



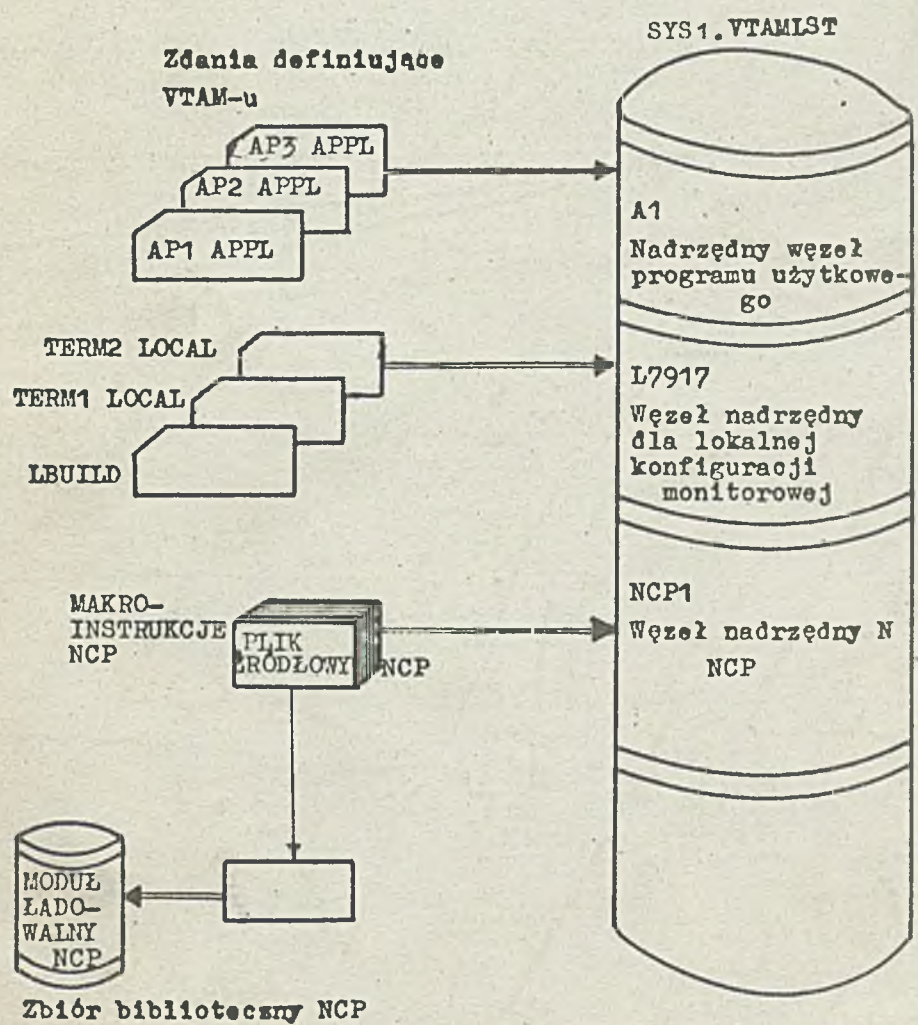
R3057/85

3  
'85

# techniki komputerowe



## BIULETYN INFORMACYJNY



Rysunek na okładce zaczerpnięty z artykułu:

Kurbczycki T., Orzeł T., Plucińska M. - Oprogramowanie  
elementarnej sieci terminalowej - TELEIMM utworzonej  
na bazie sprzętu IBM i JS, s.37



P.3057/85

# TECHNIKI KOMPUTEROWE

Rok XXIII

Nr 3

1985

## Spis treści

	str.
KLIMOWICZ J.: Przegląd metod testowania urządzeń cyfrowych zawierających układy wielkiej skali integracji .....	3
MOCAJA J., STREMBICKI I.: Problemy nauczania użytkownika komputerów na przykładzie doświadczeń szkolnictwa ZSRR .....	27
WUROZYCKI T., ORZEŁ T., PLUCIŃSKA M.: Oprogramowanie elementarnej sieci terminalowej - TELEIMM utworzonej na bazie sprzętu IBM i JS .....	37
ROSZKOWSKA E.: System maszyn wirtualnych VM/JS-P .....	53
Sprawozdania - opr. J. Pelo .....	65
Nowości techniczne - opr. J. Ryżko .....	71



DWUMIESIĘCZNIK

Wydaje:

INSTYTUT MASZYN MATEMATYCZNYCH

Branżowy Ośrodek Informacji Naukowej Technicznej i Ekonom.

Komitet Redakcyjny

dr inż. Stanisława BONKOWICZ-SITTAUER (redaktor naczelny),

mgr Hanna DROZDOWSKA (sekretarz redakcji),

mgr inż. Zdzisław GROCHOWSKI, mgr inż. Zygmunt HAUSWIRT,

mgr inż. Jan KLIMOWICZ, dr inż. Piotr PERKOWSKI,

mgr inż. Romuald SYNAK

Adres redakcji: ul. Krzywickiego 34, 02-078 Warszawa,  
tel. 28-37-29 lub 21-84-41 w. 244

mgr inż. Jan KLIMOWICZ

Instytut Maszyn Matematycznych

## Przegląd metod testowania urządzeń cyfrowych zawierających układy wielkiej skali integracji

Praca niniejsza jest wynikiem analizy problemów i metod testowania urządzeń cyfrowych zbudowanych z elementów scalonych, wśród nich LSI. Analizy dokonano z punktu widzenia producenta i użytkownika takich urządzeń. Praca nie omawia konkretnych typów testerów i nie porusza szerzej problemów producenta układów scalonych, natomiast przedstawia problemy testowania mikroprocesorów, jako najtrudniejszych w aspekcie testowania elementów LSI oraz mikrokomputerów, najbardziej typowych przedstawicieli urządzeń mikroprocesorowych.

W niniejszej pracy Autor reprezentuje pogląd, że rozwiązaniem problemów testowania współczesnych układów cyfrowych są dwie techniki projektowania, serwisu i kontroli, tj. analiza sygnatur i testowanie in-circuit, które omawia szerzej.

### WYKAZ SKRÓTÓW

MP - mikroprocesor

MC - mikrokomputer

UUT - jednostka testowania /unit under test/

SA - analiza sygnatur /signature analysis/

### I. TESTOWANIE MIKROPROCESORÓW. Przegląd

Tradycyjne metody testowania elementów SSI i MSI zastosowane dla LSI okazują się nieefektywne z kilku przyczyn, a przede wszystkim z powodu złożoności mikroprocesorów i skomplikowanego sterowania ich działaniem. Opracowano kilka nowych koncepcji testowania MP, wyróżniając trzy etapy:

- producent
- użytkownik pierwotny, tzn. wytwórca systemów MP
- użytkownik wtórny, tzn. użytkownik systemów MP

Na każdym z tych etapów testuje się w odmiennych warunkach: różna jest liczba MP, stosuje się różne narzędzia, wykorzystuje inny personel. Różne też są cele testowania.

W najlepszej sytuacji jest producent MP, ponieważ ma do czynienia z dużą ilością elementów tego samego typu, może więc stosować skomplikowany sprzęt testujący i zna bardzo dobrze wewnętrzną

strukturę swych produktów. Tak więc najprostszym rozwiązaniem byłoby pozostawienie testowania producentowi i płacenie za rozwiązanie kwestii wyższą ceną elementu. Nie jest to jednak podejście realistyczne, dzisiejsze bowiem metody kontroli niezawodności nie mogą zagwarantować elementów absolutnie niezawodnych, szczególnie w ostrych warunkach ich eksploatacji. Obecna tendencja rozwoju technologii wykazuje redukcję stopy uszkodzeń o jeden rząd wielkości w każdym czteroleciu, potrzeba testowania poza linią produkcyjną powinna więc stopniowo zanikać.

#### Testy produkcyjne

Producent, znając strukturę wszystkich swych produktów może ułożyć i wykonać testy kompletne, tj. zbiory testów pokrywających wszystkie uszkodzenia /stosownie do wybranego modelu uszkodzeń/.

Testy strukturalne stosowane przez wytwórców MP są zwykle stosunkowo krótkie, odnosi się to szczególnie do MP segmentowych /bit-slice/. Strukturalny test procesorowego elementu Intel 3002 ułożony w Instytucie Maszyn Matematycznych w Pradze składa się np. ze 109 wektorów dla samego procesora i 615 wektorów dla rejestrów /wędrujące 0 i 1/. Strukturalny test Motoroli 6800 ma 850 wektorów, a test 16-bitowego MP Texas Instruments 9900 ponad 10000 wektorów. Test segmentowego MP AMD 2901, który jest przykładem testu częściowo strukturalnego, częściowo funkcjonalnego składa się z 12 000 wektorów testowych. Stosuje on zasadę uczulania modułów /module sensorialization/, którą opiszemy niżej.

Typ sprzętu testującego używanego przez producentów zależy od rodzaju testowania. W zasadzie na każdym MP wykonuje się trzy rodzaje testowania:

- sortowanie płytek /wafer sort/,
- test zmontowanej kostki i
- test krytyczny /burn-in test/.

Pierwsze dwa wymagają kosztownych, specjalnych testerów, używanych przede wszystkim do zdejmowania charakterystyk elementów w fazie uruchamiania produkcji. Po przejściu tej fazy stosować można prostsze testery. Podczas sortowania płytek Intel 8080 podaje się sekwencję 40 000 instrukcji na każdy chip - sprawdza ona każdą komórkę w kostce: mierzone są i weryfikowane jej parametry zmiennoprądowe. Po zmontowaniu kostki dokonuje się wybranych pomiarów statystycznych w temperaturze 85°C i wykonuje się sekwencję 1200 testów. Sprawdzono doświadczalnie, że test ten na 100 000 sztuk 8080 i 8080A nie wykrył tylko jednego defektu tranzystora. W końcu puszcza się testy krytyczne /burn-in/ i niezawodnościowe /system life tests/. Testy te wykonywane są w komorze testowej pod kontrolą prostego sterowania znajdującego się poza tą komorą. Sterowanie to zapewnia mikrokomputer oparty na I8080, który może sprawdzać do 196 MB przez określony zadany czas /1000, 5000 godzin lub dłużej/.

#### Pierwotny użytkownik

Użytkownik jest w sytuacji o wiele trudniejszej niż wytwórca, ponieważ zwykle testować musi kilka różnych typów MP w małych ilościach. Nie może zastosować drogiego specjalistycznego testera i nie posiada informacji koniecznych do generacji strukturalnego testu kompletnego. Z tego powodu wielu użytkowników testuje swoje MP w odrębnych firmach, które specjalizują się w testowaniu i dlatego mogą osiągnąć wysoki stopień perfekcji i ekonomii.

Jeśli użytkownik decyduje się na instalację sprzętu do kontroli dostaw MP musi podjąć kilka decyzji. Przede wszystkim kupić lub zbudować tester. Użytkownicy o niskich dostawach preferują stosowanie standardowych testerów rozszerzając je o pewien dodatkowy własny sprzęt i rozwijając własne metody testowania.

Najprostszą metodą testowania jest sprawdzanie pakietów mikrokomputerowych w warunkach normalnego sterowania programem. Metoda ta ma jednak wiele wad, które omówimy dalej. Dlatego też użytkownicy wybierają na ogół nieco mniej problematyczną metodę testowania opartą na stosowaniu prostych testerów. Najczęściej spotykane testery "własnej roboty" wykonują tzw. testy porównawcze. Sieć wzorcowa /sprawdzony poprawny MP/ i sieć badana zasilane są wspólnym źródłem sygnałów i oceniane za pomocą zestawu komparatorów. Stopień złożoności testera zależy od wymaganej dokładności testu. Problem synchronizacji jest tu krytyczny. Oba MP muszą startować z tego samego stanu początkowego i zmieniać swe stany stosownie do wspólnego zegara. Jeśli chcemy wykonać test marginesów, musimy podać go na oba MP, co może dać różne rezultaty. Jeśli zdarza się niezgodność, trudno rozstrzygnąć, czy przyczyną jest rzeczywiste uszkodzenie, czy dopuszczalna różnica tolerancji jakiegoś parametru.

Jeśli stosuje się algorytmicznie generowane pobudzenia prosty specjalistyczny tester może składać się z kilku zaledwie rejestrów i układu pamięci. Algorytmiczny generator pobudzeń ma proste sterowanie mikroprogramowe pozwalające opisać długie sekwencje testujące regularne układy macierzowe w MP /rejestry, liczniki/ za pomocą kilku mikroinstrukcji. Mikroinstrukcje te generują także wzorcowe odpowiedzi, które podawane są na zestaw komparatorów.

Zadaniem najtrudniejszym jest znalezienie takiej sekwencji pobudzeń, która stanowiłaby test kompletny. Użytkownik kupujący tester wysokiej klasy unika kłopotu generacji testów, bowiem testy najczęściej spotykanych MP są zwykle w zestawie standardowego oprogramowania dostarczanego wraz z takim testerem. Testy te są przeważnie typu funkcjonalnego, a ich wykonanie dla jednego MP trwa kilka lub kilkanaście sekund.

Z przyczyn, o których była już mowa wyżej użytkownik może zazwyczaj wygenerować tylko test funkcjonalny, który nie pokrywa wszystkich uszkodzeń wewnętrznych, choć zwykle jest dłuższy niż test strukturalny. Pisanie testów MP różni się od pisania testów innych układów logicznych. Najważniejszą różnicą jest język wejściowy, który trzeba opanować przy każdym nowym MP. Metodą najlepiej nadającą się do ręcznego przygotowywania sekwencji testowych MP jest tzw. uczulanie modułów /module sensorialization/. Łączy ona podejście strukturalne z funkcjonalnym i jest w istocie tylko rozszerzeniem dobrze znanego modularnego podejścia, które stosuje się przy złożonych układach logicznych. Oparta jest na analizie architektury sprzętu i określeniu programowej odpowiedzi MP. Na tej podstawie konstruktor testu pisze w języku programowania MP zbiór sekwencji testowych sprawdzających kolejno, jeden za drugim moduły MP. Ponieważ jednak ich wewnętrzna struktura jest nieznana, testowane są jedynie funkcjonalnie. Dla Intela 8080 rezultatem takiego podejścia jest testowanie modułów w następującej kolejności: zerowanie, licznik programu, pamięć, wskaźnik stosu, ALU, akumulator, zegar i sterowanie, dekodery rozkazów. Napisanie wszystkich pobudzeń koniecznych w takiej sekwencji i zapis ich w pamięci jakiegoś typu jest przedsięwzięciem bardzo kosztownym. Dlatego też należy łączyć metody algorytmicznej generacji pobudzeń i uczulania modułów.

#### -Wtórny użytkownik

Użytkownik systemu MP potrzebuje na ogół prostych programów kontrolnych, które powiedzą mu, czy MP i inne istotne elementy pracują poprawnie. Preferuje on więc raczej test wykonywany w naturalnych warunkach niż kupowanie specjalnego testera. Dwa są sposoby testowania MP w warunkach naturalnych:

- "na pakiecie"/on-board/,
- "w systemie" /in-system/.

Testy "on-board" wykonywane są zwykle przez tester pakietów podający pobudzenie i oceniający odpowiedzi. Na ten typ testowania pozwalają tylko najnowocześniejsze testery, przede wszystkim z powodu wymaganej wysokiej częstotliwości podawania pobudzeń i przełączania dwukierunkowych szyn. Poza tym zestaw testujący z dostępem do złącza pakietu nie może na ogół pomieścić kompletnego testu MP z powodu jego długości.

Specjalnym rodzajem testów "on-board" są testy "w układ" /in-circuit/ wykonywane za pomocą "łóżka fakira" /bed of nails/. Bezpośredni dostęp do wszystkich elementów pozwala testować je w zasadzie w ten sam sposób, jakby były niezamontowane w pakiecie. Aby uniknąć trudności pamiętania pełnych odpowiedzi ciągi wyjściowe redukuje się często używając sygnatur /zob. analiza sygnatur/.

Najbardziej naturalnym otoczeniem MP jest system, w którym on normalnie pracuje. System ten może być używany do testowania bez żadnych dodatkowych narzędzi, ponieważ zwykle zawiera urządzenia komunikacji z operatorem. Można tu stosować albo pewne programy użytkowe, których poprawne wyniki znamy, albo specjalne programy diagnostyczne.

Stosowanie programów użytkowych w charakterze sekwencji testujących jest rozwiązaniem najprostszym i najtańszym, ma jednak pewne wady. Przede wszystkim mamy prawie pewność, że pokrycie uszkodzeń nie jest kompletne, tzn. że uszkodzony MP może łatwo przejść taki test. Poza tym często trudno jest stworzyć dokładnie te same warunki, które zaistnieją podczas zamierzonej eksploatacji, np. w systemach sterujących czasem rzeczywistego. Pomimo wszystkich tych wad testowanie za pomocą programów użytkowych jest często stosowane, głównie jako zło konieczne, gdy nie ma innych rozwiązań do dyspozycji.

Jedynie wartościowe testowanie "w systemie" wykonać można tylko za pomocą specjalnych programów diagnostycznych. Podobnie jak programy użytkowe, wystartować one mogą tylko wtedy, gdy MP jest w wymaganym zakresie funkcji sprawny. Inaczej jednak niż programy użytkowe są one napisane w ten sposób, że w pierwszej kolejności sprawdzają najprostsze instrukcje, następnie coraz bardziej skomplikowane, aż sprawdzony zostanie cały MP. Ta metoda, znana w konwencjonalnych komputerach jako zasada rozszerzania kontroli, była już szczegółowo opisywana w zastosowaniu do mikroprocesorów.

## Wnioski

Wybór odpowiednich metod testowania jest przede wszystkim kwestią ekonomii. Zwrot nakładów na testowanie następuje na ogół bardzo szybko, nie ryzykujemy więc wiele inwestując w tę dziedzinę. Ważne jest jednak porównanie kosztów testowania z kosztami awarii systemu. Wybór testera i metody testowania powinny być rezultatem takiego porównania.

## II. TESTOWANIE MIKROKOMPUTERÓW. Przegląd

### Charakterystyka mikrokomputera jako UUT /jednostki testowanej/

- Wiele charakterystycznych cech MC stanowi nowość w dziedzinie technik cyfrowych, a szczególnie:
- obecność elementów LSI. Jest to niewątpliwie najbardziej charakterystyczna cecha wszystkich MC. Wśród kostek LSI największej uwagi skupił MP, są jednak inne elementy LSI, których złożoność jest co najmniej porównywalna, np. kontrolery urządzeń zewnętrznych, układy komunikacyjne i duże kostki RAM i ROM. Test kompletny takich elementów LSI jest możliwy tylko w specjalnym testerze, po wyjęciu ich z MC.
  - Synchronizacja MC. Typowy MC ma wbudowany generator zegara działający na ogół niezależnie od otoczenia - nie może on być sterowany czy zatrzymywany z zewnątrz. W wypadku dynamicznych elementów



MOS nie wolno go nawet zwolnić poza określoną granicę /np. dla INTEL-a 8080 - poniżej 500 kHz/ - ginie bowiem informacja pamiętana w rejestrach wewnętrznych.

- Stosowanie dwukierunkowych szyn. Ta cecha, choć nie nowa, zdyskredytować może wiele testerów, ponieważ pewne grupy kontaktów złącz - zależnie od wykonywanej instrukcji, muszą być traktowane jako wejścia lub jako wyjścia. Przełączanie kierunku transmisji musi być bardzo szybkie, na ogół w ciągu jednego cyklu zegara /tzn. ułamka mikrosekundy/.
- Obecność elementów niecyfrowych. Ogromny wachlarz elementów elektronicznych daje projektantowi możliwość umieszczenia na pakiecie MC także pewnych układów pomocniczych i interfejsowych, są to np. układy analogowe i hybrydowe, konwertery A/D i D/A i moduły zasilania. Metody ich testowania różnią się w większości od metod stosowanych dla sieci cyfrowych.
- Ograniczony dostęp do wewnętrznych węzłów sieci. Współczesna kostka LSI stanowi ekwiwalent dawnego pakietu, a współczesny pakiet - dawnego systemu. Wielka liczba złącz systemowych pozostaje więc wewnątrz jednego pakietu i staje się niedostępna z zewnątrz. Poza tym, pakiety MC często w ogóle nie mają złącz, ponieważ nie są nigdzie wsuwane. Tak więc liczba punktów dostępu jest niewystarczająca dla przeprowadzenia diagnostyki i trzeba zapewnić pomocniczy dostęp do pakietu MC.
- Wbudowana inteligencja. Obecność MP i pamięci w MC stanowi o godnej uwagi ilości wbudowanej inteligencji. Może to skomplikować testowanie, ale może je także ułatwić, jeśli z inteligencji tej zrobi się właściwy użytek. Zdolność wykonywania zapamiętanego programu daje możliwość autonomicznego wykonywania procedur diagnostycznych MC, bez stosowania zewnętrznego testera. Zasada rozszerzania kontroli /bootstrapping/ stosowana w kilku już generacjach procesorów jest podstawą organizacji diagnostyki tego typu.
- Odmienne modele uszkodzeń. Duże upakowanie elementów na płytkach LSI sprawia, że klasyczne modele s-a- $\beta$ , s-a-1 stają się nieadekwatne, nawet jeśli rozszerzymy je o zwarcia i błędy przemijające. Nowe typy uszkodzeń, określane jako czułość na informację /pattern sensitivity/, czułość na rozkazy /instruction sensitivity/ i uszkodzenia dynamiczne muszą być brane pod uwagę przy testowaniu. Te nowe typy uszkodzeń w połączeniu z występowaniem dwukierunkowych szyn sprawiają, że lokalizacja uszkodzenia z dokładnością do kostki staje się bardzo trudnym zadaniem.

#### Podstawowe charakterystyki systemu testującego dla MC

Z dokonanego krótkiego przeglądu cech MC wynikają najistotniejsze charakterystyki systemu testującego MC. Diagnostyka ma zawsze trzy fazy, a mianowicie: generację testu, wykonanie testu i ocenę testu /tj. lokalizację uszkodzenia/. W każdej z tych faz obecność MP w UUT powoduje komplikację algorytmów i sprzętu.

Nie można generować ręcznie zbioru testów dla MC, ponieważ jest on odpowiednikiem kilku dziesiątków tysięcy elementarnych bramek, a złożoność procedur generacji testów rośnie w przybliżeniu wykładniczo wraz ze złożonością UUT. Rozwiązanie idealne, tj. automatyczny system generacji testów stosowalny do MC dotychczas nie istnieje, a producenci systemów testowych odnoszą się sceptycznie do możliwości jego stworzenia. W tej sytuacji za wartościowe uważa się rozwiązania oparte na konwencjonalnym systemie symulacji, dającym operatorowi możliwość weryfikacji jego wektorów testowych i generacji pewnych bloków testów. Takie wstępne przetworzenie zbioru testowego konieczne jest zarówno przy testowaniu stand-alone, jak i testowaniu zewnętrznym.

Wybór poziomu symulacji zbiorów testowych MC stanowi bardzo poważny problem, każda kostka LSI reprezentuje bowiem sama kilka tysięcy elementarnych bramek. Tworzenie modelu pakietu na poziomie bramek elementarnych /zwykle najlepszy sposób reprezentacji dla układów SSI i MSI/ nie jest możliwe

z powodu ograniczonej pamięci operacyjnej i długiego czasu symulacji. Jedynym możliwym do przyjęcia rozwiązaniem jest funkcjonalna symulacja bloków, które nie są w danym momencie testowane, ale których działanie w czasie testu musi być znane. Zastąpienie opisu strukturalnego opisem funkcjonalnym przyspiesza symulację około dziesięciokrotnie. Wśród istotnych własności systemów symulacji dla MC wymieńmy także zdolność modelowania rzeczywistych czasów opóźnień elementów logicznych.

Zamiast generacji i symulacji testu kompletnego dla MC można zastosować prostsze i tańsze metody testowania. Najlepiej znane są wśród nich metody testowania pseudolosowego i testowania aplikacyjnego /zadań kontrolnych/. Sekwencja pseudolosowego testu musi być przede wszystkim celowa funkcjonalnie, tj. musi reprezentować taki program MP, którego wykonanie nie wprowadzi MP w stan, z którego nie ma wyjścia. Testowanie aplikacyjne za pomocą programów użytkowych jest najtańszym i najprostszym substytutem testu kompletnego. Jego wady są oczywiste i znane: brak gwarancji kompletności i zależność od otoczenia MC.

Najważniejszym elementem sprzętowym systemu testującego jest tester. W roku 1977 na rynku oferowano tylko trzy testery pakietów MC. W rok później oferowano już dziewięć i liczba ta stale rośnie. Podstawowe parametry techniczne typowego testera MC:

- możliwość wykonywania testów złożonych z kilku tysięcy pobudzeń testowych z szybkością zmienną od 1 do 20 MHz,
- złącze o co najmniej 256 stykach,
- zdolność szybkiego przełączania na stykach związanych z szynami informacyjnymi.

Programowe sterowanie zapewnia ustawianie i pomiar kilku różnych poziomów napięć. Na jeden pakiet podawać można kilka przebiegów zegarowych – mogą być one używane jako źródła niezależne lub zsynchronizowane z UUT. Przedziały czasu mogą być mierzone z dokładnością do kilku nanosekund. Szeroko stosuje się sprzętowe i programowe narzędzia lokalizacji uszkodzeń, takie jak np. wielostykowe klipsy logiczne i słowniki uszkodzeń.

