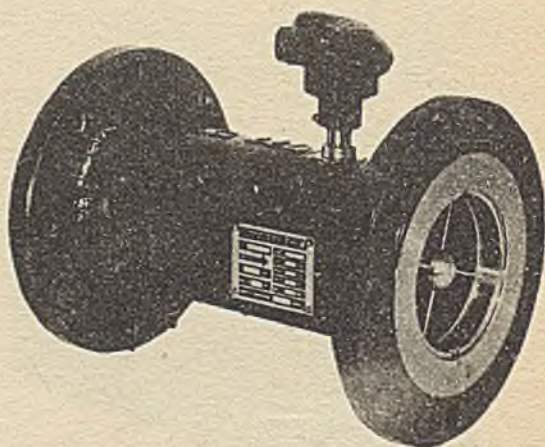
Napięcie zasilania systemu 12V DC $\begin{matrix} +15\% \\ -20\% \end{matrix}$

Zakres dopuszczalnych temperatur pracy wynosi 0...70°C.

System TLC 11M jest hierarchicznym systemem teletransmisji danych dla celów kontroli i sterowania obiektami skomplikowanymi i rozproszonymi na dużej przestrzeni. Jest on szczególnie przydatny dla:

- systemów energetycznych /elektrownie, przesyłanie i dystrybucja energii/,
- systemów transportu gazów cieczy i materiałów sypkich /rurociągi/,
- systemów sterowania ruchem drogowym,
- obiektów wielkiej chemii.

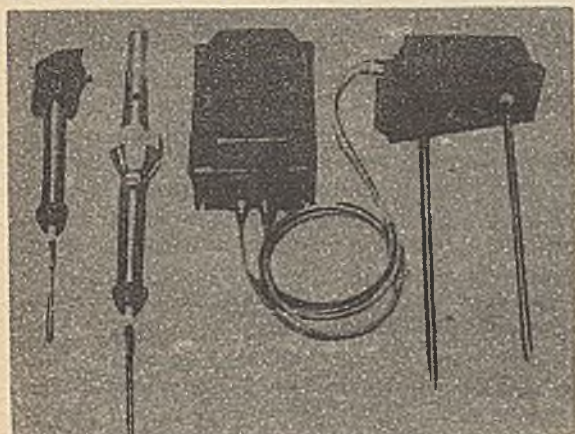
System zbudowany jest w sposób hierarchiczny. Centralna stacja /nadrzędna/ może współpracować w dowolnej konfiguracji z 1 do 32 stacjami podrzędnymi, z których każda może obsługiwać 256 punktów pomiarowych, 1024 punkty sygnalizacyjne i 1024 punkty sterownicze. W skład systemu wchodzi 50 modułów zapewniających bardzo elastyczną architekturę systemu. Bazowym elementem systemu jest produkowany przez Energoinvest mikroprocesor PUMAT z pamięcią operacyjną 64K słów 16-bitowych, o czasie dostępu 4 μ s. Przyrządy systemu zasilane są napięciem stałym 48V. Zakres temperatur pracy wynosi 0...45°C.



Fot. 3. Przepływomierz turbinkowy typu TMPK

"Energoinvest" jest również producentem bogatego zestawu aparatury pomiarowej. Nie sposób tu prezentować pełnego wachlarza tej aparatury wydaje się jednak, że warto zwrócić uwagę projektantów na przepływomierze turbinkowe i ultradźwiękowe, mierniki kalorymetryczne oraz sygnalizatory poziomu. Przepływomierze turbinkowe dla rurociągów o średnicach 15... 300 mm kalibrowane wraz z przyłączem rurowym, mobowanym kołnierzo /DIN/, lub za pomocą przyłączy Ermeto, odcinkiem pomiarowym, w którym umieszczony jest wirnik nadajnika impulsów, zapewniają bardzo wysokie parametry metrologiczne:

dokładność $\pm 0,25\%$ do $\pm 0,5\%$, niepewtarzalność rzędu $-0,1\%$.