#### Organizacja procesu testowania

Zanim scharakteryzujemy najistotniejsze procedury testowania MC musimy najpierw określić, kto i po co testuje. Ogólnie biorąc są trzy typowe okoliczności, w których pożądanym jest testowanie MC:

- testowanie inżynierskie /podczas projektowania nowego produktu/,
- testowanie produkcyjne,
- testowanie serwisowe.

Każda z tych sytuacji stanowi w istocie odrębny problem inżynierii testów.

Podczas testowania inżynierskiego uruchomić trzeba zarówno sprzęt, jak i oprogramowanie. Jest to zadanie szczególne i dla jego rozwiązania wynaleziono różne urządzenia pomocnicze, np. analizatory stanów logicznych, emulatory in-circuit czy mikroprocesorowe systemy wspomaganie projektowania. Podstawowym celem na tym etapie jest dokładne sprawdzenie; koszty i czas testowania schodzą na drugi plan. Ludzka inwencja i doświadczenie są tu niezbędne; trudno sobie wyobrazić całkowitą automatyzację testowania na tym etapie.

Testowanie produkcyjne, zwłaszcza w produkcji seryjnej, stanowi dla diagnostyki otoczenie całkowicie odmienne. Powtarzalność procesu podawania testów uzasadnia wymaganie bardziej wyrafinowanych metod generacji testów i pełnej automatyzacji wykonywania testów. Typowym rozwiązaniem w testowaniu produkcyjnym jest podawanie testu z zewnętrznego testera. Jak można sądzić z publikacji, jest to metoda najczęściej stosowana. Złożoność MP i innych elementów LSI sprawia, że przeważnie niemożliwe jest przetestowanie całego pakietu MC w jednym przebiegu testowym. Wprowadzono więc kilka faz testowania. Wykonywanie testu MC w kilku fazach stwarza możliwość podziału testu MC na kilka segmentów, z których każdy staje się łatwiejszy do zaprogramowania, wykonania i oceny.

Test trójfazowy składa się z przetestowania MP w specjalnym testerze, przetestowania pakietu MC bez MP i na koniec przetestowania kompletnego pakietu z włączonym MP. W pierwszej fazie wykonać można kompletny test MP /jest to zwykle test dynamiczny w warunkach worst-case/. Kompletność jest oczywiście uwarunkowana znajomością rzeczywistej wewnętrznej struktury testowanego MP. Przetestowanie pakietu MC bez MP oznacza albo fizyczne usunięcie MP z pakietu, albo jego izolację elektryczną. To ostatnie można osiągnąć albo przez wprowadzenie trójstanowych szyn w stan wysokiej impedancji, albo przez tzw. impulsy o wysokiej energii. Pozostałą część MC testuje się statycznie lub dynamicznie. W trzeciej fazie sprawdza się tylko współdziałanie MP z innymi elementami, co można osiągnąć przy częściowym wykorzystaniu programu użytkowego, np. zliczając zbrocza impulsów /transition counting/ na złączach.

Test dwufazowy różni się od trójfazowego zazwyczaj ominięciem drugiej fazy. Test całego pakietu MC jest oczywiście dokładniejszy niż poprzednio, ponieważ teraz sprawdzić trzeba, zarówno działanie wszystkich elementów, jak i ich współpracę. Niektóre rozwiązania zalecają stosowanie w drugiej fazie trybu krokowego. MP włączony w pakiet nie pracuje ze swą normalną szybkością, chociaż zegar biegnie normalnie - MP jest zatrzymywany sygnałem na wejściu READY po wykonaniu każdego kroku. Kolejność wykonywanych kontroli jest następująca:

- zawartość ROM,
- zapis/odczyt RAM,
- inne funkcje adresowe MP,
- podstawowe operacje MP.

Test jednofazowy stawia testerowi największe wymagania, ponieważ w jednym przebiegu sprawdzany jest cały pakiet MC z MP włącznie, a testowanie prowadzi się z szybkością normalnej pracy. Wyjścia UUT ocenia się przez porównanie z wyjściami wzorcowego dobrego pakietu albo z zapamiętanymi odpowiedziami wzorcowymi przesyłanymi w czasie testowania do szybkiego bufora. Przygotowanie testów jednofazowych jest zadaniem bardzo trudnym i jeśli wymagany jest test kompletny wszystkich elementów zastosować trzeba potężny system symulacji.

Wśród metod testowania zewnętrznego wymienić należy także "testowanie partnerskie" /adversary testing/ polegające na zastosowaniu innego procesora, np. minikomputera lub MC, który steruje programem testowym wykonywanym w testowanym MC i ocenia jego wyniki. Ta metoda używana jest przede wszystkim przy badaniach niezawodnościowych i przy testowaniu w specjalnych warunkach klimatycznych.

Zdolność wykonywania pamiętanego w pamięci programu umożliwia testowanie MC w trybie stand-alone, w którym MC stanowi równocześnie tester i UUT. Zanim wystartuje test, uruchomione więc muszą być co najmniej pewne podstawowe funkcje MC. Ta metoda nadaje się dobrze do celów testowania serwisowego, gdy testowane są systemy uprzednio już uruchomione, a wybór pomocniczych środków testowania jest ograniczony. Nie należy jednak rozumieć, że test samodiagnostyczny w ogóle nie używa narzędzi diagnozowania. Mogą one być prostsze niż przy testowaniu zewnętrznym, jednak zapewnione być muszą co najmniej pewne środki komunikacji z operatorem, tj. podawanie danych wejściowych i wyjściowych z pożądaną szybkością.

Najprostszym testem samodiagnostycznym jest wykonywanie programu użytkowego, którego wyniki są znane. Kontrola tych wyników daje naturalnie bardzo niekompletny obraz rzeczywistego stanu technicznego MC, ale może wiarygodnie wykazać jego zdolność do wykonania pewnego specyficznego zadania.

Zastosowanie programu użytkowego w charakterze testu jest w istocie problemem wyboru danych testowych. Każdy konkretny zestaw danych wejściowych powoduje wejście programu w jakąś konkretną gałąź sieci działań. Nawet jeśli potrafimy wygenerować dane konieczne dla przejścia wszystkich możliwych gałęzi, pozostaje problemem obserwacja wyników i oczywiście - wykonanie testów w rozsądnym czasie. Zaproponowano tu koncepcję punktów obserwacji i komend. Punkt obserwacji jest instrukcją,

której wyniki mogą być obserwowane na wyjściu systemu. Punkt komendy jest instrukcją wprowadzenia danych lub instrukcją odczytu z RAM danych, które nie były zapisane przez program. Program zostaje podzielony na fazy, z których każda zaczyna się punktem komendy i kończy punktem obserwacji. Podczas wykonywania programu MP może wejść w fazę tylko przez punkt komendy i wyjść z niej tylko przez punkt obserwacji. Punkty komendy i obserwacji są momentami, w których MC komunikuje się ze swoim otoczeniem, tj. z "testerem" - niezależnie od tego, co sobą ten tester przedstawia. Sam test składa się z wykonywania faz programu przy różnych danych i porównywania rezultatów otrzymywanych z poprawnymi wynikami.

Bardziej wiarygodne i bezpośrednie oceny technicznego stanu MC dają specjalne programy testowe napisane zgodnie z regułami diagnostyki systemu. Ich pisanie jest trudnym i żmudnym zadaniem obejmującym dokładne studium mechanizmów uszkodzeń i wewnętrznej struktury wszystkich stosowanych elementów.

Przykładem może być organizacja systemu testów oparta na zastosowaniu zbiorów testów rezydentnych i nierezydentnych. Testy rezydujące w ROM wykonują przede wszystkim periodyczne sprawdzenia, tj. prowadzą detekcje uszkodzeń ze stosunkowo ograniczonym pokryciem. Testy te są napisane z zachowaniem zasady uczulania modułów, poczynając od najprostszych /tzn. najbardziej niezawodnych/ instrukcji. Uruchamiane są one przez nadzorczy program diagnostyczny w każdym stanie oczekiwania, tj. wtedy, gdy nie wykonuje się żaden fragment programów użytkowych. Jeśli wykryty zostanie błąd, ładowana jest z pamięci zewnętrznej do RAM i wykonywana odpowiednia sekcja lokalizacji uszkodzenia. Wyniki testu oceniane są za pomocą słownika uszkodzeń, który na ogół zapewnia rozdzielczość diagnostyczną z dokładnością do kostki. Taka organizacja systemu testów, przypominająca dawne, klasyczne, konwencjonalne systemy testów dużych maszyn, jest cennym narzędziem dla serwisu. Koszt takiego systemu jest jednak bardzo wysoki i może być usprawiedliwiony tylko żądaniem wysokiej niezawodności i dyspozycyjności testowanej instalacji.

### III. ANALIZA SYGNATUR - PROJEKTOWANIE I SERWIS URZĄDZEŃ MIKROPROCESOROWYCH

#### Wstęp

Jednym ze sposobów rozwiązania problemów serwisu urządzeń mikroprocesorowych jest wbudowywanie w produkt układów ułatwiających serwis. Projektanci zwykle nie znają jednak wymagań, jakie stawia serwis urządzeń mikroprocesorowych.

Niektóre cechy takich urządzeń stanowią problem dla tradycyjnego sprzętu testowego, np. trudno określić charakterystykę sieci, ponieważ często jej działanie jest niejako ukryte w oprogramowaniu. Podobnie problemem jest dynamiczne działanie urządzenia, gdyż sygnały często aktywizują się na kilka mikrosekund, po czym znikają, np. w klawiaturze sterowanej mikroprocesorowo. Kontrola błędów sygnałów wymaga wiedzy nie tylko o tym, gdzie szukać, ale i kiedy szukać.

Trzecim problemem jest dwukierunkowość szyn procesora, która utrudnia bardzo interpretację informacji - adresów i danych. Trudności potęguje równoległa struktura szyn, sumujących logicznie wiele urządzeń, co wymaga żmudnej detekcji uszkodzeń. Co więcej, urządzenia testujące muszą obsłużyć wiele operacji, dzisiejsze urządzenia mogą bowiem wymagać przejścia procesora przez kilka tysięcy kroków w cyklu pomiarowym, podczas gdy dawne wymagały zwykle nie więcej niż 100 operacji. Częściowym rozwiązaniem tych problemów są analizatory stanów logicznych, które pozwalają prześledzić algorytm działania MP za pomocą badania sekwencji stanów maszyny. Kiedy zlokalizuje się stan, w którym po raz pierwszy pojawia się uszkodzenie, stosuje się tradycyjny sprzęt testujący w celu znalezienia

elementu, w którym błąd powstaje. Procedura ta jest jednak czasochłonna i wymaga wysoko wykwalifikowanego personelu.

Uszkodzenia dotyczące MP i układów interfejsowych procesora są trudne do wykrycia i lokalizacji. W praktyce identyfikacja tych uszkodzeń za pomocą tradycyjnego sprzętu i metod diagnostycznych może okazać się niemożliwa. Wielogodzinne badania nie dają odpowiedzi, czy błąd wynika z uszkodzenia sterowania w MP, czy z uszkodzenia interfejsu, czy z uszkodzenia jakiegoś innego układu; z tego powodu niemożliwe jest zakończenie pomiaru lub można wejść w niewłaściwy segment programu.

Aby umożliwić serwis, projektanci urządzeń mikroprocesorowych muszą zastosować nową metodykę projektowania wbudowanej "serwisowości". Rozwiązania szukać powinni wśród trzech metod, które dają szanse przezwyciężenia problemów diagnostyki. Są to:

- analiza sygnatur,
- wbudowane procedury diagnostyczne
- odwzorowanie /mapping/.

#### Analiza sygnatur

Analiza sygnatur stosuje kompresję danych, aby zredukować złożone, szeregowo strumienne dane dowolnej długości, pojawiające się w pewnym węźle układu do jednoznacznej, czterocyfrowej heksadecymalnej sygnatury. Porównywanie sygnatury podejrzanego węzła sieci z określoną empirycznie poprawną sygnaturą podaną w instrukcji serwisowej pozwala szybko sprawdzić działanie układu. Przyczyna błędu sygnatury może być wykryta przez próbkowanie odpowiednich węzłów - obserwację złych i dobrych sygnatur na drodze sygnału.

Poza danymi czerpanymi z badanego węzła, czytnik sygnatur wymaga trzech sygnałów. Sygnał startu otwiera okno pomiarowe, w którym dane próbkowane są zegarem, a wyniki podawane na 16-bitowy rejestr przesuwny z liniowym sprzężeniem zwrotnym wewnątrz analizatora. Sygnał zegara generowany przez UUT synchronizuje dane i pracę analizatora sygnałów. Sygnał stopu zamyka okno pomiaru, a 16-bitowa reszta w rejestrze przesuwym pojawia się na monitorze jako 4-cyfrowa sygnatura heksadecymalna /rys.1/.

Koncepcja zapisu przebiegów krótkim kodem, wzięta z praktyki obsługi urządzeń analogowych, otwiera możliwości szerokiego wykorzystania takiej notacji dla opisu poprawnego działania urządzeń od schematu począwszy, a na trawieniu wzorcowych sygnatur na pakietach skończywszy.

Dwa podstawowe zastosowania analizy sygnatur w urządzeniach mikroprocesorowych, to:

- bieg jałowy /free-running/,
- i analiza sterowana programowo /software-driven analysis/.

#### Bieg jałowy

Analiza na biegu jałowym polega na wymuszeniu cyklicznego przebiegu mikroprocesora przez całe pole jego adresów, przy czym sygnały startu i stopu brane są z szyny adresowej. Analiza sterowana programowo polega na generacji za pomocą pobudzeń sygnałów startu i stopu zapisanych w pamięci stałej i pisaniu powtarzalnych ciągów danych na szynę danych w celu sprawdzenia układów podłączonych do procesora.

Tryb biegu jałowego nie wymaga miejsca w ROM, natomiast tryb sterowania programami wymaga na ogół mniej niż 5% zajętości ROM. Tryb programowy sprawdza jednak większy procent sieci, niż bieg jałowy. Przy wymaganiu starannego testowania obie metody mogą być zrealizowane w tym samym wyrobie.

Analiza biegu jałowego sprawdza w zasadzie działanie jądra /=kernel/: minimalnej konfiguracji MP, ROM i RAM koniecznej do cyklicznego przebiegania pola adresowego.

Przykładem efektywnej implementacji analizy sygnatur na biegu jałowym jest mikrofalowy licznik częstotliwości MP5342A: w układzie procesora dodano jedynie kilka przełączników i oporników.

#### Analiza sterowana programowo

Zapisanie w ROM specjalnych wysterowań zapewnia dokładniejszą analizę sygnatur niż tryb biegu jałowego. Wysterowania te zapewniają pobudzenie układów znajdujących się poza jądrem. Pozwalają na diagnostykę rejestrów wejściowych urządzeń przyjmujących dane z MP i dają sygnatury układów związanych z danymi w tych rejestrach.

Typowy 8-bitowy wyjściowy szablon testowy składa się z samych 0, samych 1 i wędrującej jedynki /10000000,01000000,...00000001/. Procesor przesyła pierwszy bajt szablonu na szynę danych i każe przyjąć bajt w jakimś miejscu lub rejestrze. Jeśli odbiornik ma np. 12 miejsc, procesor zapisuje ten bajt do całej dwunastki - rozsyła tak kolejne bajty do wszystkich odbiorników cyklicznie. Jako start/stopu używa się najstarszego bitu adresu ustawionego sztucznie na początku programu testowego.

Programowo-sterowana analiza sygnatur może także sprawdzać wejścia z układów, które nie są częścią jądra. Wbudowany przełącznik serwisowy może ustawiać MP w reżimie kontroli wejść. Taki test polega na podawaniu z badanych układów danych z każdego miejsca ich pamięci na szynę danych celem sprawdzenia przez analizę sygnatur. Przy sprawdzaniu operacji odczytu szczególnie ważne jest zapewnienie właściwej inicjacji sieci.

#### Zasady projektowania z analizą sygnatur

Doświadczenie projektowania urządzeń mikroprocesorowych, które mają współpracować z analizatorami sygnatur pozwala sformułować sześć zasad, których przestrzegać trzeba w procesie projektowania:

- wyróżnić jądro /kernel/,
- zapewnić sposoby rozcinania lokalnych ścieżek sprzężenia zwrotnego,
- zapewnić inicjację testowanych pakietów,
- wykorzystywać dekodowanie adresów przy identyfikacji uszkodzeń ROM,
- stabilizować sygnatury urządzeń trójstanowych,
- starannie dokumentować sygnatury.

- ⊙ Jądro powinno być na osobnym pakiecie; jednym z pierwszych bowiem kroków diagnostycznych jest sprawdzenie jałowego biegu jądra. Testu tego nie można przeprowadzić, póki jądro nie zostanie kompletnie wyizolowane. Jeśli osobny pakiet nie wchodzi w grę, zapewnić trzeba przełączniki odłączające elementy jądra od reszty systemu.
- ⊙ Projektant musi zapewnić możliwości rozwarcia ścieżek sprzężeń zwrotnych. Jeśli sprzężenia nie byłyby przecięte podczas wysyłania pobudzeń do urządzenia przez program analizy sygnatur, to błąd propagowałby się przez pętlę sprzężenia i wszystkie kolejne sygnatury byłyby przekłamane. Dobrą metodą jest wytypowanie do rozcięcia oczywistych sprzężeń zwrotnych, takich, jak szyna danych jądra, gdzie wstawienie dodatkowych przełączników zapewnia rozwarcie ścieżki.
- ⊙ Inicjacja testowanych pakietów jest niezwykle ważna w analizie sygnatur. Bez inicjacji nie można zagwarantować zgodności sygnatur różnych poprawnych egzemplarzy urządzenia, a nawet sygnatur tego samego urządzenia po kolejnych włączeniach zasilania. Niektóre układy inicjują

się same własnymi impulsami zerującymi po przejściu jednego kompletnego cyklu. Dla innych projektant musi zapewnić w podręczniku serwisowym dokumentację procedury inicjującej, którą pracownik serwisu musi wykonać. Podobnie zapewniona musi być możliwość blokady układów asynchronicznych, jak uniwbiratory i układy przerwań. Gdy linie szyn prowadzą do kilku urządzeń należy unikać pakietów wielowarstwowych - na pakietach takich bowiem trudno za pomocą sondy stwierdzić, które urządzenie wymusza stały wysoki lub niski stan linii szyny.

- ⊙ Projektant może ułatwić identyfikację uszkodzenia ROM używając dla generacji okna pomiarowego każdej kostki ROM sygnałów dekodera adresu - czasem wymaga to dodatkowych elementów. Sygnał start/stop dla każdego ROM-u odpowiada jego pierwszemu/ostatniemu adresowi i sygnatury pojawiające się na szynie danych pochodzą kolejno zawsze tylko od jednego urządzenia.
- ⊙ Układy trójstanowe stwarzają szczególne trudności odczytania stabilnych powtarzalnych sygnatur. Niepowtarzalność sygnatur występować może wtedy, gdy okno pomiarowe obejmuje przełączenie wyjscia ze stanu wysokiej impedancji do stanu "enable", a na wyjściach nie ma oporników podpierających.
- ⊙ Obowiązkiem projektanta jest staranna dokumentacja sygnatur, które znajdują się w podręczniku serwisowym. Istnieje kilka sposobów upewnienia się o wiarygodności sygnatur dla różnych egzemplarzy. Jednym z nich jest sprawdzenie stabilności i powtarzalności sygnatur na kilku egzemplarzach tego samego urządzenia. Najsurowszym testem jest 50% redukcja częstotliwości zegara. Sygnatury nie powinny się zmieniać - jeśli się zmieniają, oznacza to potencjalny problem synchronizacji, prawdopodobnie związany z przekroczeniem marginesów czasów ustalania się szyn - na ogół wskutek braku oporników podpierających na liniach szyn. Ważne jest, aby zawsze dokumentować dla każdego nowego ustawienia sygnaturę charakterystyczną, która pozwala pracownikowi serwisu upewnić się, że sygnały start/stopu podawane na analizator sygnatur są poprawne. Sygnaturę charakterystyczną otrzymuje się przykładając próbnik w takim punkcie sieci, który pozostaje niezmiennie wysoki /w konwencji TTL/ w ramach okna start/stopowego.

#### Wbudowane procedury diagnostyczne

Drugim po analizie sygnatur środkiem zwiększenia "serwisowalności", z którego szeroko korzystają projektanci są wbudowane procedury diagnostyczne. Stosowane już od wielu lat w maszynach mikroprogramowanych nie stworzyły dotychczas żadnej określonej metodyki, dowolność ich programowania i różnorodność sprawdzanych struktur otwierają bowiem przy każdym nowym projekcie szerokie pole dla inwencji projektanta.

Mikroprogramy diagnostyki podzielić można ogólnie na samoinicjujące się i wywoływane. Pierwsze wykonywane są automatycznie, za każdym razem, kiedy spełniony jest pewien warunek wstępny, np. włączenie zasilania, sygnał awarii, wejście procesora w stan oczekiwania. Procedury diagnostyczne wywoływane wykonywane są tylko na żądanie pracownika serwisu i ich charakter zależy od typu urządzenia. Do ogólnie przydatnych należy tu mikroprogram śledzenia algorytmu. Urządzenie wykonuje swój normalny algorytm operacyjny, ale w jego kluczowych punktach wyświetla jednocześnie mnemoniczne kody, dając w ten sposób rodzaj obserwacji mikroprogramu. Niektóre podprogramy mogą być wywoływane naciśnięciem odpowiedniego klucza pulpitu, inne odpowiednim ustawieniem wewnętrznych przełączników serwisowych.

Użyteczną wywoływaną procedurą diagnostyczną jest też kontrola klawiatury monitora, w której naciśnięcie odpowiedniego klucza powoduje wypisanie związanego z nim znaku. Test taki zapewnia pewien poziom ufności co do używania klawiatury i monitora w innych kontrolach diagnostycznych.

## Mapping jako test go/no-go

Trzecią techniką testową upraszczającą serwis urządzeń mikroprocesorowych jest odwzorowanie /mapping/. Pomaga on pracownikowi serwisu zorientować się, gdzie rozpocząć poszukiwanie uszkodzenia, np. w liczniku impulsów HP5370A pierwsza jest kontrola jądra i związanych z nim szyn, ponieważ muszą one być sprawne, aby można było poszukiwać uszkodzenia w pozostałych zespołach urządzenia - układach liczących, monitorze i obwodach analogowych.

Za pomocą odwzorowania można szybko sprawdzić działanie mikroprocesora i jego komunikację z innymi pakietami urządzenia bez straty czasu na zdejmowanie sygnatur. Jest to jednak ocena typu go/no-go - kiedy więc wskazuje ona złe działanie jądra trzeba zastosować analizę sygnatur lub jakąś inną metodę, aby znaleźć uszkodzony element.

Odwzorowanie jest w zasadzie obrazem szyny adresowej i procesor generuje ten obraz podczas wykonywania swoich normalnych programów. W 5370A zamiast podłączać analizator stanów logicznych do wszystkich 16 linii szyny adresowej podłącza się do niej specjalny pakiet serwisowy z zamontowanymi na nim dwoma przetwornikami cyfrowo-analogowymi, które podają analogowe sygnały na wejścia X i Y oscyloskopu. Każdy punkt matrycy adresów wyświetlany na ekranie reprezentuje jeden adres. W miarę jak procesor przebiega swój program punkty te łączą się razem i tworzą obraz.

5370A jest tak zaprojektowany, że włączenie zasilania bez podania sygnału na wejście daje obraz, który pojawia się, gdy MP wykonuje wbudowaną procedurę diagnostyczną zainicjowaną włączeniem zasilania i sprawdzającą RAM, ROM i inne układy jądra. Obraz ten świadczy o poprawnym działaniu jądra, o sprawności szyn danych i adresów i o tym, że licznik oczekuje podania sygnału wejściowego, który rozpoczyna właściwą pracę przyrządu.

Istotne jest, że nie ma potrzeby sprawdzania każdego punktu obrazu, wystarczy stwierdzić poprawność jego ogólnego kształtu. Dla omawianego przyrządu istnieje np. 12 wzorcowych obrazów, które serwis powinien umieć rozpoznać. Podane w podręczniku serwisowym dokumentują one takie funkcje przyrządu, jak: trwający pomiar, brak sygnału stopu po sygnale startu, bieg jałowy, odczyt/zapis, zerowanie, wyświetlanie.

Jeśli na ekranie pojawia się jakiś inny wzór, oznacza to, że procesor wykonuje jakiś nieokreślony program lub, że "zapętlił się" w jakiejś części algorytmu. Jeśli istnieje podejrzenie, że przyczyną jest wymuszanie przez jakiś pakiet stałego, niskiego lub wysokiego poziomu którejś z linii szyny adresowej, próbować można wyjmować kolejno po jednym pakiecie, póki na ekranie nie pojawi się poprawny obraz. Jeśli to nie daje efektu, trzeba przejść do analizy sygnatur.

## IV. TESTOWANIE IN-CIRCUIT

### Wstęp

Opracowanie wartościowej procedury testowania w produkcji pakietu cyfrowego stało się zadaniem prawie równie złożonym, jak samo zaprojektowanie zawartej w nim sieci. Wraz ze wzrostem złożoności systemów elektronicznych stało się oczywiste, że tradycyjne "funkcjonalne" procedury testowe napotykały granice swych możliwości. Metoda testowania "kostka za kostką", znana pod nazwą testowania "w układ" /in-circuit/, badana jako wyjście alternatywne dowiodła wkrótce, że jest w stanie zapewnić na linii produkcyjnej wyszukiwanie uszkodzeń z wysoką efektywnością. Współczesne systemy testować mogą każdy element na pakiecie, od prostego opornika do złożonych mikroprocesorów w ciągu kilku



zaledwie sekund. Systemy te zawierają także oprogramowanie, które pozwala konstruktorowi testu zbudować i uruchomić program testu in-circuit w czasie wielokrotnie krótszym niż poprzednio.

#### Porównanie testowania funkcjonalnego i "in-circuit"

System testów funkcjonalnych sprawdza sieć zrealizowaną w sprzęcie jako całość, stosownie do funkcji, dla których spełnienia była ona zaprojektowana. Włącza się zasilanie pakietu i podaje sygnały wejściowe możliwie najbardziej podobne do tych, które występują w normalnym działaniu. Mierzy się wyjścia i porównuje wyniki z przewidzianymi w projekcie. Podczas funkcjonalnego testu nie mierzy się wartości żadnych parametrów w wewnętrznych węzłach sieci. Połączenia sprzętu testującego z siecią testowaną sprowadzają się na ogół do złącza pakietu. Dla różnych sieci stosuje się różne programy sterujące testami.

Testowanie in-circuit izoluje elektrycznie każdy element na pakiecie, sprawdza jego charakterystyki i ignoruje jego funkcje w całości sieci.

Wszystkie elementy pakietu testowane są kolejno. Połączenie między systemem testującym in-circuit a pakietem zapewnia podciśnieniowe, wielopinowe gniazdo ustalające - "łożo fakira", czy też "łożo z gwoździami", przy czym każdy typ pakietu ma swoje własne specjalne gniazdo.

Test danego elementu przygotowany jest wcześniej, dlatego program tego testu może być wywołany zawsze, ilekroć element taki się pojawi, niezależnie od tego, gdzie i jak jest on użyty na którymś z pakietów. Dysponując biblioteką testów elementów konstruktor testu może wygenerować cały program testu in-circuit na podstawie listy elementów i ich topologii - przez ułożenie w odpowiednią sekwencję uprzednio zapamiętanych segmentów testujących elementy. Gdy któryś z elementów nie przechodzi swego testu, drukuje się komunikat diagnostyczny podający położenie tego elementu i rodzaj uszkodzenia.

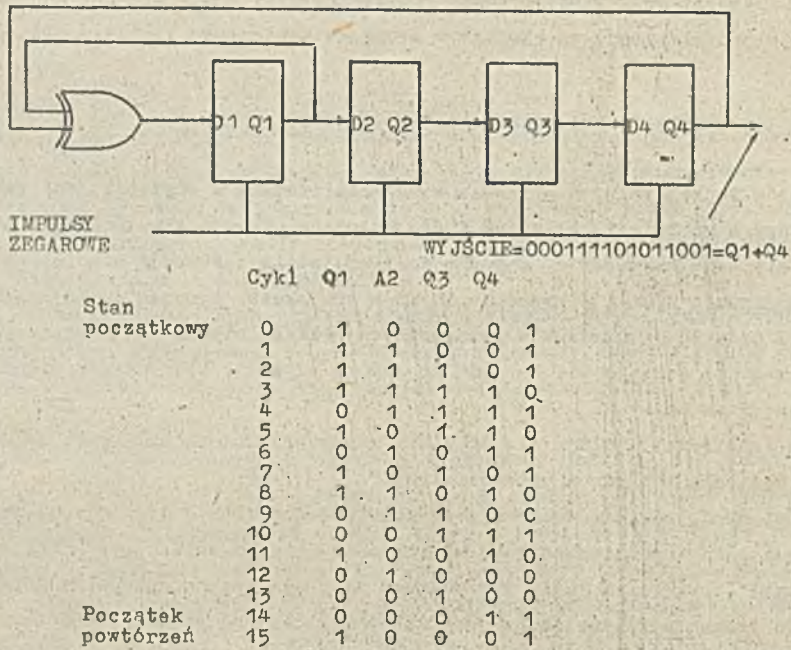
W zasadzie testowanie in-circuit odpowiada na pytanie, czy poprawnie zmontowano układ, podczas gdy testowanie funkcjonalne - na pytanie, czy układ działa. Tym niemniej wzgląd na ograniczenia możliwości testowania funkcjonalnego powoduje, że wielu "testowców" preferuje na linii produkcyjnej testowanie in-circuit jako wstępny krok do lub nawet zamiast testowania funkcjonalnego.

#### Ograniczenia testowania funkcjonalnego

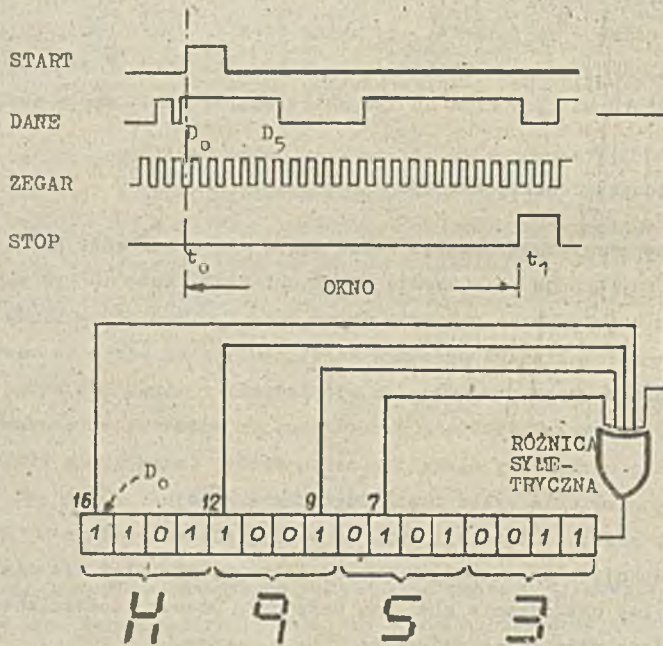
Testy funkcjonalne sprawdzają układy. Jeśli układ nie przechodzi przez test funkcjonalny, następnym krokiem jest lokalizacja uszkodzenia w tym układzie. Może to być zadanie bardzo poważne, ponieważ wiele różnych uszkodzeń na pakiecie dawać może ten sam objaw błędu w teście funkcjonalnym. W systemie funkcjonalnym lokalizacja prostego nawet zwarcia na pakiecie zawierającym setki elementów wymagać może skomplikowanego programowego modelowania i minut lub nawet godzin wspomaganego badania pakietu sondą przez wykwalifikowanego technika. Te kosztowne procedury diagnostyczne lokalizują na ogół jeden tylko uszkodzony element w danym czasie. Uszkodzenia wielu elementów wymagają często wielokrotnego powtarzania cyklu test/diagnoza/reperacja.

Czasami wyjście z granic tolerancji dwu lub więcej elementów daje efekt wzajemnego kasowania, który uniemożliwia detekcję w czasie testu funkcjonalnego, ale błąd ujawnia się później w normalnej eksploatacji. Zwarcie lub uszkodzenie elementu może przy włączeniu zasilania całego pakietu w celu wykonania testu funkcjonalnego zniszczyć także inne elementy.

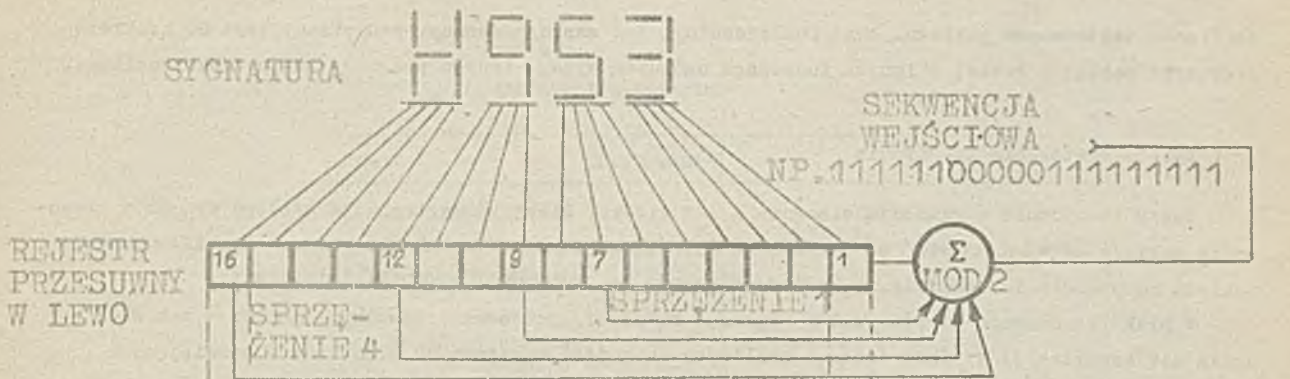
Na koniec - testy funkcjonalne są same przez się trudne w programowaniu, szczególnie w wypadku sieci cyfrowych. Ideałem byłoby sprawdzenie wszystkich możliwych stanów wejść i wyjść całego pakietu. Programowanie testu funkcjonalnego pakietu wymaga często szczegółowej wiedzy o wewnętrznym



Rys.1A. Zasady analizy sygnatur. W analizie sygnatur generator pseudolosowych sekwencji binarnych produkuje cykliczny redundancyjny kod kontrolny /CRC/, rodzaj sumy kontrolnej. Przykład dla 4 bitów - rejestr przesuwany ze sprzężeniem zwrotnym przebiega cyklicznie wszystkie 15 niezerowych stanów. W praktyce stosuje się rejestr 16-bitowy, a na wejście rejestru wchodzi ciąg bitów wzdłuż, którego sygnaturę wyznaczamy.



Rys.3B. Zasady analizy sygnatur. Analiza sygnatury. Ta 4-cyfrowa sygnatura heksadecymalna jest kompresją 20-bitowego strumienia danych podanego na 16-bitowy rejestr przesuwany z liniowym sprzężeniem zwrotnym. Zbiór znaków złożony z 0...9, A, C, F, H, P i U pozwala na użycie 7-segmentowego wyświetlacza



	ZAWARTOŚĆ REJESTRU PRZESUWNEGO	NASTĘPNE WEJŚCIE DO REJ. PRZES.	BIT NASIŁA SEKWENCJI WEJŚCIOWEJ
0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1	1
1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	1	1
2	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1	1	1
3	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1	1	1
4	0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1	1	1
5	0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1	1	1
6	0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1	0	0
7	0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1	1	0
8	0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 1	1	0
9	0 0 0 0 0 0 1 1 1 1 1 1 0 1 1 1	0	0
10	0 0 0 0 0 1 1 1 1 1 1 0 1 1 0 1	0	0
11	0 0 0 0 1 1 1 1 1 1 0 1 1 0 0 0	1	1
12	0 0 0 1 1 1 1 1 1 0 1 1 0 0 0 1	0	1
13	0 0 1 1 1 1 1 1 0 1 1 0 0 0 1 0	1	1
14	0 0 1 1 1 1 1 0 1 1 0 0 1 0 1 1	0	1
15	0 1 1 1 1 1 0 1 1 0 0 1 0 1 0 1	1	1
16	1 1 1 1 1 0 1 1 0 0 1 0 1 0 1 1	0	1
17	1 1 1 1 0 1 1 0 0 1 0 1 0 1 0 1	0	1
18	1 1 1 0 1 1 0 0 1 0 1 0 1 0 0 0	1	1
19	1 1 1 0 1 1 0 0 1 0 1 0 1 0 0 1	1	1
20	1 1 0 1 1 0 0 1 0 1 0 1 0 0 1 1	-	-

Rys.1C. Zasady analizy sygnatur

Konwersja danych z rys.1B. Praca rejestru w 20 taktach wyznaczonych zegarem między STARTEM i STOPEM. Stan końcowy jest sygnaturą

działaniu testowanego pakietu. Test funkcjonalny, już zaprogramowany, przypisany jest do konkretnego typu pakietu. Pakiet o innych funkcjach ogólnych wymaga innego programu testu funkcjonalnego.

#### Zalety testowania in-circuit

Testy in-circuit sprawdzają elementy a nie układy. Elektryczna izolacja każdego obwodu i trasowanie pakietu punkt /montażu/ po punkcie umożliwiają systemowi testów in-circuit szybko sprawdzić pakietu na zwarcia i rozwarcia, i każdego elementu na poprawność montażu i działania.

W praktyce zwarcia i rozwarcia korygowane są przed puszczeniem testów elementów - tak więc unika się komplikacji wyników testu i możliwych uszkodzeń kosztownych elementów w późniejszych fazach testu. 200-elementowy pakiet przetestować można w mniej niż 100 sekund - jest to wielkość typowa - przy czym ważne jest, że jednoczesne uszkodzenia wielu elementów nie zwalniają tego procesu. Ponieważ każdy segment testu związany jest z jednym tylko elementem, znika potrzeba wspomaganego próbkowania w celu lokalizacji defektu i "ręcznego" diagnozowania. Ponieważ odpowiednie komunikaty diagnostyczne drukowane są automatycznie, przy każdym uszkodzeniu elementu wykrytym testem in-circuit, operator po prostu dołącza do pakietu jego wydruk diagnostyczny, a ten służy jako instrukcja dla technika usuwającego uszkodzenia pakietów.

Oprogramowanie systemu testów może być tak zaprojektowane, aby sprawdzało i tabulowało uszkodzenia elementów. Jeśli dane ujawniają błąd, np. lutowania lub brak elementu w tym samym miejscu na na pewnej liczbie pakietów wnieść można poprawkę przed uruchomieniem pełnej produkcji. Automatyczna generacja danych o uszkodzeniu upraszcza poważnie sporządzanie raportów o niezawodności elementów i stratach produkcyjnych.

Programowanie testu in-circuit jest stosunkowo proste, ponieważ każdy segment programu dotyczy jednego tylko elementu. Testowanie in-circuit ma wysoką efektywność, każdy bowiem element sprawdza się co do jego charakterystyk abstrahując od jego funkcji w sieci. Użytkownik zbudować może bibliotekę segmentów testowych elementów i stosować je dla tych samych elementów na dowolnym pakiecie.

Prawdą jest, że testowanie in-circuit wymaga wyprodukowania specjalnego gniazda ostrzowego dla każdego typu pakietu, aby zapewnić interfejs między nim a systemem testującym - koszt gniazda /2500 dolarów dla pakietu o 600 węzłach/ zwraca się jednak prawie zawsze z nadwyżką ponieważ niższe są koszty programowania i obsługi i wyższa niż przy funkcjonalnym testowaniu przepustowość. Konstruktor testu funkcjonalnego traci na programowanie do dwóch miesięcy, a test in-circuit takiego samego pakietu może być napisany w ciągu jednego - dwóch tygodni przez technika.

Przyjmuje się na ogół, że obecne testery in-circuit zlokalizować mogą 90% możliwych uszkodzeń pakietu. Jest to często stopień detekcji dostatecznie wysoki, aby w ogóle zrezygnować z testowania funkcjonalnego, ponieważ może bardziej opłacać się ewentualne wykrywanie pozostałych uszkodzeń pakietu po prostu w pracującym systemie. Niekiedy jednak konstruktor testu żąda wyższego poziomu ufności, czasem nie jest też skłonny przyjąć, że poprawna praca elementów równoznaczna jest z poprawną pracą całego układu. Wówczas stosuje się jeszcze testowanie funkcjonalne. Poprawnie zaprojektowany test funkcjonalny jest w stanie podnieść poziom detekcji uszkodzeń do 98%, jednak tylko pod warunkiem uprzedniego wykonania testu in-circuit. Wady i zalety obu sposobów testowania zebrano w tabeli.

#### Testowanie cyfrowe in-circuit

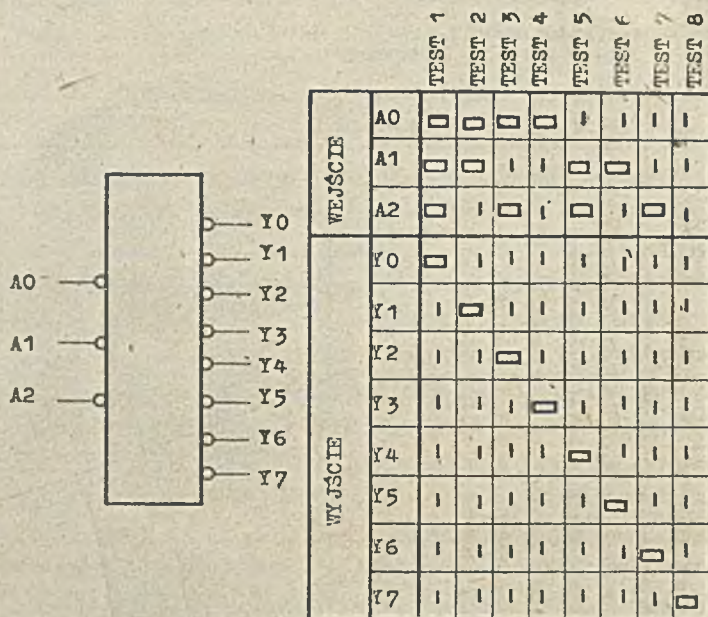
W ciągu ostatnich kilku lat postęp technologii układów scalonych zwiększył raptownie procent skomplikowanych urządzeń cyfrowych montowanych na pakietach, ostro stawiając problem ich przetestowania. Pierwsze próby zastosowania testowania in-circuit do elementów cyfrowych podjęto około 1974 roku.

Tab.1. Sposoby testowania

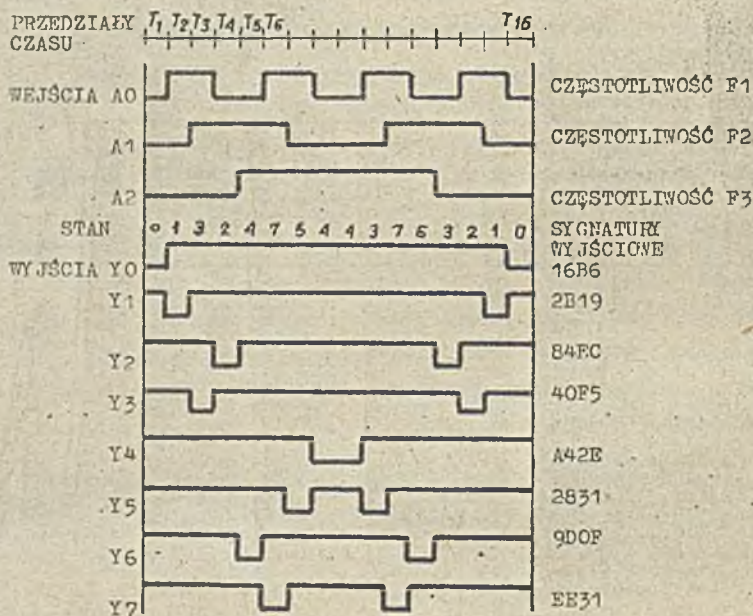
Testowanie	
<u>funkcjonalne</u>	<u>in-circuit</u>
Sprawdza działanie zmontowanego układu traktowanego jako całość	Sprawdza poszczególne elementy
Lokalizacja uszkodzeń elementów wymaga złożonej symulacji programowej i/lub ręcznego sondowania. Kosztowna i czasochłonna	Łatwo lokalizuje wszystkie uszkodzenia elementów
Uszkodzenia wielu elementów wymagają iteracji procesu diagnozy	Uszkodzenia wielu elementów diagnozowane są w jednym przebiegu testu
Uszkodzone elementy mogą pozostać niesprawdzone	Wszystkie elementy są sprawdzane
Włączenie zasilanie uszkodzonego pakietu może zniszczyć elementy	Niedestruktywne. Zwarcia i rozwarowania są lokalizowane i naprawiane we wstępnych etapach procedury testowej. Elementy dyskretne są sprawdzane bez włączania zasilania całego pakietu
Skomplikowane programowanie. Programowanie wymaga szczegółowej wiedzy o sieci zawartej na pakiecie	Proste programowanie. Programowanie wymaga wiedzy tylko o elementach i ich rozmieszczeniu
Możliwość użycia tej samej łączówki dla testowania różnych typów pakietów	Unikalne gniazdo ostrzowe dla każdego typu pakietu
Różne programy testowe dla każdej sieci	Różne segmenty testowe dla każdego elementu. Programy różnych sieci generowane łatwo na podstawie biblioteki segmentów testowych elementów

#### Pierwsze rozwiązania

Początkowo testowano elementy cyfrowe w układzie /in-circuit/ wymuszając na wejściach urządzenia znane wartości logiczne i porównując odczytane stany wyjść z oczekiwanymi wartościami tabeli prawdy /rys.2/. Pobudzenia były krótkotrwałe /na ogół mniej niż 1 ms/ by zminimalizować efekt ciepłyny i zmniejszyć możliwość uszkodzenia elementu; miały niską impedancję /około  $5\Omega$  /, by "przebić" ewentualne sygnały pojawiające się na wyjściach innych elementów sieci podłączonych do tego samego wężła. Ponieważ "testowanie tablicy prawdy" wymagało przygotowania różnych zbiorów instrukcji programowych dla sprawdzenia każdego stanu, programowanie okazało się czasochłonne i kosztowne, szczególnie dla złożonych elementów cyfrowych. Aby utrzymać oprogramowanie w granicach rozsądku konstruktor testu szedł na kompromis kosztem pokrycia, polegając na statystycznie sprawdzonym poziomie ufności w sprawność elementu. Co więcej, ta metoda "jednego stanu na raz" okazała się kłopotliwa przy testowaniu pamięci i innych złożonych sieci sekwencyjnych. Stało się wkrótce jasne, że użyteczność testowania in-circuit stanęła pod znakiem zapytania.



Rys.2. Klasyczne podejście - test dekodera 3 na 8. Na wejścia podaje się wartości tabeli prawdy i mierzy się odpowiednie wyjścia. Sprawdzenie wszystkich możliwych stanów wymaga ośmiu osobnych testów. Program generacji różnych stanów wejściowych jest czasochłonny, a wynikowe programy testów nieoptymalne, szczególnie dla złożonych elementów zawierających logikę sekwencyjną



Rys.3. Analiza sygnatur. Stosuje się spójny strumień zmiennych w czasie wejść: wejście A1 o połowie częstotliwości wejścia A0, a wejście A2 o połowie częstotliwości wejścia A1. W ciągu jednego pełnego cyklu wejścia A2 przebiega się wszystkie możliwe stany wejść. Każde wyjście próbkowane jest przez ten sam okres czasu i podawane na rejestr generujący sygnatury, który produkuje unikalne 4-cyfrowe heksadecymalne sygnatury dla każdego wyjścia. Są one porównywane z sygnaturami otrzymanymi z odpowiednich pinów wzorcowego dobrego urządzenia. Jeden test bada wszystkie możliwe stany

## Analiza sygnatur

Problem rozwiązano przez wprowadzenie analizy sygnatur cyfrowych, metody stosowanej od dawna dla analizy błędów pamięci komputerowych, a ostatnio także dla potrzeb serwisu systemów cyfrowych. Rys.3 ilustruje zastosowanie tej metody dla urządzenia pokazanego na rys.1. Te same stany logiczne są generowane na wejściach i te same wartości z tablicy prawdy oczekiwane na wyjściach. Tu jednak wejścia są zbiorem harmonicznym sygnałów zwanych "spójnymi, normowanymi w czasie ciągami bitów". Wszystkie one są prostokątnymi przebiegami zmiennymi od logicznego stanu 0 do 1. Wejście A1 jest przebiegiem o częstotliwości pół AO, A2 ma częstotliwość pół A1. Zegar główny zapewnia właściwe relacje fazowe między wejściami i tworzy strumień pobudzeń, który przebiega wszystkie stany kombinatoryczne pod warunkiem, że pobudzenia wejść trwają przez co najmniej jeden okres wejścia o najniższej częstotliwości.

Każde wyjście próbkowane przez dokładnie ten sam okres czasu, podawane jest na rejestr generujący sygnatury, który produkuje 4-cyfrową hexadecymalną liczbę, zwaną sygnaturą. Ta sygnatura porównywana jest z wzorcową sygnaturą, zdjętą z identycznego urządzenia, o którym wiemy, że nie zawiera uszkodzeń, a zapamiętaną w pamięci. Jeśli obie 4-cyfrowe liczby są równe, element przechodzi inspekcję pozytywnie. Jeśli nie, drukuje się komunikat diagnostyczny identyfikujący element jako uszkodzony. Jeden prosty zbiór instrukcji wystarcza do zaprogramowania całego testu.

Metoda analizy sygnatur eliminuje konieczność analizowania stanów wyjść urządzenia. Konieczne jest tylko porównanie sygnatury urządzenia z oczekiwaną sygnaturą zapamiętaną w pamięci - ta oczekiwana sygnatura określana jest empirycznie. Tak więc, np. programista testu nie musi znać programu zawartego w programowalnej pamięci stałych /PROM/, którą testuje. Magazynuje po prostu w pamięci sygnaturę poprawnie działającego PROM-u i używa jej jako standardu testując inne PROM-y tego samego typu, zaprogramowane w ten sam sposób. Jeśli później nastąpi korekta zawartości PROM-u łatwo jest określić nową sygnaturę.