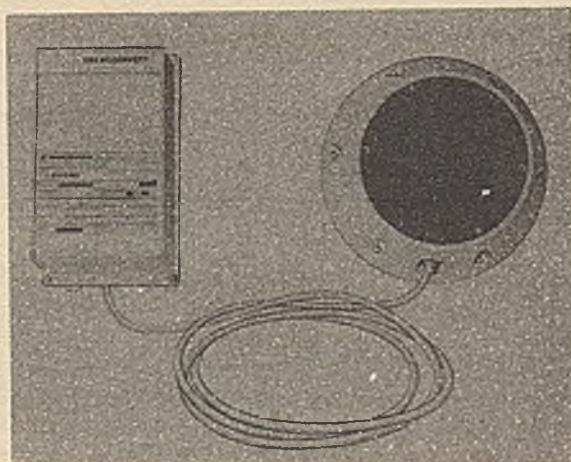


Fot. 4. Rezonansowy sygnalizator poziomu cieczy typu RES 100P

Wchodzący w skład zespołu pomiarowego przetwornik częstotliwości na standardowy sygnał prądowy 4... 20mA / 0... 20mA/ umożliwia współpracę z dowolnym systemem automatyki analogowej. Przepływomierz wyposażony jest w miernik chwilowej wartości natężenia przepływu, oraz licznik przepływającej cieczy /objętości/. Dla rurociągów o dużych średnicach /100... 200mm/ przewidziano montaż wirnika turbinki bezpośrednio w rurociągu na wysięgniku. Uzyskuje się wtedy dokładność pomiaru rzędu $\pm 1,5\%$... $\pm 2\%$. Na bieżąco przepływomierza turbinkowego i termometrów oporowych, umieszczonych na wlocie i wylocie czynnika grzejnego odbiornika energii cieplnej, opracowano zestawy pomiarowe serii MTE pozwalające na pomiar chwilowego natężenia strumienia cieplnego i natężenia przepływu czynnika grzejnego zliczanie ilości dostarczonego ciepła /energii cieplnej/ spadku temperatury na odbiorniku itp.

Inny rodzaj przepływomierza, to przepływomierz ultradźwiękowy przeznaczony do pomiaru przepływu cieczy w kanałach otwartych. Bazuje on na zmianach poziomu w pomiarowym, o zmiennym jak dla dyszy Venturiego przekroju, odcinku kanału. Zmiany poziomu, mierzone za pomocą ultradźwiękowej sondy, przeliczone są przez mikroprocesor na chwilową wartość natężenia przepływu lub zliczoną w żądanym okresie czasu objętość przepływającej cieczy, które to parametry

wskazywane są na miernikach przyrządu, oraz wyprowadzone w postaci sygnałów wyjściowych 0... 10V lub 0... 20mA.



Fot. 5. Rezonansowy sygnalizator materiałów sypkich typu RES 100G

Ciekawym elementem nie tyle pomiarowym, co przeznaczonym raczej dla układów blokad i sygnalizacji jest rezonansowy czujnik poziomu cieczy /również cieczy agresywnych i wybuchowych/ lub materiałów sypkich o średnicy ziarna do 50mm o parametrach:

- dopuszczalna temperatura cieczy: -40°C ... $+250^{\circ}\text{C}$
- dopuszczalne ciśnienie cieczy: 0... 6,4 MPa,
- dopuszczalna temperatura ciał sypkich: -40°C ... 100°C
- dopuszczalne ciśnienie ciał sypkich: 40... 0,25 MPa
- zasilanie: 220V $\pm 10\%$, 50Hz,
- element wyjściowy: zestaw o obciążalności 250V, 5A /chwilowo 10A/,
- dokładność sygnalizacji ± 2 mm.

Jedną z interesujących pozycji, dla ewentualnego importu, wydają się również elektrohydrauliczne napędy do zaworów odcinających o dużych średnicach. Są to napędy serii EHPU przeznaczone dla zaworów o średnicach nominalnych od 100 do 1000mm i nominalnych ciśnieniach od 1,6 do 16MPa.

Podstawowe parametry tych przyrządów są następujące:

- zasilanie. 220/380V, 50Hz
- kąt obrotu wału 0... 70°
- moment siły 250... 4000Nm
- czas przestawiania 19... 120s
- zakres temperatur pracy: -20°C ... $+50^{\circ}\text{C}$

Firma produkuje również typoszereg napędów elektromechanicznych do zaworów odcinających, w tym również w wykonaniu przeciwybuchowym.

W niniejszym artykule przedstawiono wycinkowo tylko niektóre przyrządy i systemy automatyki produkowane przez Energoinvest. Poważnymi producentami przyrządów automatyki są także kombinaty TELEOPTYK i ATM. Czytelników zainteresowanych dokładniej sprzętem automatyki produkowanym w SFRJ, informujemy, że posiadamy odpowiednie katalogi firmowe.

INFORMACJE - NOWOŚCI

KOMPUTEROWY CZYTNIK - SORTER LISTÓW

Francuska firma CIT - Alcatel opracowała skomputeryzowane urządzenie służące do automatycznego odczytu adresów i sortowania kopert. Urządzenie zainstalowane w centralnym urzędzie pocztowym w pierwszej fazie segreguje listy na dwie grupy: adresowane ręcznie lub maszynowo. Listy adresowane maszynowo kierowane są do czytnika pisma. Urządzenie odczytuje dwa dolne wiersze adresu /nazwa miasta, ulicy/ i kieruje kopertę do zespołu indeksującego. Zespół indeksujący drukuje na brzegu koperty kod kreskowy odpowiadający odczytanemu adresowi. Kod adresowy naniesiony farbą fluorescencyjną może być później łatwo odczytany przez stosunkowo pro-

ste czytniki znaków stanowiące fragment urządzeń do sortowania, w które wyposażone mogą być lokalne placówki pocztowe.

Wydajność urządzenia odczytującego adresy i kodującego listy wynosi około 800 tys. szt. na dobę. Urządzenie może odczytać do 92% wszystkich listów. Tylko 0,03% adresów zostanie odczytana błędnie. Według wymagań poczty francuskiej odpowiednie liczby wynoszą 85% i 1,2%. Ze względu na wymaganą dużą niezawodność pracy układy elektroniczne łączone są przy pomocy specjalnego przewodu powleczonego fluoropolimerem "Tefzel" firmy Du Pont.

Opracował na podstawie informacji firmy Du Pont inż. Ludomir Kowalski - Zjednoczenie "Mera".