Oferowany obecnie sprzęt testowania in-circuit potrafi podawać serie równoległych, spójnych strumieni wejściowych związanych podaną formułą "potęgi dwójki":  $F1=2F2$ ,  $F2=2F3$ , itd. Przy typowej wielkości  $F1=500$  kHz urządzenie z przykładu na rys.2 może być przetestowane o ciągu zaledwie 8 mikrosekund. Testery in-circuit zapewniają też możliwość podawania pobudzeń inicjujących urządzenie, jak również podawania stałych logicznych 0 i 1 na wybrane wejścia, co może być konieczne przy testowaniu pewnych funkcji.

Programista testu in-circuit nie jest całkowicie zwolniony z obowiązku znajomości stanów wewnętrznych złożonego elementu. Dla urządzeń zawierających logikę sekwencyjną, takich jak mikroprocesory i pamięci musi brać pod uwagę relacje czasowe, ponieważ te same końcówki mogą być używane, zarówno jako wejście, jak i wyjście lub jako adres albo dane. Tym niemniej łatwość, z jaką segmenty programowe mogą być napisane, a raz napisane - używane wielokrotnie dla sprawdzania taki samych elementów na pakietach różnych typów, sprawia, że testowanie in-circuit jest wartościowym rozwiązaniem nawet dla najbardziej złożonych urządzeń wielkiej skali integracji /LSI/.

## Programowanie cyfrowego testu in-circuit

### Dokumentacja gniazda testowego

Gniazdo testowe jest interfejsem między systemem testującym in-circuit, a testowaną siecią na pakiecie /rys.4/.

Pierwszą fazą programowania testu in-circuit jest dokumentacja gniazda testowego, tzn. identyfikacja węzłów i generacja listy elementów.

Podstawa gniazda łączy się z systemem testującym przez standardowy wtyk. Doprowadzenia wewnętrzne gniazda testowego łączą węzły systemu testującego z końcówkami gniazda, które z kolei używane są jako styki węzłów testowanego pakietu. "Identyfikacja węzłów" oznacza przyporządkowanie węzłów pakietu ponumerowanym węzłom systemu testującego. Zamiast deterministycznie prowadzić połączenia np. 600 do 1000 końcówek z określonymi końcówkami gniazda podczas jego montażu można połączyć końcówki gniazda ze złączem w sposób przypadkowy /rys.5/. Operator za pomocą oprogramowania systemu testującego identyfikuje następnie sondą węzły i wprowadza listę elementów - oszczędzając pracę i czas.

Najpierw, nie używając jeszcze pakietu, kładzie się na wszystkie końcówki płytę przewodzącą, która łączy je wzajemnie. Prosta komenda z pulpitu pomaga systemowi programowemu określić końcówki nie podłączone. Informacja ta może być później wyświetlana, aby pomóc w ujawnieniu błędów montażu gniazda.

Następnie programowane są ścieżki. Modelowy pakiet, niekoniecznie wolny od uszkodzeń, umieszczony jest w gnieździe i włączane jest podciśnienie wymuszające połączenie końcówek z węzłami pakietu. Na komendę z pulpitu system testowy identyfikuje i zapamiętuje wszystkie ścieżki, tj. określa które ponumerowane węzły testera zwarte są ze sobą przez pakiet. Na tym etapie wszystkie ścieżki traktuje się jako zamierzone. Później, w fazie generacji programu, wszystkie ścieżki, które w istocie były niepożądanymi zwarciami na modelowym pakiecie mogą być wykryte i wyeliminowane.

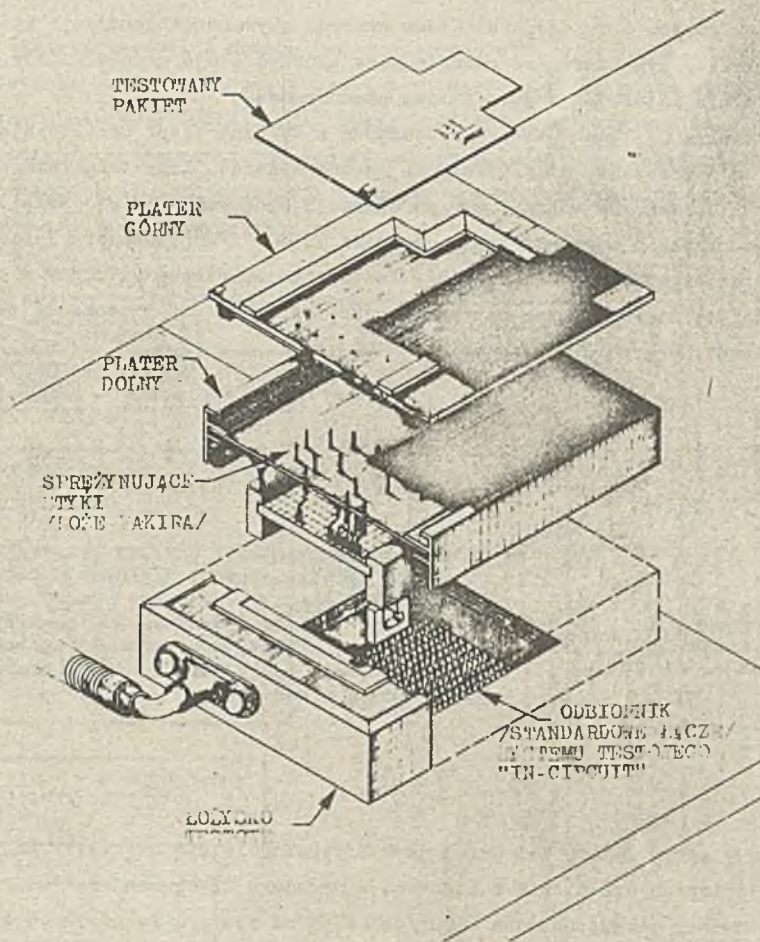
Po zaprogramowaniu ścieżek sondowane są kolejno elementy i generowana kompletna lista elementów. Operator pisze np. "R13" identyfikując konkretny opornik, a system testowy odpowiada instrukcją przyłożenia sondy najpierw do jednego, później do drugiego końca opornika R13 wprowadzając do pamięci numery węzłów końcówek opornika R13. Przy bardziej złożonych elementach, takich jak kostki scalone, operator wskazuje typ elementu i numer nóżki, a system daje instrukcje dotyczące kolejności próbkowania sondą nóżek.

Gniazdo testowe ma pełną dokumentację, gdy zidentyfikowano wszystkie węzły i elementy i wprowadzono listę elementów. Można wtedy zażądać od systemu pokazania "listy wykluczeń" - wszystkich nieużywanych węzłów systemu testowego i podania w kolejności węzłów list nazw elementów podłączonych do każdego węzła pakietu. Lista wykluczeń ułatwia uruchomienie sprzętu gniazda, a listy elementów węzłów pozwalają operatorowi na wpisanie numerów węzłów na schemacie pakietu, co ułatwia ich identyfikację w następnej fazie układania programu.

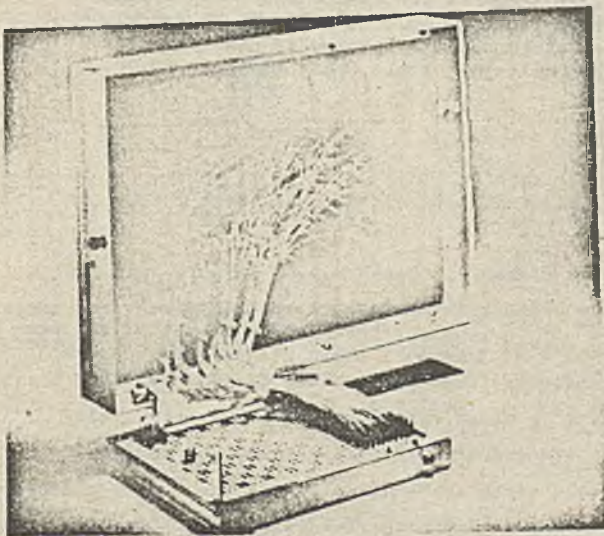
#### Generacja programu

Informacja zebrana podczas dokumentowania gniazda wykorzystywana jest teraz do generacji listy elementów i ich położenia, lista służy jako dane dla generatora programu będącego częścią oprogramowania systemu.





Rys.4. Obudowa urządzenia testującego - zespół rozmontowany. Odbiornnik lub inny standardowy konektor łączy obudowę urządzenia testującego z licznymi punktami węzłowymi systemu testującego. Szpilki obudowy stykają się z węzłami płytki układu w momencie podłączenia podciśnienia do systemu



Rys.5. Urządzenie testowe przedstawiające uzwojenie płyty rozdzielczej. Połączenia między sondą i szpilkami płyty rozdzielczej /wtykowej/ mogą być wykonywane dowolnie. Oprogramowanie systemowe otwiera węzły płyty układowej, gdy operator dotyka je sonda

Dla urządzeń cyfrowych istnieje biblioteka wzorców używana do generacji spójnych pobudeń wejściowych dla analizy sygatur. Programista może tworzyć swoje własne wzorce, aby testować urządzenia, dla których nie istnieją jeszcze wzorce standardowe.

Każda programowalna maszyna musi być wyposażona w wygodny język umożliwiający użytkownikowi efektywne sterowanie operacjami, które jest ona zdolna wykonać. Komputery-wymagają języków dostosowanych do obliczania, maszyny testujące - do operacji stosowanych w procesie testowania.

Rys.6 pokazuje przykład segmentu testu opornika, diody i kondensatora zapisany w pewnym konkretnym języku charakterystycznym dla systemów testowania in-circuit. Teksty w cudzysłowach drukowane są jako komunikaty diagnostyczno-naprawcze, jeśli element nie przechodzi testu. Następny wiersz podaje wartości parametrów testowania.

```
RES
"R 13"
470K, 375-922/12
```

```
DIODES
"CR 17"
542A-286C HRL
```

```
CAP
"C 26"
750P,85-23/289
```

Rys.6. Typowe segmenty użytkownika testujące elementy pakietu. Nagłówek w cudzysłowie pojawi się jako komunikat diagnostyczny, jeśli element nie przejdzie testu. Ostatni wiersz segmentu określa test, np. test dla opornika 470 kΩ, którego węzły pobudzenia, pomiaru i izolacji mają numery 375, 922 i 12.

Segment testowy sprawdzający dekodery 3 na 8 z rys.2 pokazano na rys.7. Zauważmy, że cały segment tego 8-wyjściowego urządzenia zakodowano w zaledwie 12 wierszach. Jeszcze wyraźniejsza jest prostota kodu testowania mikroprocesora 6100 /pokazana na rys.8/. Zaledwie 15 wierszy wystarcza dla przetestowania całego sterowania tego urządzenia.

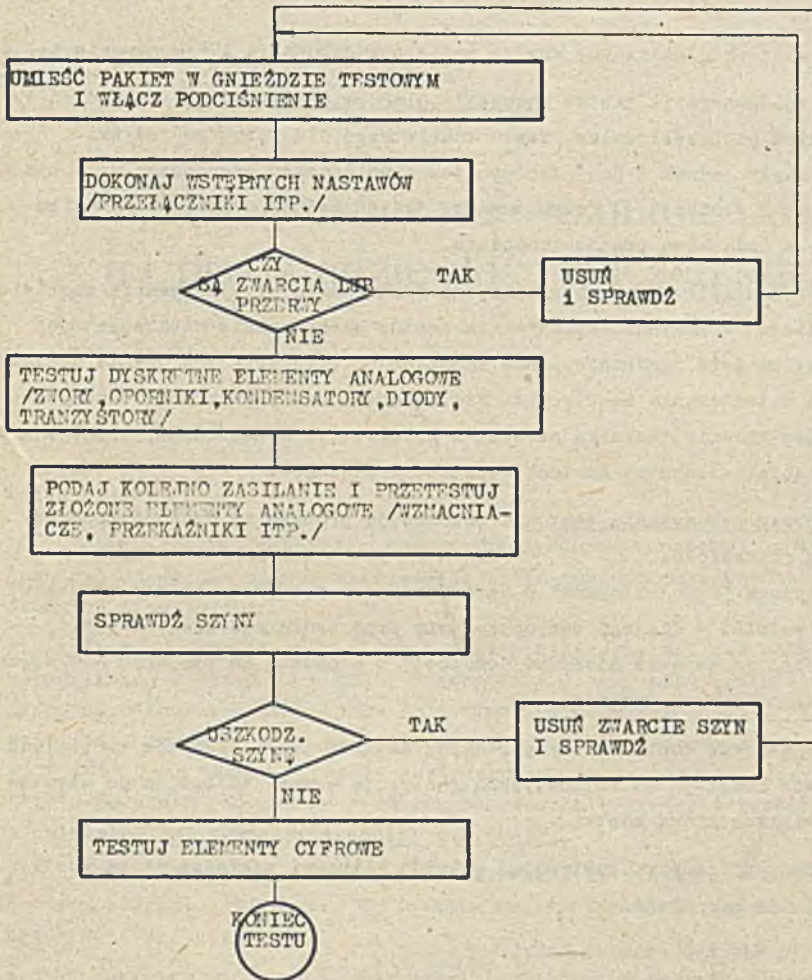
```
"T5 74LS138"
41,F1 'AO
42,F2 'A1
104,F3 'A2
CRC 16B6 M43 'Y0
CRC 2B19 M77 'Y1
CRC 84E2 M52 'Y2
CRC 40F5 M117 'Y3
CRC A42E M137 'Y4
CRC 2B31 M48 'Y5
CRC 9D0F M56 'YE
CRC EE31 M107 'Y7
```

Rys.7. Testowanie dekodera 3 na 8 z rys.1. Określa się tylko węzły podania częstotliwości wejściowych /41, 42 i 104/ i oczekiwane sygatury węzłów wyjściowych /np. 16 B6 dla Y0 mierzonego w węźle 43/. Założono dla prostoty, że dekodery w tym przykładzie jest stale w stanie "enable".

```
"IC E124 61500"
'TEST MICROPROCESSOR TIMING OUTPUTS
530,F1 'OSC
475,F6 'RUN/HLT
703,F9 'RESET
537,F12 'WAIT
658,F3 'SCOPE REF E175 PIN 5
CLOCK B '2 US/STATE
CRC BB49 M472 FROM F12 TO F14 'XTA
CRC 5618 M552 'XTB
CRC 531E M541 'XTC
CRC 7180 M344 'RUN
CRC 5E7A M543 'LXNAR
CRC 70D5 M464 'IFETCH
CRC 66EB M465 'MEMSEL
```

Rys.8. Test złożonego urządzenia LSI. Cały program sprawdzenia logiki zegarowej i sterującej mikroprocesora 6100 mieści się łatwo na ekranie monitora

Oprogramowanie systemu testów, z pomocą programującego, buduje według algorytmu pokazanego na rys.9 procedurę testowania całego pakietu. Zwarcia i rozwarcia, najczęstsze uszkodzenia pakietów wykrywane są i usuwane w pierwszej kolejności. Każdy błąd tej fazy oznaczać może komplikację późniejszego testu lub zniszczenie elementu po indywidualnym podaniu zasilania. Uruchamianie - zwykle najbardziej czasochłonny etap generacji programu - można znacznie uprościć stosując programowanie interaktywne i procesor interpretacyjny.



Rys.9. Algorytm testowania in-circuit. Najpierw sprawdza się uszkodzenia najpopularniejsze. Jeżeli wykryje się zwarcia lub rozwarcia szyn - przerywa się test

Procesor interpretacyjny ułatwia uruchamianie programu

Ponieważ program testu in-circuit składa się z wielu osobnych segmentów testów elementów, procesor interpretacyjny jest idealnym narzędziem uruchamiania. Pojedynczy segment, np. testowania mikroprocesora może być wielokrotnie modyfikowany przez programistę i wykonywany przez procesor interpretacyjny bez konieczności każdorazowej kompilacji.

Interpreter taki przydatny jest także do wykonywania testów na linii produkcyjnej, choć wykonanie programu interpretowanego trwa dłużej niż wykonanie tego samego programu po kompilacji. Przeciwnie niż w klasycznym przetwarzaniu, gdzie przepustowość zależy od szybkości jednostki centralnej, przepustowość testowania in-circuit zależy bardziej od czynności związanych ze sprawdzanym sprzętem,

takich jak ustawienie przełączników, oczekiwanie na rozładowanie pojemności, ręczna wymiana pakietów i włączanie podciśnienia gniazda testującego dla każdego testowanego pakietu. Główna zaleta kompilacji - szybkość wykonywania programu, nie ma więc znaczenia przy testowaniu in-circuit - istotną natomiast okazała się wygoda uruchamiania będąca zaletą interpretera.

#### V. WNIOSKI

Rosnące scalenie układów spowodowało gwałtowny kryzys metod przygotowania i wykonywania testów.

Kryzys metod przygotowywania, tj. generacji testów pozostaje nierozwiązany. Modele uszkodzeń są nieznanne, a modelowanie uszkodzeń nieprzeliczalne. Testu kompletnego nie potrafimy uzyskać. Rosnąca niezawodność elementów pozwala jednak z dostatecznym poziomem ufności poprzestać na testach niekompletnych, mających charakter próbkujący. Przygotowanie takich testów staje się zadaniem konstruktora, obciążając projekt dodatkową pracochłonnością.

W technikach wykonywania testów nastąpił zwrot polegający na odchodzeniu od wykonywania możliwie kompletnych testów "złącza pakietu" w kierunku wykonywania testów zdecydowanie niekompletnych "z nówek elementu". Obserwujemy tu dwie tendencje, obie zapożyczone z technik testowania urządzeń analogowych - analizę sygnatur i testowania in-circuit. Nie są one rozdzielne ani alternatywne, jakkolwiek analiza sygnatur jest głównie techniką serwisu i lokalizacji uszkodzenia, podczas gdy testowanie in-circuit jest wyłącznie techniką kontroli produkcji pakietów.

Obie te techniki przełamują kryzys wykonywania testów i rozwiązują problem na dzień dzisiejszy i prawdopodobnie na najbliższą przyszłość.

Analiza sygnatur powinna być jednak "wprojektowana" w urządzenie, co podwyższa koszty projektu, obniżając jednocześnie koszty serwisu - dlatego też opłaca się przy produkcji długich serii i odpowiedzialnym serwisie. Można ją wszakże stosować "ex post" w stosunku do projektu - w ograniczonym zakresie i nie dla jednostkowych urządzeń.

Testowanie in-circuit opłaca się także dopiero przy produkcji długich serii pakietów ze względu na koszt gniazda. Daje ono pewne nadzieje na automatyczną generację - mamy tu bowiem do czynienia co najwyżej z pojedynczą wielkoscaloną kostką.

Mniejsi producenci stosują metody pośrednie, zastępując gniazdo klipsami zakładanymi na kostki lub w podstawki, z których wyjmuje się elementy.

W opinii autora należy w kraju wdrożyć obie techniki:

- analizę sygnatur i
- testowanie in-circuit.

Jeśli tego nie zrobimy, za kilka lat nie będziemy w stanie testować urządzeń cyfrowych.

#### ⊙ Wdrożenie analizy sygnatur oznacza:

- produkcję czytnika sygnatur,
- wprowadzenie projektowania z analizą sygnatur,
- dokumentację sygnatur produkowanych urządzeń włącznie z trawieniem wzorcowych sygnatur na pakietach.

#### ⊙ Wdrożenie testowania in-circuit oznacza:

- wybór lub produkcję odpowiedniego testera,
- produkcję gniazd testowych in-circuit /podciśnieniowych z wymiennymi matrycami ostrzowych styków/,
- produkcję matryc ostrzowych styków,
- oprogramowanie testera in-circuit /interpretacyjny, konwersacyjny system testowania i biblioteki testów elementów/.

mgr inż. Jerzy MOCALA

mgr inż. Ignacy STREMBICKI

Instytut Maszyn Matematycznych

## Problemy nauczania użytkowania komputerów na przykładzie doświadczeń szkolnictwa ZSRR

### Wstęp

Rosnąca rola techniki cyfrowej w gospodarce narodowej, a także w życiu codziennym, wymusza nowe spojrzenie na zakres wiadomości, jakie powinni posiadać absolwenci szkół podstawowych, średnich i wyższych.

Nie wystarczy nasycić gospodarki sprzętem cyfrowym, ani też zwiększyć liczby absolwentów - specjalistów w dziedzinie EMC i programowania, aby wykorzystać potencjalne możliwości techniki cyfrowej. Warunkiem koniecznym jest wyrobienie praktycznych nawyków użytkowania techniki cyfrowej. Zatem poznanie możliwości techniki cyfrowej w sferze swojej profesji zawodowej powinno mieć taką samą wagę w procesie kształcenia, jak matematyka, fizyka, gramatyka itd. Dlatego potrzeba poważnie podejść do problemu rozszerzenia wiedzy w procesie kształcenia społeczeństwa o dziedzinę "użytkowania techniki cyfrowej".

Upowszechnienie sprawnego i efektywnego korzystania z komputera wymaga nowego jakościowego systemu przygotowania kadr. Zbudowanie takiego systemu równoznaczne jest ze znalezieniem odpowiedzi na pytania:

- jaki powinien być zakres przekazywania wiedzy na poszczególnych poziomach kształcenia?
- jak zagwarantować środki zapewniające dostępność techniki cyfrowej w procesie kształcenia?
- jak efektywnie wykorzystywać komputery w procesie kształcenia?
- w jaki sposób przygotować kadrę pedagogów do tych zadań?
- w jaki sposób wprowadzić "nowy system nauczania" w życie przy istniejących ograniczeniach: środków (aspekt ekonomiczny i techniczny), kadry wykładowców, braku tradycji wykorzystania techniki cyfrowej w takich dziedzinach, jak medycyna, rolnictwo itp?

Wagę tych problemów w ZSRR zrozumiano już stosunkowo dawno i podjęto wiele przedsięwzięć zmierzających do zintensyfikowania przygotowania studentów do stosowania techniki cyfrowej. Zdaniem autorów artykułu doświadczenia te warte są upowszechnienia w sytuacji i gdy obecnie u nas w kraju opracowuje się program "komputerowego wspomagania nauczania" oraz analizuje się program elektryfikacji gospodarki.

Zakres umiejętności stosowania techniki cyfrowej

Zintensyfikowanie przygotowania studentów do efektywnego korzystania z komputerów oraz przekonanie, że wiedza o technice obliczeniowej jest wiedzą podstawową we wszystkich kierunkach działalności zawodowej - wymaga powszechnego wprowadzenia do programów nauczania przedmiotu "podstawy informatyki", aczkolwiek zakres i poziom wiedzy przekazywanej w ramach tego przedmiotu jest znacznie zróżnicowany w ZSRR. Ministerstwo wyższego i średniego kształcenia zawodowego zleciło Naukowo-Badawczemu Instytutowi Problemów Szkoły Wyższej opracowanie takich zróżnicowanych programów nauczania podstaw informatyki.

Instytut opracował ramowy w sensie zakresu i objętości program przygotowań absolwentów szkół wyższych w dziedzinie stosowania techniki cyfrowej, czy nauczania podstaw informatyki. Zgodnie z tym programem, nauczanie korzystania z komputerów dzieli się na bazowe i specjalistyczne.

Bazowe obejmuje trzy poziomy, specjalistyczne - dwa. Dla każdego poziomu określono kierunki czyli dziedziny naukowe, których wiedzę należy przekazać studentom (tab. 1,2) i liczbę godzin zajęć (tab. 3). Doświadczenie wskazuje, że przygotowanie absolwenta władającego techniką cyfrową na poziomie III wymaga rozwiązania na EMC 50-60 zadań. Dla II i I poziomu 25-30 i 10-15 zadań.

Dla każdej specjalności określono wymagany poziom przygotowania studentów w dziedzinie wykorzystania techniki cyfrowej (tab. 4). Na podstawie tych wytycznych każda wyższa uczelnia przygotowuje swój szczegółowy program nauczania kierując się m.in. takimi ogólnymi zasadami:

- zagwarantowanie ciągłości nauczania w wykorzystaniu techniki cyfrowej przez cały czas studiów np. w planie kształcenia specjalistów z grupy "automatyzacja i kompleksowa mechanizacja procesów chemiczno-technologicznych" przewiduje się nauczanie wykorzystywania techniki cyfrowej w ciągu 7 semestrów (z ogólnej liczby 9 - nauczania tej specjalności);
- wykluczenie dublowania informacji przekazywanej na poszczególnych poziomach;
- poprzedzenie przygotowania specjalistycznego przygotowaniem bazowym wg tab. 5;

Tab. 1

Zakres wiadomości wymagany w przygotowaniu bazowym

I poziom	Znajomość możliwości stosowania techniki cyfrowej w swojej i pokrewnych specjalnościach Zdobycie elementarnych nawyków rozwiązywania zadań za pomocą EMC Umiejętność napisania programu w jednym z algorytmicznych języków i przygotowania go do wykonania na EMC.
II poziom	Opanowanie metod układania algorytmów rozwiązywania zadań w swojej specjalności Opanowanie metod modelowania procesów i obiektów Znajomość i stosowanie języków algorytmicznych oraz języków sterowania zadaniami (poziom SO).
III poziom	Opanowanie metod programowania systemowego i optymalizacji w swojej specjalności Znajomość metod organizacji danych w EMC Umiejętność tworzenia pakietów programów swojej specjalności i włączania ich do biblioteki programów Umiejętność tworzenia pakietów programów i włączania ich do biblioteki programów EMC.