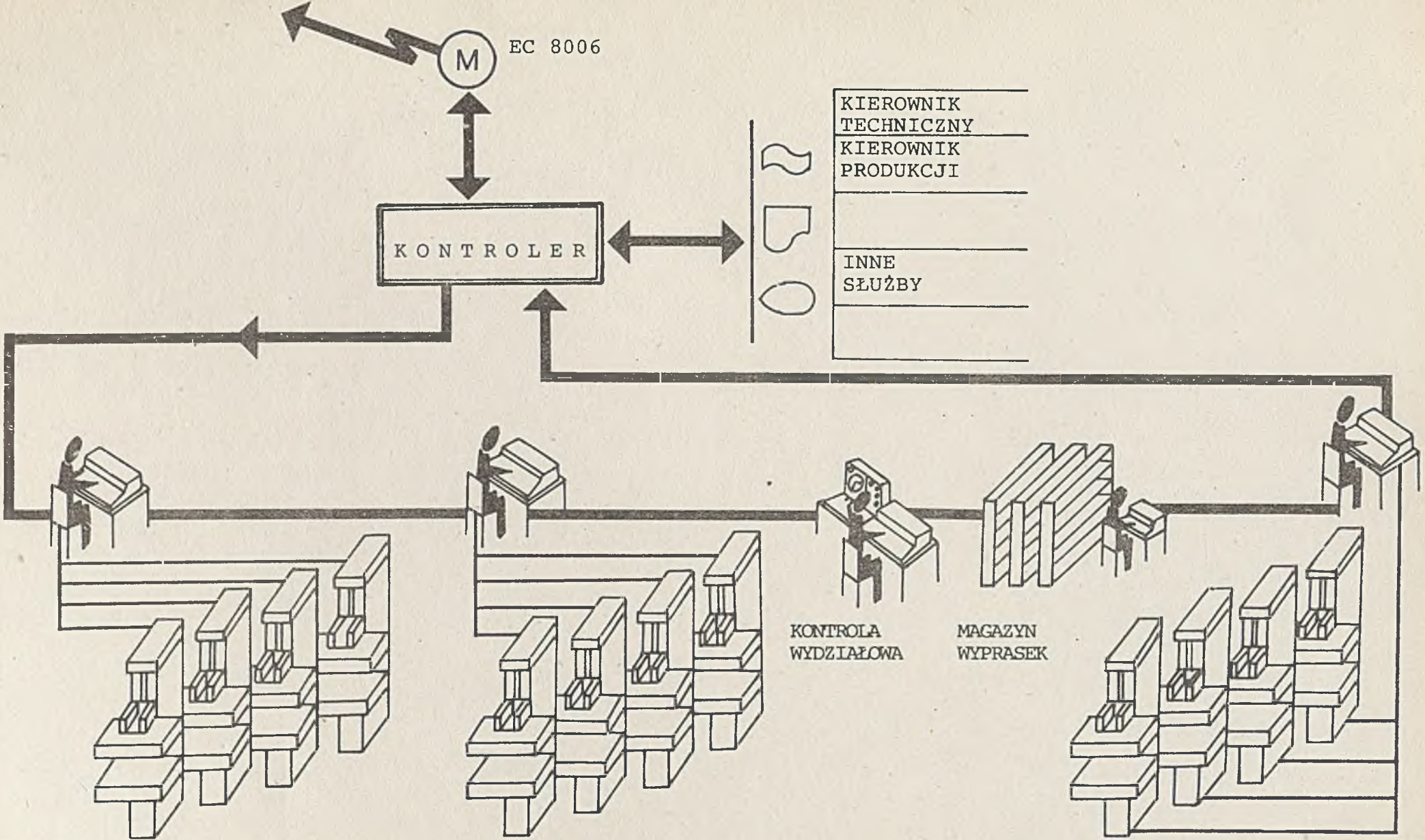
EC 8371.01

M

EC 8006

KONTROLER

KIEROWNIK TECHNICZNY
KIEROWNIK PRODUKCJI
INNE SŁUŻBY



LINIA PRAS 1

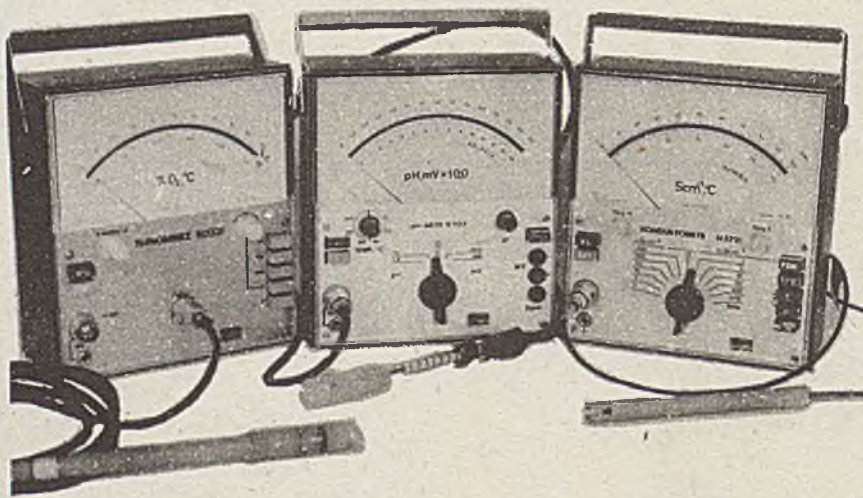
LINIA PRAS 2

LINIA PRAS „N”

Podsystem terminali w zastosowaniu do zbierania danych na wydziale tłoczni

BIBLIOTEKA GŁÓWNA
Politechniki Śląskiej

P 2900/80




MERA