Tabl. 2

Zakres wiadomości wymagany w celu uzyskania specjalistycznego

	I poziom	II poziom
Technika mikroprocesorowa	Znajomość: możliwości stosowania techniki mikroprocesorowej do automatyzacji maszyn, architektury mikroprocesorów, programowania systemów mikroprocesorowych. Umiejętność wg zadanego algorytmu przygotowania i uruchomienia programu systemu mikroprocesorowego przeznaczonego do sterowania obiektem.	Znajomość możliwości współczesnej bazy techniki mikroprocesorowej, podstaw projektowania systemów mikroprocesorowych i programowania. Umiejętność przygotowania efektywnych algorytmów sterowania procesami, maszynami, agregatami w swojej specjalności.
Komputerowe Wspomaganie Projektowania (KWP)	Przygotowanie do korzystania z KWP, znajomość specyfiki środków technicznych i programowania, podstawowych pakietów swojej specjalności i umiejętności posługiwania się nimi w swojej pracy.	Znajomość zasad konstrukcji, funkcjonalnych możliwości i organizacji informacji KWP. Znajomość pakietów swojej specjalności i umiejętność posługiwania się nimi oraz organizowania wymiany informacji między oddzielnymi urządzeniami w procesie projektowania.
Systemy sterowania procesami technologicznymi (SSPT)	Znajomość podstaw techniki cyfrowej, systemów sterowania programowego, systemów automatyzacji procesów technologicznych i tworzenia na ich podstawie sterowania elastycznego. Znajomość algorytmów sterowania typowymi procesami technologicznymi. Umiejętność stosowania technicznych i programowych środków w swojej specjalności.	Znajomość podstaw projektowania systemów automatyzacji procesów technologicznych w swojej specjalności. Umiejętność stosowania środków technicznych. Opanowanie współczesnych metod modułowego projektowania specjalistycznych systemów cyfrowych kontroli i sterowania pracujących w czasie rzeczywistym. Umiejętność wyboru efektywnych algorytmów sterowania i zrealizowania ich.
Systemy laboratoryjne (SL)	Znajomość podstaw automatyzacji badań naukowych, planowania eksperymentu, sterowania nim i przetwarzania jego wyników. Umiejętność stosowania technicznych i programowych środków w swojej specjalności.	Znajomość teoretycznych metod modułowego projektowania specjalistycznych systemów cyfrowych do kontroli i sterowania w czasie rzeczywistym, typowych algorytmów, podstaw sprzężenia systemu z obiektem sterowanym, umiejętność wyboru efektywnych algorytmów sterowania przebiegiem eksperymentu.

Tabl. 3

Godziny przeznaczono na przygotowanie studentów do korzystania z techniki komputerowej

Lp.	Poziom	Przygotowanie				
		bazowe	technika mikroprocesorowa	KWP	SSPT - system sterowania procesami technologicznymi	SL - systemy laboratoryjne
1	I	60-120	40-70	60	60	60
2	II	150-250	80-140	90-120	90-150	90-120

Tab. 4

Zalecane poziomy przygotowania studentów do korzystania z techniki cyfrowej

Lp.	Specjalizacja	Przygotowanie				
		bazowe	technika mikro-procesorowa	KWP	systemy sterowania procesami technologicznymi	systemy laboratoryjne
1.	Geologia i poszukiwanie kopalin użytkowych	I,II	I	-	-	-
2.	Wydobycie kopalin użytkowych	I	I	-	-	-
3.	Energetyka	I, II	I, II	I, II	I	-
4.	Metalurgia	I,II	I	I	I	-
5.	Budowa maszyn i przyrządów	I, II III	I, II	I,II	I, II	I,II
6.	Technika elektronowa, automatyka, przyrządy elektronizne	II,III	I, II	I, II	I, II	I, II
7.	Radiotechnika i łączność	II,III	I, II	I, II	I, II	I, II
8.	Technologia chemiczna	I,II	I	I, II	I, II	-
9.	Technologia drewna i papieru, inżynieria lasu	I,II	I	I	I, II	-
10.	Technologia przetwórstwa środków spożywczych	I,II	I	I	I, II	-
11.	Technologia produkcji towarów masowych	I,II	I,II	I	I, II	-
12.	Budownictwo	I,II	I	I	I	-
13.	Geodezja i kartografia	I,II	I	I,II	-	-
14.	Hydrologia i meteorologia	I,II	I	-	I	-
15.	Gospodarstwo rolne i leśne	I	I			
16.	Transport	I,II	I,II	I,II	I,II	-
17.	Ekonomia	I,II, III	-	I	-	-
18.	Práwo	I	-	-	-	-
19.	Ochrona zdrowia i kultura fizyczna	I	-	-	-	-
20.	Uniwersytet	I,II,III	-	-	-	I,II
21.	Instytut pedagogiczne i wyższe szkoły kultury	I				



Tabl. 5

Wymagane poziomy przygotowania bazowego poprzedzające przygotowanie specjalistyczne

Przygotowanie specjalistyczne		Przygotowanie bazowe
dziedzina	poziom	
Technika mikroprocesorowa	I	II
	II	III
Komputerowe wspomaganie projektowania	I	II
	II	III
Systemy laboratoryjne	I	II
	II	III

Organizacja nauczania korzystania z techniki cyfrowej w przodujących wyższych uczelniach ZSRR

Doświadczenia Moskiewskiego Instytutu Stali i Stopów (MISIS)

W instytucie tym od 15 lat jest prowadzona planowa praca nad szerokim i wszechstronnym wykorzystaniem komputerów w procesie dydaktycznym. Dla każdej specjalności opracowano plan uczenia obejmujący cały okres szkolenia. W pierwszym etapie studentom wyklada się trzy przedmioty.

- Programowanie i metody numeryczne - wykład na II i III semestrze. Czas trwania kursu - 85 godzin, z czego 60% - zajęcia praktyczne, w czasie których każdy student rozwiązuje 6 - 8 zadań polegających na opracowaniu i uruchomieniu w klasach dysplejowych - typowych zadań inżynierskich. Warto podkreślić, że studenci programują w Fortranie i na naukę tego języka poświęca się większość czasu.

- Matematyczne planowanie eksperymentów. Na tym kursie (V-VI semestr) połowę czasu przeznacza się na praktyczne zajęcia z programowania.

- Zastosowanie komputerów i zautomatyzowanych systemów sterowania (ZSS). Ten przedmiot jest wykładany na semestrach VII-VIII. Ma on na celu nauczanie studentów modelowania i metod optymalizacji. W nowych programach nauczania ten przedmiot nazywa się "Teoria systemów i jej zastosowanie".

Drugi etap przygotowania studentów MISIS jest ukierunkowany na wypracowanie nawyków konkretnego zastosowania komputerów, np. do wielowariantowych obliczeń, do matematycznego modelowania procesów technologicznych, obróbki danych uzyskanych z eksperymentu, przy uczeniu się inżynierskich i naukowych dyscyplin wiedzy, przy wykonywaniu pracy naukowo-badawczej pod koniec studiów oraz przy projektowaniu w czasie wykonywania prac planowych.

Wprowadzanie komputerów do procesu dydaktycznego w Moskiewskim Instytucie Lotnictwa (MAI)

Metody zautomatyzowanego projektowania procesu nauczania wprowadza się w MAI od ponad 10 lat. Jest ono podporządkowane dwoóm zadaniom:

- ciągłego stosowania komputerów w całym okresie uczenia;

- przyzwyczajanie do korzystania z komputerów w ramach tradycyjnych przedmiotów z minimalną liczbą przedmiotów specjalnie ukierunkowanych na studiowanie zagadnień komputerowych.

Najpierw studenci MAI uczą się programowania w języku Fortran w ramach 50-godzinnego przedmiotu o nazwie "Języki algorytmiczne i programowanie". Praktyczne zajęcia przewidują rozwiązywanie prostych zadań obliczeniowych w reżimie dialogowym i wsadowym. W ciągu II i III roku studenci utrwalają swoje umiejętności programowania przy rozwiązywaniu praktycznych zagadnień z przedmiotów ogólnotechnicznych. Na III roku studenci słuchają wykładu z przedmiotu "Modelowanie zadań na komputerach". Równocześnie studenci specjalności konstrukcyjnych mają wykładany przedmiot "Podstawy systemu automatyzacji prac projektowych". Wiedza uzyskana przez studentów III roku z tych przedmiotów, jest wykorzystywana przy pracy praktycznej, przy czym studenci na tym etapie coraz częściej pracują z bibliotekami programów, zamiast osobiście programować wszystkie szczegóły. Wynika to z faktu, że problemy obliczeniowe są niejednokrotnie bardzo złożone i dla pojedynczego studenta byłoby zbyt pracochłonne programowanie całego swojego problemu; natomiast zbytne uproszczenie obliczeń byłoby zaprzeczeniem idei wykorzystania komputerów. Dlatego w większości katedr MAI istnieją odpowiednie biblioteki problemowych typowych modułów, z których studenci mogą zestawiać programy swoich obliczeń, zachowując złożoność, a nawet wielo-wariantowość tych problemów.

Na zakończenie całego kursu studentom wyklada się przedmiot "Systemowe projektowanie aparatów latających".

#### Nauczanie zastosowań komputerów w Moskiewskim Instytucie Inżynieryjno-Fizycznym (MIFI).

W Instytucie tym realizuje się program przyjęty przez Radę Naukową Instytutu w 1977 r. i skorygowany dla potrzeb programu nauczania na lata 1981-1985. Przyczyną korekty były zmiany samego programu nauczania.

Realizowany plan nauczania wiedzy i umiejętności komputerowych pozwolił zorganizować trzy etapowy cykl nauczania. Na pierwszym etapie studenci uczą się programowania oraz wykorzystania komputera dla opracowania wyników z eksperymentu fizycznego. Na etapie drugim pogłębiają znajomość języków programowania. Wreszcie na etapie trzecim wykorzystują komputer w oparciu o praktyki laboratoryjne, do wykonania prac samodzielnych, badawczych, do projektowania po IV lub V roku oraz do projektowania dyplomowego.

Na wydziale fizyczno-energetycznym zwraca się głównie uwagę na obliczanie bilansów cieplnych, optymalizację i projektowanie instalacji cieplno-energetycznych. Wydział automatyki i elektroniki stosuje komputery do organizacji projektowania systemowego i modelowania aparatury elektronicznej. Do programów nauczania tego wydziału są wprowadzone również takie przedmioty, jak "Mikroprocesory i mikrokomputery w systemach pomiarowych", "EMC w systemach rejestracji i obróbki informacji", "Zautomatyzowane systemy badań naukowych", "Automatyzacja akceleratorów cząstek naładowanych" i inne. Na wydziale cybernetyki przygotowuje się inżynierów-systemowców i matematyków opracowujących systemy automatyzacji badań naukowych i zautomatyzowanego projektowania. Dlatego też tu studiuje się w sposób pogłębiony systemy operacyjne, języki programowania PL-1 i Assembler, podstawy systemów automatyzacji projektowania, metody organizacji dialogu z EMC, problemy organizacji baz danych i sieci EMC.

#### Komputery w systemie nauczania Uniwersytetu im. M. Łomonosowa

Zarys systemu nauczania wiedzy komputerowej i zdobywania umiejętności z tej dziedziny przedstawia się następująco. Na I roku studiów zaznajamia się studentów z językiem wyższego poziomu - Basic. Następnie studenci mają praktykę matematyczną, założeniem której jest samodzielne rozwiązywanie problemów programowych w tym języku, z wykorzystaniem standardowych modułów Basic, a podczas wykonywania prac laboratoryjnych z fizyki, chemii itp. przedmiotów. Moduły

standardowe są napisane przez wykładowców i prowadzących ćwiczenia laboratoryjne ze studentami, dlatego uwzględniają typowe trudności, na jakie napotykają studenci.

Na II roku wprowadzona jest nauka Fortranu, w ramach zautomatyzowanego systemu uczenia się o nazwie "Nastawnik" ("Nychowawca"). Jest to system samodzielnego uczenia się programowania. W zależności od indywidualnych predyspozycji studentów "Nastawnik" zajmuje im od 15 do 30 godzin samodzielnej nauki programowania. Resztę czasu poświęcają programowaniu w Fortranie.

Po uzyskaniu niezbędnych umiejętności z zakresu programowania w językach algorytmicznych studenci wyższych lat (III roku i wyżej) uzyskują dostęp do systemu komputerowego o dużej mocy. Dostęp ten jest realizowany w dialogowym reżimie w ramach specjalnie do tego przeznaczonego systemu "KRAB".

#### Wykorzystanie komputerów w procesie dydaktycznym wyższych uczelni nietechnicznych

Do szkół o profilu nietechnicznym można zaliczyć przede wszystkim uczelnie medyczne i rolnicze. Innych nie będziemy tu analizować ze względu na to, w uczelniach o profilu humanistycznym lub nauk społecznych temat wykorzystania komputerów w procesie nauczania praktycznie nie istnieje.

Dzisiejszy poziom przygotowania pracowników służby zdrowia nie pozwala jeszcze efektywnie wprowadzać techniki obliczeniowej do medycyny i do praktyki ośrodków zdrowia. Niedostateczne przygotowanie absolwentów uczelni medycznych w dziedzinie techniki obliczeniowej jest spowodowane wieloma wzajemnie powiązаныmi czynnikami, a mianowicie:

- studenci uczący się w uczelniach medycznych nie są przygotowani do korzystania z metod matematycznych i techniki obliczeniowej w medycynie;
- wykładowcy uczelni nie mają nawyków w dziedzinie wykorzystania techniki obliczeniowej ani w swich pracach badawczych ani też w procesie nauczania;
- olbrzymia większość uczelni medycznych nie dysponuje kwalifikowanymi kadrami wykładowców cybernetyki medycznej (czyli specjalistów-wykładowców z dziedziny komputerów i metod matematycznych w medycynie i ochronie zdrowia);
- uczelnie medyczne i bazy kliniczne w przeważającej większości nie są wyposażone w ogóle lub niedostatecznie w środki techniki obliczeniowej.

Obecnie zapotrzebowanie służby zdrowia na specjalistów z zakresu cybernetyki medycznej jest częściowo zaspakajane przez dwa Moskiewskie Instytuty Medycyny oraz takie uczelnie, jak Moskiewska Szkoła Techniczna im. N.E. Bauma, MEI, MIFI i in.

Szczególnie duże doświadczenie we wprowadzaniu komputerów do procesu nauczania zdobył 2. Moskiewski Instytut Medyczny (2.MIM), który wspólnie z Republikańskim Informacyjno-Obliczeniowym Centrum Ministerstwa Zdrowia RSFR prowadzi prace nad stosowaniem techniki obliczeniowej w medycynie.

Wykładowcy katedry cybernetyki medycznej od 1971 r. prowadzą wykłady dla aspirantów i asystentów ordynatorów 2. Moskiewskiego Instytutu Medycyny oraz dla słuchaczy uczelni medycznych RSFR. Ponadto od 1979 r. zaczęli opuszczać te uczelnie absolwenci biofizyki, ze specjalizacją cybernetyka medyczna i cybernetyka biologiczna. Do roku szkolnego 1983/1984 opuściło uczelnię około 100 absolwentów. Ich poziom przygotowania spełnia oczekiwania praktyki, ale ich liczba jest zbyt mała w stosunku do potrzeb.

Ważnym ogniwem w przygotowaniu kadr dla gospodarki narodowej w Związku Radzieckim są uczelnie rolnicze. W 104 wyższych zakładach naukowych o profilu rolniczym kształcą się ponad 300 tysięcy studentów. Jednakże wprowadzenie techniki obliczeniowej i metod matematycznych do procesu nauczania uczelni rolniczych znajduje się w stadium początkowym. Plany nauczania w specjalnościach rolniczych nie przewidują nawet uczenia podstaw matematyki i innych przedmiotów,

które mogłyby być wstępem do dalszej nauki. Pewne przygotowanie w tym zakresie otrzymują tylko specjaliści z ekonomiki rolnictwa i z automatyzacji gospodarstw rolnych. Dlatego też w Związku Radzieckim, w kręgu specjalistów rolnictwa, rozpowszechniony jest pogląd o tym, że celowe jest w ramach zadań stawianych przez Partię w programie wyżywienia ludności organizować pomoc przodujących uczelni technicznych dla uczelni rolniczych w kształceniu kadr wykładowców i najpilniej potrzebnych specjalistów z dziedziny stosowania komputerów.

#### Podnoszenie kwalifikacji wykładowców

Jest sprawą zrozumiałą, że przygotowanie studentów jest niemożliwe bez specjalistycznego przygotowania wykładowców, przy czym przygotowaniem tym powinno być objęte jak najszersze grono pracowników dydaktycznych.

Podnoszenie kwalifikacji wykładowców MISIS było prowadzone dwoma etapami. W pierwszym etapie - były organizowane 36-godzinne seminary z oderwaniem od pracy (w czasie sesji zimowej i letniej). W etapie drugim - organizowano w ramach Fakultetu Podwyższenia Kwalifikacji (FPK) 600 godzinny kurs, w czasie którego wykładowcy uczyli się dodatkowych rozdziałów matematyki, programowania, matematycznego modelowania, metod optymalizacji i innych. Kurs taki jest obowiązany przejść każdy wykładowca jeden raz w ciągu 5 lat. Wymaganiem tym objęci są wszyscy wykładowcy w całym Związku Radzieckim. Należy tu odnotować, że Fakultety Podnoszenia Kwalifikacji (FPK) istnieją na większości uczelni radzieckich.

W MIFI przygotowanie wykładowców odbywa się w tzw. Szkole Praktyki Nauczenia liczącej 9 klas. W szkole tej słuchacze uczą się Fortranu, języka sterowania zadaniami w OSJS EMC, metody wykładowania i prowadzenia zajęć praktycznych w klasach displejowych, możliwości maszyn SM i ich szczególnych właściwości, współczesnych metod i środków automatyzacji eksperymentów naukowych, technicznych i programowych środków automatyzacji projektowania, możliwości mikroprocesorów i techniki mikroprocesorowej. W 1982 r. w Szkole Praktyki Nauczenia MIFI odbyło zajęcia 600 wykładowców i pracowników naukowych instytutu, w tej liczbie 13 kierowników katedr, 25 profesorów, 180 docentów. Na lata 1981-1985 Rada Instytutu przyjęła tzw. "Kompleksowy plan podwyższenia kwalifikacji profesorów, wykładowców i pracowników naukowych MIFI w dziedzinie wykorzystania środków techniki obliczeniowej w procesie nauczenia, pracy naukowo-badawczej i w sterowaniu". W Instytucie tym systematycznie odbywają się seminary na temat wykorzystania techniki obliczeniowej w procesie nauczenia i regularnie ukazuje się międzyuczelniany biuletyn "Zastosowania EMC i środków technicznych dydaktyki w procesie nauczenia".

Bardzo intensywne prace w dziedzinie podnoszenia kwalifikacji wykładowców prowadzi się w MAI, bądź na Fakultecie Podnoszenia Kwalifikacji bądź też na organizowanych przez poszczególne katedry kursach. W MEUT im. Bauman'a corocznie jest prowadzony kurs dla wykładowców, pracowników naukowych i aspirantów. Każda katedra wykorzystująca komputer kieruje na taki kurs jednego specjalistę, jako przyszłego konsultanta w dziedzinie rozwiązywania zadań na EMC obciążenie 1-2 godzin tygodniowo. Odpowiedzialność za przygotowanie tych specjalistów spoczywa na pracownikach katedry EMC. Ich zadaniem m.in. jest sprawdzenie programów, które następnie są obliczane.

Wszystkie te przykłady dowodzą, że podnoszenie kwalifikacji wykładowców, pracowników naukowych i aspirantów powinno mieć charakter ciągły, powinno obejmować tych wszystkich pracowników danej uczelni, którym może się to przydać bądź ze względu na wykładany przedmiot bądź też na charakter prowadzonej pracy badawczej. Opanowanie techniki obliczeniowej przez wykładowców powinno być obowiązkowe i stać się jednym z ważniejszych dowodów kwalifikacji zawodowych, co jest sprawdzane przy prowadzeniu tzw. atestacji. Atestacja jest to jedna z funkcji WAK - (Wyszej Atestacyjnej Komisji) przy Radzie Ministrów, która m.in. zatwierdza uchwały Rad Naukowych Instytutów w sprawie nadawania stopni naukowych kandydata nauk i doktora.

Doświadczenia w dziedzinie przygotowania studentów - podsumowanie

Pomiłając różnice występujące w systemach nauczania różnych uczelni, można wyodrębnić kilka cech wspólnych procesu przygotowania studentów do posługiwania się techniką obliczeniową:

- wykładowcy w ciągu całego procesu dydaktycznego są zobowiązani do przestrzegania dostarczonych im planów nauczania studentów w zakresie wykorzystywania EMC;
- kompleksowe i planowe wprowadzenie techniki obliczeniowej do procesu nauczania wszystkich cykli wykładowych, zarówno w przedmiotach ogólnie inżynierskich, jak i specjalistycznych;
- programy nauczania korzystania z komputerów opracowywane są na podstawie przeglądu poszczególnych dyscyplin naukowych, istniejących metod obliczeń, związane są z treścią zadań, przy czym odnosi się to, zarówno np. do projektów wykonywanych po zakończeniu roku nauki, tzw. przejściowych, jak i do projektów dyplomowych;
- tworzenie katedralnych pakietów programów użytkowych, zapewniających automatyzację prac projektowych i obliczeniowych, w zakresie podstawowych specjalności, w których dana katedra przygotowuje studentów.

Jak pokazują doświadczenia licznych uczelni w ZSRR, ciągle doskonalenie systemu nauczania studentów w korzystaniu z techniki obliczeniowej to:

- szerokie stosowanie pracy wieloprogramowej
- połączenie wsadowego przetwarzania zadań z pracą w trybie dialogowym, co wiąże się z koniecznością kompletowania rozwiniętych konfiguracji systemów komputerowych wyposażonych w wiele terminali
- zagwarantowanie wysokiej jakości przygotowania informacji wprowadzanej do komputera
- tworzenie znacznie większej liczby (audytorium) dysplejowych i większej liczby stanowisk do uczenia się wspomaganego komputerowo
- szersze wprowadzanie do procesu nauczania programowanych mikrokalculatorów, urządzeń mikroprocesorowych i mikrokomputerów.

W ZSRR ankietowano studentów na temat posiadania, typów i częstości używania kalkulatorów. Oto wyniki tej ankiety.

	Według studiów w (%)					Średnia
	I	II	III	IV	V	
Studenti posiadający kalkulatory	61,8	73,0	89,9	89,3	89,1	78,8
w tym: - programowalne	9,9	9,9	12,0	7,8	8,0	9,6
- z pamięcią na rejestrach	28,6	40,1	45,1	51,0	47,4	41,6
- z kompletem funkcji elementarnych	19,6	16,2	33,2	34,1	44,6	28,5
- z kompletem operacji arytmetycznych	13,8	11,2	11,9	10,6	14,7	12,4
Studenti nie posiadający kalkulatorów z przyczyny:						
- niepotrzebny	4,6	2,1	3,9	0,9	4,9	3,3
- niewygodny	2,8	6,1	12,1	4,6	6,5	6,3
- zawodny	7,4	12,0	23,5	25,3	20,3	16,6
- zbyt wysoka cena	30,3	28,9	28,7	22,6	18,8	26,3
- inne	8,7	5,7	3,7	4,6	6,3	5,9
częstość wykorzystania kalkulatorów						
- 4 - 6 razy w tygodniu	31,6	40,9	45,1	44,3	33,7	38,9
- 1 - 3 " " "	35,4	47,7	43,3	45,6	61,9	46,1
- 1 - 3 razy w miesiącu	7,2	5,2	7,3	5,2	15,2	7,9
- nie wykorzystują	9,5	4,7	2,7	2,1	6,3	5,3

#### Podsumowanie

W procesie kształcenia studentów w ZSRR wykorzystuje się od kilku lat technikę cyfrową, zarówno jako pomoc techniczną dla prowadzenia procesu dydaktycznego, jak i środek techniczny, z którym należy się zaznajomić i nauczyć się obsługiwać, aby w przyszłej pracy zawodowej móc go racjonalnie wykorzystać.

Sam proces wprowadzania komputerów do toku studiów przeszedł z okresu indywidualnego traktowania problemu przez wyższe uczelnie w okres zinstytucjonalizowanej działalności. Powstały odpowiednie programy nauczania, które muszą być przez wyższe uczelnie realizowane. Warto zwrócić uwagę, że okres ten wynosi 10-15 lat. Należy zdawać sobie z tego sprawę rozpatrując w Polsce problem wprowadzania komputerów do szkół.

Artykuł opracowano na podstawie materiałów:

"Wykorzystanie środków techniki cyfrowej w procesie nauczania w wyższych uczelniach" - Moskwa 1984, Instytut Problemów Szkoły Wyższej.

mgr Tadeusz KUROCZYCKI  
 mgr Teresa ORZEL  
 mgr Małgorzata PLUCIŃSKA  
 Instytut Maszyn Matematycznych

## Oprogramowanie elementarnej sieci terminalowej - TELEIMM utworzonej na bazie sprzętu IBM i JS

W ramach optymalizowania postaci oprogramowania sieci terminalowych, złożonych z urządzeń produkcji krajowej, postanowiono utworzyć w IMM eksperymentalną sieć terminalową przeznaczoną do testowania poszczególnych składowych oprogramowania sieci. W opracowaniu przedstawiliśmy postać omawianej sieci oraz zaprezentowaliśmy testowane składowe oprogramowania.

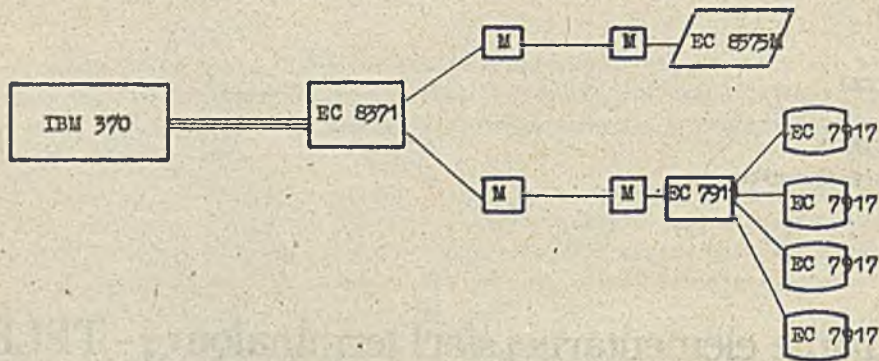
W chronologicznym rozwoju podstawowych struktur teleprzetwarzania TP można wyróżnić następujące etapy:

- bezpośrednio (lokalne) połączenie urządzeń wejścia/wyjścia (terminali) z komputerem centralnym,
- zdalne połączenie terminali z komputerem,
- sieć terminalowa,
- sieć komputerowa

Sieć terminalowa składa się z komputera centralnego (węzeł główny), jednego lub kilku procesorów telekomunikacyjnych (węzły sterujące) oraz terminali lub grup terminali (węzły końcowe) połączonych z węzłem sterującym za pośrednictwem modemów i linii komunikacyjnych.

Sieć terminalowa TELEIMM składa się z następujących elementów:

- |                    |   |  |
|--------------------|---|--|
| węzeł<br>główny    | { | - BMC IBM 370/145,   |
| węzeł<br>sterujący | { | - procesor telekomunikacyjny EC 8371,  |
| węzeł<br>końcowy 1 | { | - grupa monitorów ekranowych EC 7917 sterowana przez jednostkę EC 7911 pracującą w trybie BSC, |
| węzeł<br>końcowy 2 | { | - terminal EC 8575M pracujący w trybie asynchronicznym.  |



Rys. 1. Eksperymentalna sieć terminalowa TELEIEM

W oprogramowaniu sieci terminalowej można wyróżnić zasadniczo trzy poziomy:

- systemowy,
- sterujący,
- użytkowy (zastosowaniowy).

Na poziomie systemowym organizuje się transmisję danych w ramach sieci. Zadanie to spełnia telekomunikacyjna metoda dostępu (np. TCAM, VTAM) będącą składową systemu operacyjnego, pod którego kontrolą pracuje komputer centralny. Może to nastąpić, gdy sprzętowe elementy sieci zostaną zgłoszone systemowi operacyjnemu na etapie generacji.

Na poziomie sterującym odbywa się kierowanie "ruchem" danych (komunikatów) w ramach sieci. Przepływ komunikatów może następować w relacjach:

- < terminal > - < terminal > ,
- < terminal > - < program użytkowy > ,
- < program użytkowy > - < program użytkowy > ,

przy czym jako < terminal > przyjmuje się dowolne urządzenie końcowe akceptowane przez telekomunikacyjną metodę dostępu. Za < program użytkowy > może być tu uważany dowolny program rezydujący w pamięci operacyjnej komputera centralnego, mogący pobierać dane z wyznaczonych zbiorów kolejkowych oraz mający możliwość wysłania odpowiednio zredagowanych danych. Program ten może być napisany, zarówno w języku Assembler, jak i w języku wyższego rzędu, np. PL/I.

Na poziomie sterującym kierowanie przepływem komunikatów odbywa się w dwu węzłach: sterującym i głównym. Węzeł sterujący zlokalizowany w procesorze komunikacyjnym:

- odbiera komunikaty z terminala i kieruje je do właściwego punktu docelowego lub
- odbiera komunikaty nadawane przez węzeł główny i kieruje je do zadanego terminala lub grupy terminali. Do wybrania zadania realizuje program sterujący działający w pamięci procesora.

Najbardziej w węzle głównym odbywa się sterowanie komunikatami przychodzącymi z programów użytkowych działających w pamięci operacyjnej komputera oraz kierowanie komunikatu przesłanego przez



procesor telekomunikacyjny do określonego punktu użytkowego. Sposób realizacji funkcji sterowania przepływem komunikatów przez węzeł główny zależy od konkretnej, telekomunikacyjnej metody dostępu, pod nadzorem której następuje przepływ danych przez sieć.

Na poziomie użytkowym następuje "obróbka" danych i zwrotne przekazanie informacji do poziomu sterowania w węźle głównym.

Opracowując koncepcję testowego oprogramowania sieci TELEIMM postanowiono:

- wykorzystać sprzętową własność pamięci wirtualnej, posiadaną przez EMC IBM 370/145, przez wybór systemu VS1 - wersja 7 (najnowsza),
- pójść dwoma równoległymi drogami testowania telekomunikacyjnych metod dostępu,
  - wybrać metodę dostępu VTAM opracowaną pod kątem pracy w ramach "systemów wirtualnych" (tzn. opartych na opozi pamięci wirtualnej),
  - ze względu na dużą popularność w kraju metody TCAM wykorzystać ją również i to w najnowszej wersji - TCAM10; wersja ta może działać jedynie pod kontrolą "systemów wirtualnych",
- oprzeć się na najnowszej wersji programu sterującego siecią (działającego w procesorze telekomunikacyjnym) - NCP3.

#### Program sterujący siecią - NCP

Jeśli grupy zdalnych terminali są podłączone do komputera centralnego przez taką jednostkę teletransmisji, jak np. urządzenie IBM 2703, praca na nich w dużym stopniu obciąża zasoby komputera. Podprogramy telekomunikacyjnych metod dostępu organizują w tym wypadku logiczną strukturę powiązań w sieci, przepływ komunikatów, a także przystosowują transmisję do fizycznych wymagań trybu przesyłania i nawiązują kontakt z terminalami na liniach. W ostatnich latach na miejsce tradycyjnej jednostki teletransmisji pojawił się procesor telekomunikacyjny, będący komputerem łączącym maszynę główną z siecią telekomunikacyjną. Odpowiednio oprogramowany procesor może sterować siecią lub spełniać funkcje zwykłych jednostek teletransmisji. Procesor ma własną pamięć (np. model 8371 w IMM - 112 k bajtów), rejestry, od strony maszyny wyposażony jest w adapter kanałowy (jeden lub kilka), od strony linii - w skaner (jeden lub kilka). Odpowiednie instrukcje wejścia/wyjścia pozwalają mu na odczytywanie oraz wysyłanie informacji na linie i do kanału, a instrukcje maszynowe - na obróbkę tych informacji. Sprzętowy, 5-poziomowy system przerwań jest dostosowany do płynnej obsługi jakościowo zróżnicowanej transmisji w kanale i na liniach.

#### Budowa programu NCP

Program sterowania siecią zwany programem NCP, działający w procesorze telekomunikacyjnym, przygotowywany jest na maszynie głównej (IBM, R1ad), za pomocą makroinstrukcji generacyjnych, rozwijanych następnie przez Assembler procesora i łączony przez program linkage editor. Ładowanie programu NCP do procesora odbywa się za pomocą specjalnego programu ładującego, który w tym celu wykorzystuje tylko jeden adres podkanału w kanale maszyna-procesor.

Makrodefinicje służące do generowania programu NCP można podzielić na dwie grupy. Do pierwszej zaliczają się makrodefinicje takie jak BUILD, CSB, HOST, SYSCNTRL, specyfikujące parametry ogólne procesora programu NCP oraz styku z telekomunikacyjną metodą dostępu. W makroinstrukcji BUILD podaje się m.in. typ i nazwę generowanego programu, wielkość pamięci i typ procesora, rodzaj i liczbę jego adapterów kanałowych, nazwy zbiorów wykorzystywanych w procesie generowania. Poniżej przedstawiono makroinstrukcję BUILD zastosowaną do generowania programu NCP dla procesora 8371:

```
BUILD MODEL=8371, MEMSIZE= 112, TYPCEN=NCP,  
      SUBAREA=3, MAXSUBA=7, BFRS=80,  
      CA=TYPE1, CHANTYP=TYPE1, TYPESYS=OS,  
      LOADLIB=LOAD8371, OBJLIB=OBJNCP, UT1=NCPUT1,  
      UT2=NCPUT2, UT3=NCPUT3, JOBCARD=MULTI, ERASE=YES,  
      SLOWDOWN=25, UNIT=2314, ASMXREF=YES, ABEND=YES,  
      TRACE=(YES,50)
```

Wymagane przez metodę dostępu funkcje dynamicznej regulacji parametrów sieci definiowane są w makroinstrukcji SYSCNTRL. W makroinstrukcji HOST określa się buforę dla danych z telekomunikacyjnej metody dostępu oraz czasokresy związane z transmisją kanałową. Makroinstrukcje CSB służą do opisu skanerów procesora. Drugą grupę stanowią makroinstrukcje definiujące sieć teletransmisji. Należą tu takie makroinstrukcje jak: GROUP (opisuje grupy linii o jednakowym trybie przesyłania - asynchronicznym, BSC lub SDLC), LINE, CLUSTER, TERMINAL, SERVICE. Załączony przykład przedstawia makroinstrukcje programu NCP opisujące linię BSC, złożoną z 4 monitorów 7917, współpracującą z NCP w trybie sieciowym:

```
G17911 GROUP LNCTL=BSC, TYPE=NCP  
L7911 LINE ADDRESS=022, SPEED=2400, CRITSIT=YES,  
        INTPRI=2, CLOCKNG=EXT, CODE=EBCDIC,  
        POLLED=YES, SESSION=3, SERVLIM=2,  
        PAUSE=5, TRANSFR=3, CUTOFF=1, CDATA=YES  
        SERVICE ORDER= (CLUSTER1, TER1, TER2, TER3, TER4)  
CLUSTER1 CLUSTER CUTYPE=7911, TERM=7917, GPOLL=40407F7F,  
        XMITLIM=3  
TER1 TERMINAL ADDR=60604040, POLL=40404040  
TER2 TERMINAL ADDR=6060C1C1, POLL=4040C1C1  
TER3 TERMINAL ADDR=6060C2C2, POLL=4040C2C2  
TER4 TERMINAL ADDR=6060C3C3, POLL=4040C3C3
```

Podstkowe makroinstrukcje przeznaczone są do opisu sieci działającej wg protokołu SDLC.

W programie NCP opisane są linie telekomunikacyjne wiążące poszczególne składowe sieci. Linie te nie wymagają oddzielnych adresów podkanałowych i dlatego wielkość sieci sterowanej przez program NCP nie jest ograniczona liczbą dostępnych adresów podkanałów w kanale maszyna-procesor. Program NCP wersja 5, łącznie z telekomunikacyjną metodą postępu, może obsługiwać

różnorodną sieć, złożoną zarówno z terminali działających w trybie asynchronicznym, BSC jak i w SCLC.

#### Program NCP a telekomunikacyjna metoda dostępu

Program NCP przejmując część dawnych zadań telekomunikacyjnych metod dostępu jak: sondowanie i adresowanie terminali na linii wielopunktowej, wstawienie i usuwanie znaków sterowania linią w blokach danych, wymagane przekodowywanie, alokowanie buforów dla danych przychodzących z terminali i maszyny, wstawianie informacji o dacie. Gdy mamy do czynienia z dużą siecią zyskujemy w ten sposób znaczną oszczędność pamięci maszyny głównej, przeznaczając ją np. na pracę systemem wsadowym. Ponadto program NCP ma wiele procedur diagnostycznych, uruchamianych w razie wystąpienia błędu na linii.

Każda telekomunikacyjna metoda dostępu narzuca pewne ograniczenia i wymagania dotyczące budowy i parametrów programu NCP; inne są wymagania np. dla NCP dla VTAM-u, inne dla TCAM-u. Dlatego każda z tych metod współpracuje ze swoim programem NCP w procesorze. Wygenerowanie kilku różnych programów NCP ma również uzasadnienie w związku z różnymi zapotrzebowaniami na rodzaj i wielkość sieci. Warto też wiedzieć, że za pomocą komend metody dostępu można w sposób dynamiczny zmieniać niektóre parametry programu NCP (np. w TCAM-ie czasokres sondowania, interwał systemowy itp.), a także można aktywować i dezaktywować wybrane połączenia sieci tworząc do pracy konkretny jej podzbiór.

#### Program emulacyjny - EP

Procesor telekomunikacyjny może działać jako zwykła jednostka sterująca teletransmisją. Ten wariant pracy jest wymagany np. jeśli system operacyjny w komputerze centralnym nie zawiera sieciowej wersji telekomunikacyjnych metod dostępu, lecz programy przystosowane do komunikowania się z terminalami podłączonymi za pośrednictwem jednostki IBM2701, IBM2702 lub IBM2703. Chcąc procesor przekształcić funkcjonalnie w jednostkę teletransmisji, należy do niego załadować specjalny program emulujący jej działanie, zwany emulatorem EP i uruchomić go. Generowanie programu EP odbywa się podobnie jak generowanie programu NCP, ale jest procesem prostszym i krótszym.

#### Program PEP

Program NCP można wygenerować w postaci PEP (Partitioned Emulation Program) czyli rozszerzonej wersji NCP, umożliwiającej jednocześnie pracę procesora jako jednostki teletransmisji IBM 2701, 2702 lub 2703 (lub dowolnej ich kombinacji) dla pewnych linii (tryb pracy EP) i wykonywanie typowych funkcji programu NCP (tryb pracy NCP) dla innych linii tej samej sieci. Możliwa jest także praca linii (przy odpowiednim jej wygenerowaniu) w trybie PEP - w danej chwili linia pracuje albo w trybie NCP albo w trybie EP, a wyborem trybu pracy steruje metoda dostępu. Zaletą programu sterowania siecią w wersji PEP jest jego uniwersalność, przydatność, zarówno dla metod dostępu przeznaczonych do komunikowania się z terminalami przez IBM 2701, 2702 lub 2703, jak i dla programów przeznaczonych do komunikowania się z NCP.

### Telekomunikacyjna metoda dostępu

Głównymi zadaniami telekomunikacyjnej metody dostępu w ramach filozofii teleprzetwarzania prowadzonego na sprzęcie JS są:

- zapewnienie możliwości przepływu danych w ramach struktury teleprzetwarzania oraz możliwości modyfikowania i wykorzystywania przekazywanych danych - mamy tu na myśli głównie programy użytkowe działające w pamięci komputera centralnego,
- nadzór nad przebiegiem sesji teleprzetwarzania.

Przygotowanie się do sesji od strony telekomunikacyjnej metody dostępu (poziom systemowy) obejmuje m.in. następujące czynności:

- zdefiniowanie składowych wykorzystywanej struktury teleprzetwarzania wraz z podaniem:
  - ogólnej postaci struktury,
  - typów terminali i procesorów telekomunikacyjnych uczestniczących w transmisji danych (np. EC 7917),
  - modeli terminali i własności funkcjonalnych związanych ze wskazanym modelem (np. EC8575 z opcją "checking"),
  - adresów linii łączących poszczególne składowe, względnej adresacji w ramach grupy linii oraz względnej adresacji w ramach linii wielopunktowej,
- określenie trybu działania w ramach struktury przez ustalenie:
  - protokołu, wg którego odbywać się będzie transmisja danych,
  - sekwencji sterowania linią, którymi zostają obudowane przesyłane komunikaty,
  - sekwencji identyfikujących poszczególne składowe, związanych z sondowaniem i adresowaniem,
  - przebiegu sondowania i adresowania terminali,
  - reguł adresowania komunikatów i kierowania ich do zadanych składowych biorących udział w transmisji; jako składowe rozumie się tu zarówno terminale, jak i programy użytkowe działające w pamięci komputera centralnego,
- organizację przepływu danych w ramach rozpatrywanej struktury teleprzetwarzania przez rozwiązanie problemów związanych:
  - z buforowaniem (statystyczne, dynamiczne),
  - kolejkowaniem komunikatów (w pamięci wewnętrznej, w pamięci zewnętrznej),
  - przechowywaniem komunikatów (naturalnym lub wymuszonym),
  - przekodowywaniem komunikatów,
- wybór sposobu obsługi błędów występujących podczas sesji.

Wraz z postępującym rozwojem struktur teleprzetwarzania pod względem ich rodzaju, wielkości, zasięgu, technik transmisji danych i różnorodności składowych (nowe typy terminali), wprowadzeniem do eksploatacji nowych rodzin komputerów (np. działających z wykorzystaniem opcji pamięci wirtualnej) nastąpiła ewolucja telekomunikacyjnych metod dostępu.

We wczesnych telekomunikacyjnych metodach dostępu użytkownik musiał dbać, zarówno o włas-

oiwe zorganizowanie przebiegu transmisji danych (określenie kolejności, w jakiej będzie nawiązywana łączność z poszczególnymi terminalami w celu przesłania lub pobrania danych, zapewnienie właściwego przebiegu sondowania i adresowania, sterowania liniami, przekodowywanie komunikatów), jak i o przetworzenie transmitowanej informacji. Użytkownik był zmuszony zejść do poziomu logicznego sterowania sesją. W przypadku większej bardziej rozwiniętej struktury było to dość uciążliwe, zwłaszcza że należało przewidzieć konieczność modyfikowania wykorzystywanej konfiguracji w trakcie eksploatacji programu. Przykładem telekomunikacyjnej metody dostępu tego typu jest Basic Telecommunications Access Method (BTAM).

Później opracowane telekomunikacyjne metody dostępu wprowadziły rozdzielenie części systemowej programu (organizacja i nadzór nad przebiegiem transmisji danych) od części użytkowej (przetwarzanie komunikatów). TCAM i VTAM - dwie metody wchodzące w skład oprogramowania sieci TELEIMM można zaliczyć do takiego właśnie typu.

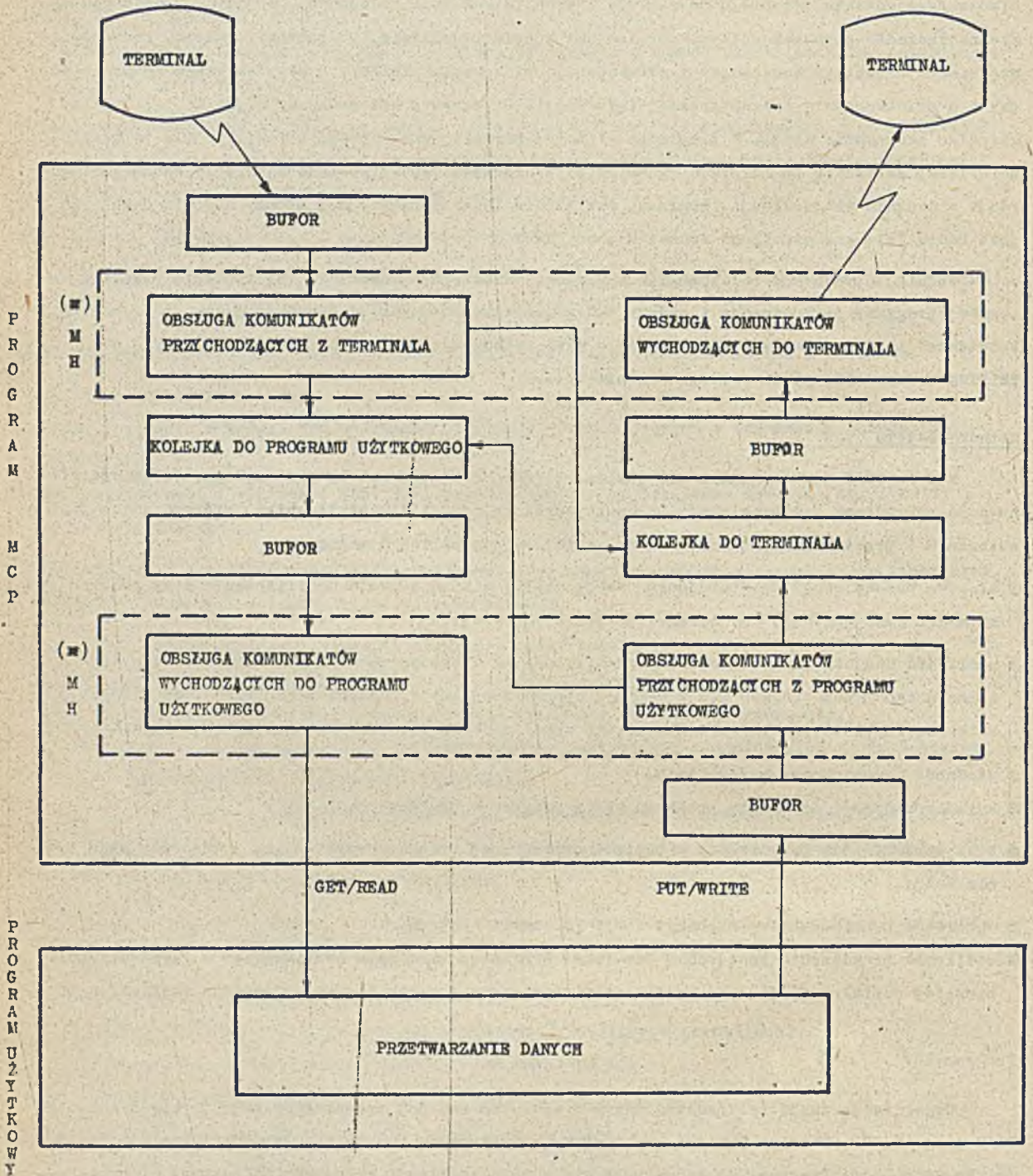
#### Metoda dostępu TCAM

Metoda TCAM jest wygodnym narzędziem, zarówno dla programisty systemowego, odpowiedzialnego za prawidłowy przebieg sesji teleprzetwarzania, jak i dla użytkownika, którego celem jest zbieranie i przetwarzanie danych. Do zalet tej metody zaliczyć można:

- blokową budowę programu sterującego sesją (MCP), a stąd łatwość modyfikowania tego programu przy zmianach konfiguracji, czy parametrów przebiegu sesji.
- możliwość dwojakiego kolejkowania komunikatów: na dysku co pozwala na zmniejszenie obszaru zajmowanego przez program MCP w pamięci operacyjnej lub w pamięci wewnętrznej, co z kolei zwiększa szybkość przepływu komunikatów; wybór sposobu kolejkowania należy do programisty,
- własność dynamicznego buforowania,
- asynchroniczność działania programu MCP i programów użytkowych,
- brak konieczności rezydowania w pamięci operacyjnej programu użytkowego, przez cały czas trwania sesji,
- możliwość kontynuowania przebiegu sesji po awarii systemu,
- możliwość prowadzenia testowania "on-line" wybranego terminala (wchodzącego w skład struktury biorącej udział w sesji) bez konieczności zawieszania pracy innych składowych konfiguracji.

#### Program MCP

Organizacją sesji teleprzetwarzania i nadzorem nad jej przebiegiem zajmuje się w metodzie TCAM program MCP (Message Control Program) wygenerowany za pomocą specjalnych makroinstrukcji języka Assembler. Program MCP komunikuje się z poszczególnymi składowymi strukturami TP organizując przepływ komunikatów we wszystkich podstawowych relacjach (rys. 2).



Rys.2. Przepływ komunikatów pod nadzorem metody TCAM

(\*) - Blok bezpośrednio obsługujący przesyłane komunikaty

STRUKTURA LOGICZNA

USTALENIE PARAMETRÓW CHARAKTERYZUJĄCYCH PRZEbieg SESJI
OKREŚLENIE ● BUFOROWANIA ● KOLEJKOWANIA
ZGŁOSZENIE ● SKŁADOWYCH SIECI UCZESTNICZĄCYCH W SESJI TP
OKREŚLENIE SEKWENCJI ADRESOWYCH
ZGŁOSZENIE ● PROGRAMÓW UŻYTKOWYCH WYKORZYSTYWA- NYCH W RAMACH SESJI TP
ZAINICJOWANIE SESJI TP
STEROWANIE PRZEBIEGIEM SESJI TP  (PROGRAM MH)
ZAKOŃCZENIE SESJI TP

STRUKTURA PROGRAMOWA  
(WYDANIE MAKROINSTRUKCJI GENERUJĄCYCH MCP)

INTRO

INTRO    DCB    PCB  
TERMINAL    TPROCESS  
LOGTYPE

GROUP  
LINE  
CLUSTER TERMINAL  
DCB

INVLIST

TPROCESS  
PCB

READY    OPEN

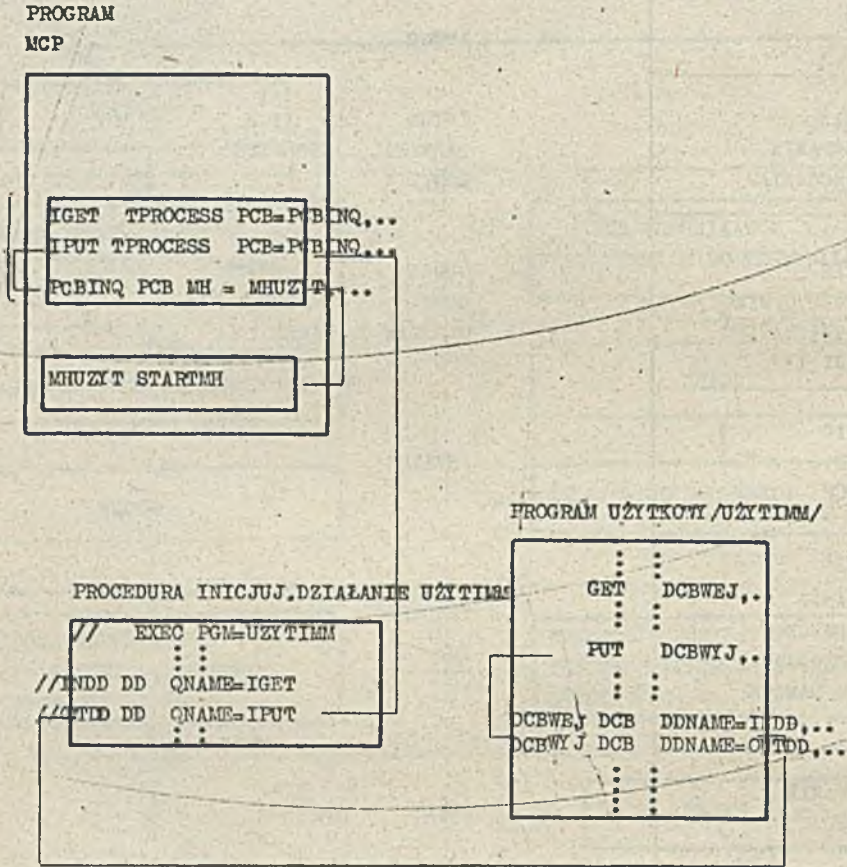
- grupa makroinstrukcji:
- generująca program MH,
  - rozpoczynająca się wydaniem STARTMH
  - kończąca się wydaniem OUTEND

CLOSE

Tys. 3. Ogólna struktura programu MCP

### Program MCP, a program użytkowy

Zabezpieczenie wymiany danych między programem MCP, a programem użytkowym następuje, z wykorzystaniem mechanizmu kolejkowego, w sposób podany poniżej (rys. 4):



Rys.4. Powiązanie między programem MCP, a jednym z obsługiwanych przez niego programów użytkowych

### Program MCP, a rostać struktury teleprzetwarzania

Metoda dostępu TCAM może być wykorzystywana do oprogramowania, zarówno prostych jak i złożonych tele-struktur.

Poniżej przedstawiamy sposób zgłaszania urządzeń końcowych programowi MCP. Przykład dotyczy fragmentu konfiguracji utworzonej w IBM (TELEIBM).  
Wariant 1 obejmuje przypadek nie-sieciowy, tzn. gdy w procesorze komunikacyjnym działa program emulujący działanie jednostki IBM 2703. Wariant 2 jest wariantem sieciowym - w procesorze komunikacyjnym działa program MCP (zob. rozdz. "Program sterujący siecią - MCP).



Warient 1

7900R DCB DDNAME=DD7900, DSORG=TX, MACRF=(G,P),  
BUFIN=3, BUFOUT=2, BUFMAX=5, BUFSIZE=2000,  
MH-MH7900, INVLIST=(INVX)

TER1 TERMINAL QBY=T, DCB=L7900R, TERM=790R, QUEUES=MO,  
ADDR=606040402D, RLN=1

TER2 TERMINAL QBY=T, DCB=L7900R, TERM=790R, QUEUES=MO,  
ADDR=6060C1C12D, RLN=1

TER3 TERMINAL QBY=T, DCB=L7900R, TERM=790R, QUEUES=MO,  
ADDR=6060C2C22D, RLN=1

TER4 TERMINAL QBY=T, DCB=L7900R, TERM=790R, QUEUES=MO,  
ADDR=6060C3C32D, RLN=1

INVX INVLIST ORDER=(TER1+404040402D, TER2+4040C1C12D,  
TER3+4040C2C22D, TER3+4040C3C32D), EOT=37

MH7900 STARTMH LC=OUT

Warient 2

PTD DCB DSORG=TR, DDNAME=DD8371, MACRF(G,P)

PRZYKL TERMINAL TERM=LVCP, DCB=PTD

GP7911 GROUP LCB=PTD, BUFIN=3, BUFOUT=2, BUFMAX=5,  
BUFSIZE=2000,  
MH-MH7900

L7911 TERMINAL TERM=LINE, GROUP=GP7911, RLN=1

CLUSTER1 TERMINAL TERM=790C, GROUP=GP7911, QBY=T, RLN=1

TER1 TERMINAL TERM=790R, QBY=T, GROUP=GP7911,  
QUEUES=MO, UTERM=NO, COMP=NO,  
GPCLSTR=CLUSTER1, RLN=1

TER2 TERMINAL TERM=790R, QBY=T, GROUP=GP7911,  
QUEUES=MO, UTERM=NO, COMP=NO,  
GPCLSTR=CLUSTER1, RLN=1

TER3 TERMINAL TERM=790R, QBY=T, GROUP=GP7911,  
QUEUES=MO, UTERM=NO, COMP=NO,  
GPCLSTR=CLUSTER1, RLN=1

TER4 TERMINAL TERM=790R, QBY=T, GROUP=GP7911,  
QUEUES=MO, UTERM=NO, COMP=NO,  
GPCLSTR=CLUSTER1, RLN=1

IFDSCPMH

MH7900 STARTMH LC=OUT

Relacje między składowymi strukturą TP, a programami użytkowymi

Ustalenie połączenia między wybranymi składowymi strukturą dla potrzeb danej sesji może być typu:

- stałego, tzn. ustalone na cały czas trwania sesji TP, przez umieszczenie odpowiedniej informacji w programie MCP (część sterująca tzw. "message handler"),
- chwilowego, tzn. na czas przepływu jednego komunikatu przez umieszczenie informacji o wykorzystywanej relacji ruchu w przesyłanej sekwencji.

Wykorzystywanie drugiego wariantu nakłada dodatkowo obowiązki na autorów programów użytkowych lub na użytkowników operujących na terminalach.

Przegląd wersji metody dostępu TCAM

Wczesne wersje TCAM-u (np. TCAM4) są związane z organizowaniem teleprzetwarzania jedynie w ramach nie-sieciowych konfiguracji będących połączeniem terminali z komputerem centralnym przez jednostkę sterującą transmisją (np. IBM 2703) - przezroczystą z punktu widzenia oprogramowania. Wersje te mogą działać pod kontrolą systemu operacyjnego OS.

Do organizowania teleprzetwarzania w ramach sieci przeznaczone są wersje: TCAM5 (w zasadzie tylko pod kontrolą systemu OS) i późniejsze. Należy podkreślić, że po wersji TCAM5 kolejną, a pełni autonomiczną wersją jest TCAM10, pracujący wyłącznie pod kontrolą systemu VS. Natomiast wersje pośrednie - TCAM8 i TCAM9 działają w trybie sieciowym wyłącznie we współpracy z metodą dostępu VTAM.

#### Metoda dostępu VTAM

VTAM jako wirtualna telekomunikacyjna metoda dostępu (Virtual Telecommunications Access Method) organizuje przesyłanie danych między programami użytkowymi działającymi w komputerze centralnym, a terminalami (lokalnymi lub zdalnymi) wchodzącymi w skład sieci telekomunikacyjnej.

Metoda ta wymaga, aby wśród rozkazów maszynowych dostępne były instrukcje Compare and Swap (CS) i Compare Double and Swap (CDS). Ponieważ konfiguracja istniejąca w IMM nie zawierała tej opcji, dokonano jej programowej symulacji w systemie VS1.

VTAM wprowadza pojęcie sieci VTAM-owskiej, w skład której wchodzi: jeden lub kilka procesorów telekomunikacyjnych, terminale lub grupy terminali, program lub programy sterowania siecią MCP działające w procesorach telekomunikacyjnych oraz programy użytkowe wykorzystujące metodę dostępu VTAM, a działające w komputerze centralnym.

VTAM wymaga założenia i zakatalogowania trzech zbiorów bibliotecznych:

- SYS1.VTAMLIB - zawierającego moduły sterujące VTAM-u i zapełnianego w czasie generowania systemu VS1 z opcją VTAM,
- SYS1.VTAMLST - składającego się z członów definiujących sieć VTAM-owską oraz parametry startowe VTAM-u,

● SYS1.VTAMOBJ - wykorzystywanego przez VTAM do tworzenia tablic konfiguracji sieci VTAM-owskiej, w celu szybkiego uaktywniania węzłów w sieci.

#### Definiowanie sieci VTAM-owskiej

Wszystkie elementy wchodzące w skład sieci VTAM-owskiej (terminale, programy użytkowe, NCP, itp.) definiowane są we VTAM-ie jako tzw. węzły nadrzędne i podrzędne. Definicje węzłów wprowadza się do zbioru SYS1.VTAMLST w postaci zdań specjalnego języka VTAM-u. Do węzłów nadrzędnych zalicza się grupę programów użytkowych, grupę terminali i program NCP. Węzłami podrzędnymi są poszczególne programy użytkowe, terminale oraz linie komunikacyjne działające w trybie NCP.

Każdy nadrzędny węzeł wyznaczony jest jednoznacznie przez człon zbioru SYS1.VTAMLST, zawierający definicje przyporządkowanych mu węzłów podrzędnych. Etykiety zdań VTAM-u definiujących węzły podrzędne są nazwami tych węzłów, natomiast nazwą węzła nadrzędnego jest nazwa członu zbioru SYS1.VTAMLST. Nazwy nadrzędnych węzłów wykorzystywane są przy definiowaniu wstępnej (standardowej) konfiguracji sieci, a następnie w czasie eksploatacji systemu, gdy zachodzi konieczność zmiany konfiguracji.

Nadrzędny węzeł programów użytkowych definiowany jest przez człon zbioru SYS1.VTAMLST, zawierający jedno lub kilka zdań APPL. Każde zdanie APPL tworzy węzeł podrzędny i służy do definiowania programu użytkowego wykorzystującego VTAM. Etykieta tego zdania musi być identyczna z nazwą programu, zakodowaną w bloku ACB utworzonym przez VTAM-owską makroinstrukcję ACB i użytą w danym programie. Programy użytkowe VTAM-u mają możliwość wysyłania i odbierania komunikatów, używając odpowiednich makroinstrukcji VTAM (SEND, RECEIVE, READ, WRITE) i działają niezależnie od sposobu przesyłania danych i rodzajów połączenia terminali.

Węzeł nadrzędny dla lokalnej konfiguracji terminali definiujemy tworząc człon w SYS1.VTAMLST, zawierający zdanie VTAM-owskie LBUILD, określające jednostkę sterującą EC 7911 oraz zdania LOCAL, określające lokalne terminale EC 7917 jako węzły podrzędne.

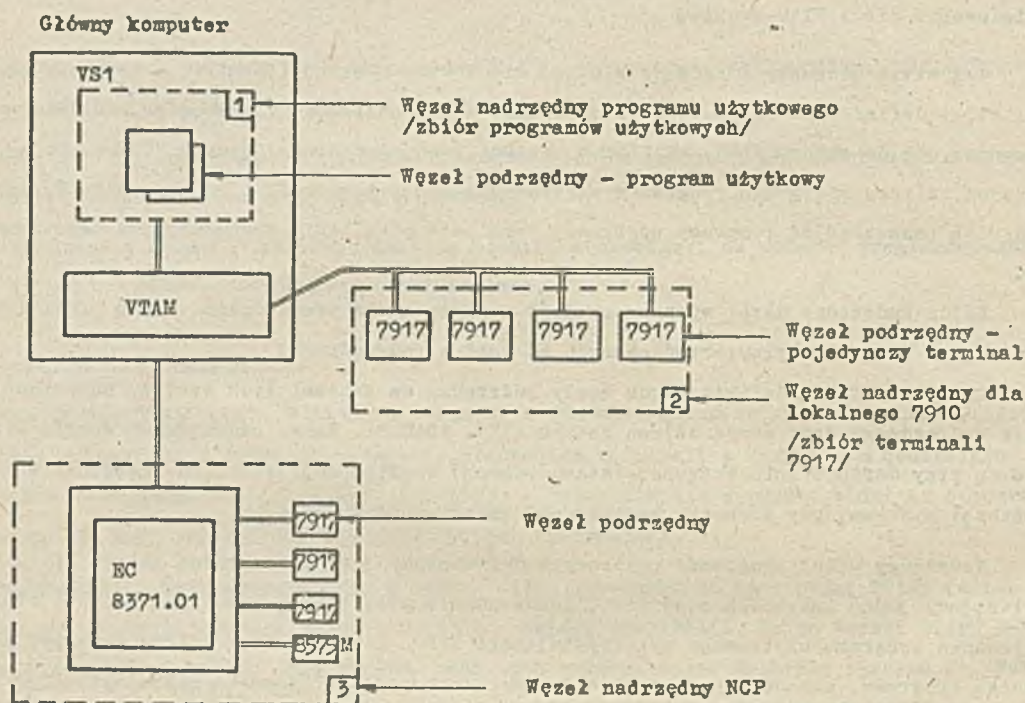
Węzeł nadrzędny NCP określamy tworząc człon w zbiorze SYS1.VTAMLST, składający się z pliku źródłowego programu NCP, poprzedzonego VTAM-owskim zdaniem PCCU. W zdaniu PCCU identyfikuje się programowalny procesor telekomunikacyjny EC 8371, którego pracą steruje dany program NCP.

Taki sposób definiowania sieci i pracy VTAM-u daje mu istotną przewagę nad innymi telekomunikacyjnymi metodami dostępu, jak TCAM, BTAM.

VTAM ułatwia tworzenie i obsługę nowych konfiguracji sieci bez konieczności zmiany programów, a także daje możliwość dynamicznego zmieniania parametrów sieci bez przerywania pracy systemu. Wystarczy tylko dokonać odpowiednich zmian w zbiorze SYS1.VTAMLST i zrealizować odpowiednio operacje połączenia (logon). TCAM i BTAM wymagają w takiej sytuacji zasadniczych zmian w programach obsługi terminali.

Konkretna konfiguracja sieci telekomunikacyjnej i struktura plików definiujących węzły przedstawiona jest na rys. 5 i 6.

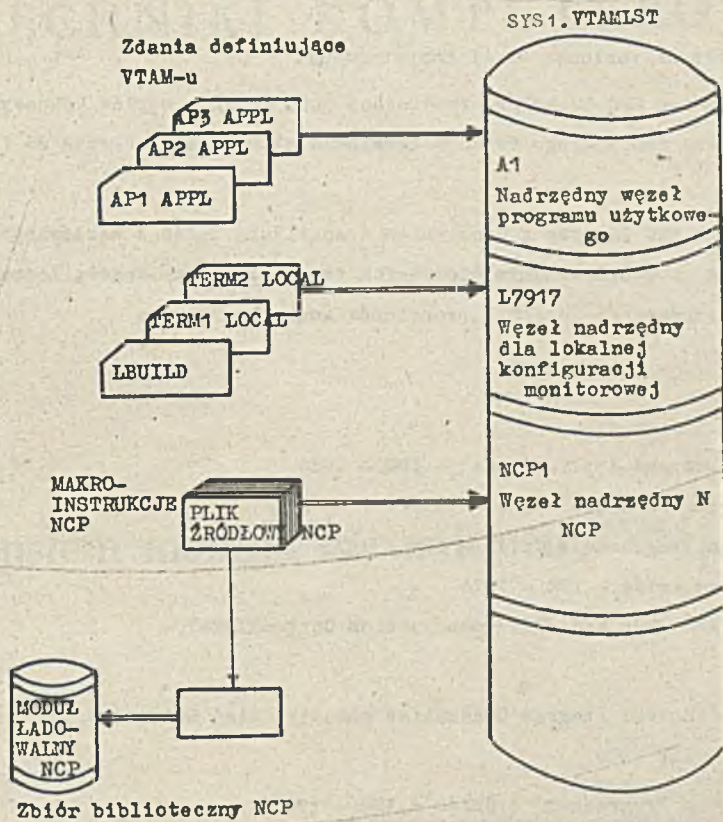
Praca we VTAM-ie



Rys.5. Nadrzędne i podrzędne węzły we VTAM-owskiej sieci telekomunikacyjnej

Pracę VTAM-u inicjuje się przez uruchomienie procedury zawierającej główny moduł VTAM-u ISTINAØ1. Procedurę tę należy uprzednio wprowadzić do zbioru SYS1. PROCLIB. Musi ona oprócz zdania EXEC wywołującego ISTINAØ1, zawierać zdanie DD, określające zbiory: SYS1. VTAMLIB. SYS1. VTAMLST, SYS1. VTAMOBJ i zbiór biblioteczny zawierający ładowny moduł NCP. W trakcie inicjowania VTAM-u następuje wybór konfiguracji, na której ma pracować VTAM, tzn. listy węzłów nadrzędnych i wybór odpowiedniej dla danej sesji listy parametrów startowych.

Warunkiem przesyłania komunikatów między terminalem a programem użytkowym jest dokonanie operacji połączenia (logon) terminala z programem użytkowym. Może ona być realizowana przez terminal, VTAM (tzn. automatyczny logon), operatora sieci lub VTAM-owski program użytkowy symulujący połączenie terminala z programem. Gdy jakiegokolwiek żądanie połączenia terminala z danym programem dotrze do VTAM-u, VTAM zawiadamia o tym dany program i jeżeli program zaakceptuje to żądanie, VTAM realizuje połączenie. Od tej chwili komunikaty do i z tego terminala może obsługiwać tylko dany program użytkowy. Kiedy terminal zostanie odłączony od danego programu, może korzystać z niego inny program użytkowy.



Rys.6. Tworzenie plików definiujących węzły nadrzędne

### Wnioski

Stworzenie w IMM eksperymentalnej sieci terminalowej i oprogramowanie jej według koncepcji przedstawionej w opracowaniu pozwoliło na wyodrębnienie wniosków dotyczących perspektyw rozwoju sieci wykorzystujących bazę JS w naszym kraju.

- W zakresie sprzętu uzyskano potwierdzenie tezy, że dostępne środki programowe wykorzystane w ramach opisywanej pracy pozwalają na oprogramowywanie sieci złożonych całkowicie ze sprzętu JS, a z chwilą przekazania do eksploatacji nowego polskiego komputera z pamięcią wirtualną R-34 - złożonych całkowicie ze sprzętu produkcji krajowej.
- Przeprowadzenie badań pozwoliło na ustawienie się na poziomie wyjściowym do prowadzenia prac perspektywicznych mających na celu zoptymalizowanie pracy sieci. Może to nastąpić np. przez wykorzystanie udoskonalonych protokołów. Jednym z nich jest protokół ramowy SDLC. Szanse na jego wykorzystanie zwiększa perspektywa skierowania do produkcji monitora ekranowego JS 7978 wraz z jednostką sterującą JS 7974 - przeznaczonego do pracy wg protokołu SDLC.
- Wynik omawianej pracy można traktować jako krok naprzód na drodze do tworzenia sieci niejednorodnych tj. łączenie maszyn JS i SM. Łączenie to mogłoby nastąpić bądź na poziomie sieci

terminalowej, bądź też na poziomie sieci komputerowej.

W pierwszym wypadku EMC SM byłaby traktowana jako jeden z węzłów końcowych przez uruchomienie na niej programu emulującego funkcje terminala akceptowanego przez JS (np. terminala IBM 2780).

W drugim wypadku EMC JS wraz z otoczeniem stanowiłaby jeden z węzłów sieci niejednorodnej, natomiast EMC SM wraz ze swoim własnym otoczeniem tworzyłaby inny węzeł. Łączność między węzłami mogłaby być organizowana na poziomie procesorów komunikacyjnych.

#### Bibliografia

- [1] OS/VS TCAM Concepts and Applications - IBM - 1976
- [2] OS/VS1 TCAM Programmers Guide - IBM - 1976
- [3] OS/VS1 VTAM System Programmer's Guide - IBM - 1975
- [4] VTAM Macro Language Guide - IBM - 1974
- [5] "Introduction to IBM 3704 and 3705 Communication Controllers" - IBM - 1978
- [6] IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual - IBM 1976
- [7] OS/VS1 TCAM10 System Programmer's Guide - IBM - 1980
- [8] Butrimienko A.W.: Projektowanie i eksploatacja sieci komputerowych. Warszawa WNT 1983
- [9] Niemczycki L.: Oprogramowanie teleprzetwarzania maszyn Jednolitego Systemu. Warszawa WNT 1979
- [10] Informator dla użytkowników komputerów Mera-Elwro. Wrocław 1982

mgr Elżbieta ROSZKOWSKA

Instytut Maszyn Matematycznych

## System maszyn wirtualnych - VM/JS-P

Niniejsze opracowanie omawia system maszyn wirtualnych VM/JS-P. Pierwszy rozdział zawiera ogólne informacje o całym systemie i możliwościach jego stosowania. Rozdziały następne są opisem poszczególnych składowych tego systemu, tzn.

- programu sterującego - CP,
- monitorowego systemu konwersacyjnego - CMS,
- systemu RSCS,
- i systemu IPCS.

### Wprowadzenie do systemu VM/JS-P

System maszyn wirtualnych VM/JS-P jest funkcjonalnym odpowiednikiem systemu operacyjnego VM IBM/370. Zarządza on pracą rzeczywistego zestawu komputerowego opartego na EMC serii RIAD drugiej generacji. Wspomiane maszyny rzeczywiste muszą działać w trybie RC, tzn. mieć zapewnioną sprzętowo opoję dynamicznej translacji adresów - DAT. Podstawową częścią tego systemu jest program sterujący CP, który zarządza zasobami maszyny rzeczywistej. Zarządzanie to odbywa się w taki sposób, aby udostępnić każdemu użytkownikowi tego systemu niezależny system komputerowy (maszynę wirtualną) z określonymi wcześniej zasobami. Oznacza to, że maszyna wirtualna symuluje, zarówno programowe, jak i sprzętowe zasoby pewnej maszyny rzeczywistej. Symulowane zasoby mogą być wykorzystane jednocześnie przez kilka maszyn wirtualnych lub przydzielane danej maszynie wirtualnej na określony odcinek czasu. Każdą z maszyn wirtualnych może zarządzać w danym momencie ten sam lub inny system operacyjny, a poszczególni użytkownicy mogą przystosować swoje maszyny wirtualne do własnych potrzeb.

### Maszyna wirtualna

Maszyna wirtualna jest pojęciem logicznym i oznacza powiązanie w jedną całość wybranych zasobów sprzętowych i programowych (lub ich symulowanych odpowiedników) maszyny rzeczywistej.

Wynika z tego, że maszyna wirtualna jest funkcjonalnym odpowiednikiem rzeczywistego systemu komputerowego, tak więc wykonanie programu w maszynie rzeczywistej i wirtualnej daje ten sam wynik. Informacje dotyczące zestawu maszyny wirtualnej (specyfikacja jej zasobów) zawarte są w odpowiedniej pozycji skrowidza systemu.

#### Skorowidz systemu

Skorowidz systemu VM/JS-P jest plikiem zawierającym pozycje dla wszystkich potencjalnych maszyn wirtualnych. Pozycje te zawierają zdania kontrolne, które definiują każdą maszynę wirtualną. Pozycja skrowidza zawiera takie informacje:

- konfiguracja maszyny,
- tryb pracy (EC lub EC),
- identyfikator użytkownika,
- hasło,
- wskaźnik klasy komend CP dostępnych dla tego użytkownika.

```
USER VIRT1 VIRT1 512K 1M G
ACCOUNT S5 SYSPRG
CONSOLE 009 J215
SPOOL OOC 2540 READER A
SPOOL OOD 2540 PUNCH A
SPOOL OOE 1403 A
MDISK 191 3330 001 001 CPR6L1 R
LINK MAINT 194 194 PR
```

Rys. 1. Skorowidz systemu VM/JS-P

```
VM/JSP ONLINE
logon user 1
ENTER PASSWORD:
LOGON AT 11:03:18 ON WEDNESDAY 11/29/84
ipl ems
```

Rys. 2. Inicjowanie pracy maszyny wirtualnej

#### Inicjowanie pracy maszyny wirtualnej

W celu podłączenia się do systemu VM/JS-P, to znaczy uruchomienia odpowiedniej maszyny wirtualnej, należy z terminala, spełniającego rolę konsoli maszyny wirtualnej, wprowadzić komendę LOGON specyfikując swój identyfikator oraz hasło. Identyfikator służy do znalezienia właściwej pozycji skrowidza, natomiast hasło pozwala na kontrolę, czy działający użytkownik jest uprawniony do korzystania z danej maszyny wirtualnej.

W momencie zainicjowania przez użytkownika pracy wskazanej maszyny wirtualnej, informacje pozwalające na uruchomienie tej maszyny są brane przez CP ze skrowidza.

#### Systemy działające w poszczególnych maszynach wirtualnych

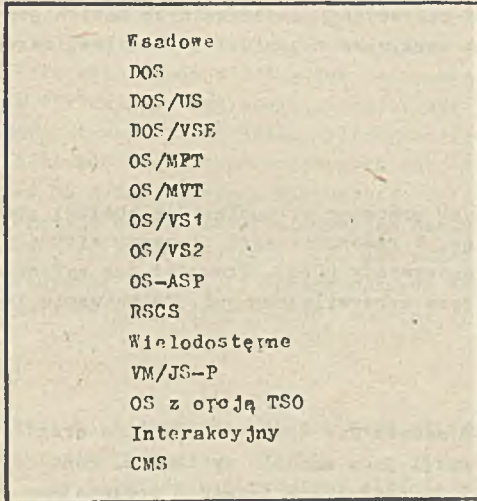
CP zarządza podziałem zasobów rzeczywistych między działające jednocześnie maszyny wirtualne, natomiast za sterowanie pracą poszczególnych maszyn odpowiedzialne są działające w nich systemy operacyjne. Ponieważ każda z maszyn wirtualnych działa niezależnie od pozostałych, pracę ich mogą kontrolować różne systemy operacyjne lub różne wersje tego samego systemu.

Rys. 3 przedstawia wykaz podstawowych systemów operacyjnych, które mogą być wykorzystane do sterowania pracą maszyny wirtualnej.

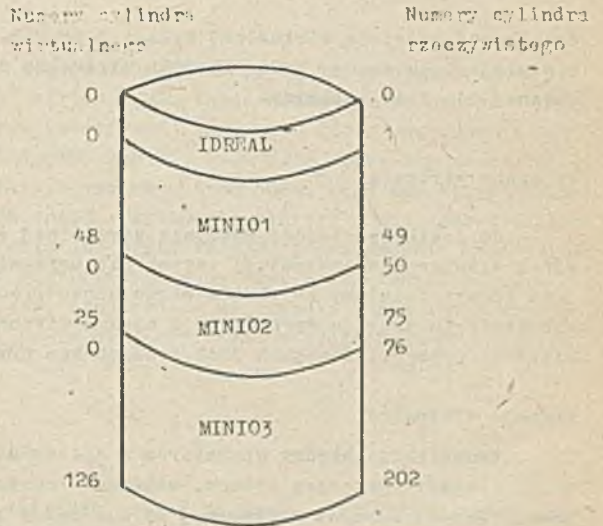
Wśród tych systemów wyróżniamy systemy interakcyjne oraz wsadowe. Systemy interakcyjne są systemami jednego użytkownika, natomiast wsadowe mogą być systemami wielodostępnymi.

Systemy jednego użytkownika to takie, które mogą być wykorzystane w sposób konwersacyjny tylko przez jednego użytkownika w danym momencie. Przykładem takiego systemu jest monitorowy system konwersacyjny /CMS/, opracowany w celu uruchamiania i testowania produktów programowych.





Rys. 3. Systemy działające w maszynach wirtualnych



Rys. 4. Dyski wirtualne

Systemy wielodostępane, takie jak OS/JS-P z opcją TSO, kontrolują pracę pojedynczej maszyny wirtualnej i jednocześnie zapewniają obsługę wielu terminali konwersacyjnych. Użytkownik maszyny wirtualnej może wykorzystać inne terminale niż podstawowy (konsola maszyny wirtualnej) przez wydanie z danego terminala (komendy) DIAL, wskazując identyfikator maszyny wirtualnej. Ponieważ taki terminal nie jest kontrolowany przez CP, lecz przez system operacyjny działający w danej maszynie, nie można wprowadzać z niego komend CP. Każdy system operacyjny może być wprowadzony do maszyny wirtualnej przez:

- wskazanie adresu urządzenia, na którym on rezyduje bądź,
- przez wyspecyfikowanie jego nazwy, jeżeli system taki został wcześniej przechowany i nadano mu nazwę.

#### Zasoby maszyny wirtualnej

Zasoby maszyny wirtualnej można podzielić na takie, które mają swoje bezpośrednie rzeczywiste odpowiedniki (taśmy magnetyczne) oraz takie, które takich odpowiedników nie mają. Zasoby, które występują w konfiguracji maszyny rzeczywistej, lecz nie są na stałe przypisane żadnej maszynie wirtualnej, są kontrolowane przez CP. Zasoby rzeczywiste przypisane na stałe danej maszynie wirtualnej są przez tę maszynę kontrolowane.

#### Pamięć wirtualna

System maszyn wirtualnych może symulować niezależne obszary pamięci operacyjnej o wielkości podzielnej przez 4K w granicach od 8K do 16M. Wielkość pamięci operacyjnej jest definiowana w skorowidzu systemu oddzielnie dla każdej maszyny wirtualnej. Może ona przewyższać wielkość pamięci rzeczywistej i nie jest tą wielkością limitowana w żaden istotny sposób. Symulowana pamięć operacyjna (pamięć wirtualna) jest podzielona na segmenty i strony, a zarządzanie tą pamięcią oraz jej ochrona odbywa się przez obsługę segmentów i stron. Wielkość tych jednostek dla poszczególnych obszarów pamięci, z których każdy związany jest z konkretną maszyną wirtualną, jest stała i wynosi odpowiednio 64K i 4K. Z tego wynika, że dopuszczalna liczba segmentów przy-

dzielnicy maszynie wirtualnej wynosi 1 do 256. Pamięć operacyjna poszczególnych maszyn jest niezależna, to znaczy jedna maszyna wirtualna nie może korzystać z pamięci operacyjnej zarezerwowanej dla innej maszyny.

#### Procesor wirtualny

CP dostarcza każdej maszynie wirtualnej pojedynczy procesor wirtualny dla obsługi przerw i wykonywania instrukcji innych niż uprzywilejowana. W rzeczywistości procesor wirtualny jest to przydzielony na pewien okres czasu procesor rzeczywisty (CPU). Procesor ten wykonuje wszystkie instrukcje wychodzące z maszyn wirtualnych poza uprzywilejowanymi. Wykonywanie instrukcji uprzywilejowanych jest realizowane przez CP.

#### Konsole wirtualne

Komunikacja między operatorem a systemem VM/JS-P odbywa się przez rzeczywiste urządzenie we/wy obsługiwane przez system, wskazane podczas generacji jako konsola systemowa. Ponadto użytkownik każdej maszyny wirtualnej ma dla celów komunikacji przypisany terminal spełniający rolę konsoli. Dopuszcza się, aby jeden terminal rzeczywisty spełniał rolę konsoli więcej niż jednej maszynie wirtualnej (komenda DISCONN).

#### Urządzenia wirtualne

Maszyna wirtualna obsługuje takie same typy urządzeń jak maszyna rzeczywista. Ma też możliwość kontrolowania ich pracy. Konfiguracja maszyny musi być zdefiniowana w pozycji skorowidza, ale dodatkowe urządzenia we/wy mogą być definiowane dynamicznie przez wydanie odpowiednich komend obsługiwanych przez CP. Użytkownik maszyny wirtualnej może wykorzystywać te adresy, które są przypisane urządzeniom rzeczywistym lub przyporządkować inne - wirtualne, a wtedy CP dokona konwersji adresów wirtualnych na adresy rzeczywiste oraz zapewni transmisję wszystkich potrzeb danych. Możliwe jest wykorzystanie tych samych adresów wirtualnych przez użytkowników różnych maszyn wirtualnych. VM/JS-P uwzględnia kilka sposobów komunikacji między urządzeniem wirtualnym a rzeczywistym i z tego względu dzieli urządzenia przypisane maszynom wirtualnym na cztery grupy:

- urządzenia dedykowane,
- wirtualne dyski,
- wirtualne podstawowe urządzenia we/wy,
- wirtualne procesory telekomunikacyjne (EC 8371).

#### Urządzenia dedykowane

Urządzenia rzeczywiste przypisane bezpośrednio jakiejś maszynie wirtualnej są uważane za urządzenia dedykowane. Urządzenie dedykowane danej maszynie wirtualnej jest całkowicie przez nią kontrolowane. Dedykowanie urządzenia może nastąpić przez:

- użycie odpowiedniej komendy CP (ATTACH),
- przypisanie go, w skorowidzu, danej maszynie wirtualnej.

Operacja dedykowania poprawia przepustowość maszyny wirtualnej, ale zwiększa wymagania na zasoby rzeczywiste. Urządzeniami, które zawsze muszą być dedykowane są przewijaki taśmy magnetycznej, ponadto użytkownik może dedykować dla maszyny, w której działa urządzenie:

- o bezpośrednim dostępie,
- terminale,
- podstawowe urządzenia we/wy.

## Dyski wirtualne

Jeden rzeczywisty pakiet dyskowy może być logicznie podzielony na kilka niezależnych części, związanych z jedną lub kilkoma maszynami wirtualnymi. Każda z omawianych części tworzy dysk wirtualny, zwany minidyskiem. Dyski tego typu (minidyski) mogą mieć różną pojemność i być wykorzystywane przez różne systemy operacyjne CMS, DOS lub OS. Na rys. 4 pokazany jest przykład trzech minidysków założonych na jednym pakiecie dyskowym typu 5061. Mogą one wszystkie należeć do jednej maszyny wirtualnej, lub każdy do innej. Użytkownik minidysku może zabezpieczyć informacje na nim zawarte używając hasła do pisania lub czytania albo przez nadanie minidyskowi statusu tylko do czytania. CP, aby zabezpieczyć się przed jednoczesnym korzystaniem z dysku (o statusie do pisania) przez różnych użytkowników, standardowo udostępnia w tym samym czasie pisania tylko jednemu z nich, chociaż użytkownik może wyspecyfikować wielodostępność, jeżeli tego chce.

## Wirtualne podstawowe urządzenia wejścia/wyjścia

Do podstawowych urządzeń wejścia/wyjścia zaliczamy czytniki kart, perforatory kart oraz drukarki. Zazwyczaj rzeczywiste podstawowe urządzenie nie jest zajęte permanentnie. Dlatego dane wejściowe lub wyjściowe, związane z kilkoma wirtualnymi podstawowymi urządzeniami we/wy, mogą być obsługiwane przez jedno rzeczywiste podstawowe urządzenie we/wy. Podczas definiowania konfiguracji swojej maszyny wirtualnej użytkownik może wyspecyfikować tyle podstawowych urządzeń we/wy, ile uważa za potrzebne. CP kontroluje pracę urządzeń wirtualnych (podobnie jak rzeczywistych), a do ich obsługi wykorzystuje możliwość przechowywania danych na urządzeniu o bezpośrednim dostępie (operacja SPOOL). Następnie CP przesyła przechowywane:

- dane wejściowe do czytnika odpowiedniej maszyny wirtualnej,
- dane wyjściowe z drukarki wirtualnej lub perforatora kierowane są na odpowiednie urządzenia rzeczywiste.

## Procesor telekomunikacyjny

Konfiguracja maszyny wirtualnej może zawierać procesor telekomunikacyjny (odpowiednik EC 8071). Rzeczywisty procesor telekomunikacyjny nie może być w tym samym czasie dedykowany kilku maszynom wirtualnym. Natomiast może być używany do symulowania kilku wirtualnych procesorów telekomunikacyjnych. Symulowanie kilku wirtualnych procesorów telekomunikacyjnych polega na tym, że rzeczywisty procesor telekomunikacyjny jest dzielony między różne maszyny wirtualne lub wykorzystywany przez nie przemiennie w krótkich odcinkach czasu.

## Adapter kanał - kanał

Użytkownik systemu maszyn wirtualnych może włączyć do konfiguracji swojej maszyny adapter kanał-kanał (AKE) niezależnie od tego czy istnieje, czy też nie jego rzeczywisty odpowiednik w zestawie wykorzystywanego systemu komputerowego. Gdy rzeczywisty adapter jest przypisany do danej maszyny wirtualnej, wówczas może ona komunikować się z innym rzeczywistym systemem komputerowym niż ten, na którym została zainicjowana jej praca.

Jeżeli takiego rzeczywistego odpowiednika nie ma wśród zasobów danej maszyny wirtualnej, wówczas może ona komunikować się jedynie z maszynami wirtualnymi działającymi w ramach tego samego co ona zestawu komputerowego.

Inna możliwość komunikowania się między sobą maszyn wirtualnych jednego zestawu komputerowego została opisana w punkcie następnym.

## Komunikacja pomiędzy maszynami wirtualnymi

Najprostszy sposób komunikacji pomiędzy maszynami wirtualnymi jest wykorzystanie opoju VMCP. Opoja VMCP pozwala jednej maszynie wirtualnej (nadawca) komunikować się z inną (odbiorca) działającą pod kontrolą jednego VM/JS-P. Komunikacja ta polega na bezpośrednim przekazywaniu instrukcji i danych z pamięci operacyjnej nadawcy do pamięci odbiorcy.

## Charakterystyka systemu VM/JS-P

System ten przeznaczony jest do zarządzania pracą dużych zestawów komputerowych, które są wykorzystywane jednocześnie przez wielu użytkowników o zróżnicowanych wymaganiach. Pogodzenie tych wymagań jest realizowane przez udostępnienie poszczególnym użytkownikom oddzielnych maszyn wirtualnych z możliwością działania w każdej z tych maszyn systemu najodpowiedniejszego do typu wykonywanej pracy. Tak więc dla testowania i uruchomienia programów został specjalnie opracowany system CMS - (zob. pkt "Monitorowy system konwersacyjny - CMS"). Do zdalnego wprowadzania pracy z możliwością zwrotnego otrzymania wyników służy system RSCS - (zob. pkt "System RSCS"). Użytkownik maszyny wirtualnej ma także możliwość pracy w każdym z dopuszczalnych dla danego zestawu komputerowego wsadowych systemów operacyjnych. Ponadto pewne opoje systemu VM/JS-P stwarzają dodatkowe możliwości efektywniejszego wykorzystania wszystkich możliwości sprzętowych.

## PROGRAM STERUJĄCY - CP

CP (podstawowa część systemu VM/JS-P) buduje maszyny wirtualne oraz kontroluje ich dostęp do poszczególnych zasobów maszyny rzeczywistej. Budowanie maszyny wirtualnej ma miejsce, gdy użytkownik podłączy się do systemu VM/JS-P. Wówczas to CP wykonuje poniższe czynności.

- Buduje, opierając się na informacjach przekazywanych w skorowidzu, odpowiednie bloki kontrolne opisujące parametry i zasoby maszyny wirtualnej.
- Umieszcza daną maszynę w kolejce maszyn oczekujących na dostęp do dzielonych zasobów rzeczywistych.

Kontrola pracy maszyn wirtualnych polega na tym, że program sterujący zarządza zasobami maszyny rzeczywistej przydzielając je przemiennie wielu maszynom wirtualnym i kontrolując ich wykorzystanie. Tak więc CP przechwytuje, transluje i zarządza wykonaniem wszystkich rzeczywistych operacji we/wy w maszynach wirtualnych. Jedynie CP działa w stanie "system", natomiast wszystkie maszyny wirtualne działają w stanie "problem", dlatego program sterujący przechwytuje i realizuje dodatkowo wydawane w maszynie wirtualnej przerwania i instrukcje uprzywilejowane.

## Zarządzanie czasem w maszynie wirtualnej

Chociaż użytkownikowi maszyny wirtualnej wydaje się, że dysponuje oddzielnym procesorem, który wykonuje wskazane instrukcje, to jednak w rzeczywistości na rzecz wszystkich użytkowników pracuje tylko jeden procesor rzeczywisty, tj. CPU. VM/JS-P używa techniki odinkowania czasu dla przydzielenia tego procesora poszczególnym maszynom, w efekcie więc jeden procesor rzeczywisty

zastępuje działanie wielu procesorów wirtualnych. Każda maszyna wirtualna okresowo otrzymuje dostęp do procesora rzeczywistego na krótki okres czasu, zwany odcinkiem czasowym. CP wyznacza, jak często i na jak długo maszyna wirtualna otrzymuje dostęp do procesora przez zakwalifikowanie jej do grupy maszyn wadowych lub konwersacyjnych. Decyzja o przynależności do grupy podejmowana jest na podstawie komend wysyłanych z konsoli głównej oraz przerw przebiegających z terminala maszyny wirtualnej działającej przez ostatnio wyznaczony odcinek czasu.

Gdy liczba wspomnianych operacji we/wy przekracza pewną wyznaczoną granicę CP określa maszynę wirtualną jako konwersacyjną, i wyznacza dla niej krótszy z dwóch możliwych odcinków czasu. Natomiast, gdy liczba operacji jest mniejsza od wspomnianej granicy, maszyna zaliczana jest do wadowych i CP wyznacza dla niej dłuższy z możliwych odcinków czasu. Maszyny konwersacyjne są obsługiwane w pierwszej kolejności, natomiast wadowe dopiero wtedy, gdy żadna z maszyn konwersacyjnych nie może uzyskać dostępu do CPU. CP przydziela maszynie wirtualnej dostęp do procesora tylko wtedy, kiedy nie czeka ona na inne zasoby lub akcje takie, jak:

- sprowadzenie strony pamięci z urządzenia o bezpośrednim dostępie do pamięci rzeczywistej,
- przetłumaczenie operacji we/wy, jej rozpoczęcie lub zakończenie,
- wykonanie komendy CP.

#### Zarządzanie pamięcią maszyny wirtualnej

Każda maszyna wirtualna ma związaną z nią pamięć operacyjną, której wielkość jest definiowana w skorowidzu. Pozycja w skorowidzu zawiera dwie wielkości określające pamięć maszyny wirtualnej, tj. jej wielkość normalną i maksymalną. Pamięć operacyjna maszyny wirtualnej może być mniejsza lub większa niż pamięć maszyny rzeczywistej. Normalna wielkość pamięci nie może jednak być mniejsza niż 8K bajtów, natomiast maksymalna nie może przekraczać 16M bajtów. Każda z tych dwu wielkości musi być wielokrotnością 4K. W momencie inicjowania pracy pamięć maszyny wirtualnej przyjmuje wielkość zdefiniowaną jako normalną. Użytkownik może na pewien okres czasu predefiniować wielkość dostępnej dla siebie pamięci nadając jej dowolną wartość, podzieloną przez 4K, ale nie większą niż maksymalna. System operacyjny kontrolujący pracę maszyny wirtualnej działa tak, jak gdyby miał do dyspozycji pamięć rzeczywistą, chociaż pamięć operacyjna każdej maszyny wirtualnej jest tworzona i kontrolowana przez CP, jako pamięć wirtualna. System VM/37-P zarządza więc jednocześnie kilkoma niezależnymi obszarami pamięci wirtualnej - jednym dla każdej maszyny wirtualnej. Pamięć maszyny wirtualnej jest logicznie dzielona na 64Kbajtowe części, zwane segmentami. Segmenty są dalej dzielone na strony o wielkości 4K. Dla każdej maszyny wirtualnej, CP tworzy i aktualizuje zbiór tablic segmentów i stron pamięci wirtualnej oraz tablice opisujące alokację stron w pamięci rzeczywistej. Omawiane tablice są używane przez opcję DAT podczas wykonania operacji określania adresu pamięci rzeczywistej dla odpowiedniego adresu pamięci wirtualnej.

#### Stronicowanie

Strony pamięci wirtualnej przechowywane są na urządzeniu o bezpośrednim dostępie w stronicowanym obszarze PAGE. Sprowadzanie stron jest dokonywane na żądanie CP wynikłe z zapotrzebowania maszyny wirtualnej. Oznacza to, że strona pamięci wirtualnej nie jest pobierana z obszaru stronicowanego do pamięci rzeczywistej, dopóki nie jest potrzebna do działania maszyny wirtualnej. Pamięć maszyny rzeczywistej jest fizycznie i logicznie dzielona na obszary o wielkości 4K, zwane ramkami.

Strona pamięci wirtualnej pobierana do pamięci rzeczywistej umieszczona jest w wolnej ramce pamięci rzeczywistej. Ramka uważana jest za wolną pod warunkiem, że:

- żadna strona pamięci wirtualnej nie była umieszczona w danej ramce,
- CP, na podstawie specjalnego algorytmu opartego na kontroli liczby odwołań do danej strony, podjęło decyzję o jej zwolnieniu (strona nieaktywna).

Zwolnienie ramki pamięci musi być poprzedzone:

- przeniesieniem jej zawartości do zbioru stronicowanego (w pierwsze wolne miejsce),
- uaktualnieniem informacji w tablicach segmentów i stron pamięci wirtualnej.

Jeśli zawartość nieaktywnej strony nie została zmieniona podczas działania maszyny wirtualnej, alokacja jej w obszarze stronicowanym nie jest zmieniana (nie ma potrzeby odnawiania jej zawartości). Pewna część systemu VM/JS-P jest zapisana na stronach zaznaczonych jako często wykorzystywane i na stałe przechowywana w pamięci rzeczywistej. Skracca to znacznie czas dostępu do strony, jednak dla zoptymalizowania wykorzystania pamięci rzeczywistej tylko niewielką liczbę stron należy przechowywać na stałe w pamięci rzeczywistej.

Zarządzanie operacjami wejścia/wyjścia w maszynie wirtualnej

System operacyjny działający w maszynie wirtualnej jest zdolny do wykonywania operacji wejścia/wyjścia na wszystkich urządzeniach wirtualnych z nim związanych. Urządzenia wirtualne mogą być definiowane w skorowidzu dla danej maszyny wirtualnej lub mogą być dynamicznie dołączone do (odłączone od) maszyny wirtualnej w trakcie jej pracy. Ze względu na sposób dostępu wśród urządzeń wirtualnych wyróżniamy:

- dedykowane, gdy odpowiada mu konkretne urządzenie rzeczywiste,
- dzielone, jeżeli urządzenie wirtualne jest minidyskiem dostępnym dla więcej niż jednej maszyny wirtualnej jednocześnie,
- SPOOL, gdy dane wejściowe lub wyjściowe dla urządzenia wirtualnego są przechowywane przez CP tymczasowo na urządzeniu o bezpośrednim dostępie.

Gdy wsadowy system operacyjny zarządza pracą maszyny rzeczywistej, operacje we/wy są inicjowane w momencie sygnalizowania przez program użytkowy potrzeby wysłania instrukcji START I/O dla wskazanego urządzenia. Błędy urządzenia są obsługiwane przez dany system operacyjny. W maszynie wirtualnej system wsadowy może wykonywać te same funkcje, lecz specyfikowany adres urządzenia i miejsce pamięci będącej początkiem obszaru przesyłania danych są wirtualne. Ponadto maszyna wirtualna działa tylko w wirtualnym (nie rzeczywistym) stanie "system". Dlatego, po wysłaniu instrukcji START I/O przez system operacyjny maszyny wirtualnej, kontrola przekezywana jest do programu sterującego. CP dokonuje translacji adresów wirtualnych na rzeczywiste oraz kopiuje do swojego własnego pola (obszaru roboczego) komendy kanałowe specyfikowane przez system operacyjny i przenosi do pamięci rzeczywistej wszystkie strony wirtualne potrzebne do przesyłania danych. Wspecyfikowane strony są na stałe w pamięci, dopóki nie będzie zakończona operacja we/wy. Gdy pojedyncze słowo komend kanałowych specyfikuje obszar pamięci rozciągający się na wiele stron pamięci wirtualnej, CP generuje program kanałowy, używając takiego adresowania danych kanałowych, które umożliwi wykorzystanie nieciągłych ramek pamięci. Jeżeli wirtualnym urządzeniem jest minidysk, dokonywana jest również modyfikacja wszystkich specyfikowanych numerów cylindrów, aby określić właściwą lokalizację danych. Zawsze program sterujący przekształca adres urządzenia wirtualnego na adres rzeczywisty i kontroluje wykonanie operacji we/wy. Podczas tego procesu maszyna wirtualna zaznaczona jest jako nie gotowa do normalnej pracy. Jednocześnie z ponownym przekazaniem kontroli, maszynie wirtualnej zwracany jest specyficzny kod powrotu (tak, jak w maszynie rzeczywistej) wskazujący stan operacji we/wy dla późniejszej ich interpretacji i przetwarzania. Gdy zdarzy się błąd we/wy, CP notuje wszystkie niezbędne informacje, natomiast samo inicjowanie programów obsługi błędów pozostawione jest systemowi operacyjnemu w maszynie wirtualnej. Te prace, które są wykonywane w maszynie wirtualnej (nie dotyczy to maszyny V=R) nie mogą zawierać dynamicznie modyfikowanych programów kanałowych.

Komunikacja z użytkiem DIAGNOSE

Realizacja przez CP wirtualnych operacji we/wy może być uproszczona, jeżeli maszyna

wirtualna używa do tej komunikacji specjalnej postaci instrukcji DIAGNOSE. Przykładem tak zbudowanego systemu jest CMS napisany specjalnie jako komponent VM/JS-P. Używa on tego typu komunikacji zamiast zwykłych instrukcji START I/O. W omawianej sytuacji CP realizuje całkowicie obsługę błędów we/wy. Operacje we/wy inicjowane przez CP dla własnych potrzeb, np. stronicowanie i składowanie, są przedsięwzięte bezpośrednio i nie jest wykonywany proces translacji opisany poprzednio.

#### Operacja SPOOL

Operacja SPOOL realizowana przez CP, pozwala kilku maszynom wirtualnym korzystać w tym samym czasie z jednego rzeczywistego urządzenia podstawowego. Pojedyncza maszyna wirtualna ma najczęściej niewielkie wymagania od urządzeń podstawowych, dlatego dzielenie urządzenia rzeczywistego jest standardem operacji systemowych. Nie maszyna wirtualna, lecz CP kontroluje urządzenia podstawowe, które są zaznaczone w pozycji skorowidza jako SPOOL. Kiedy maszyna wirtualna inicjuje operacje we/wy dla urządzenia podstawowego SPOOL, CP przechwytuje związane z tą operacją instrukcje i odpowiednio je modyfikuje.

Dane przenoszone są do rekordów o wielkości strony-(bloki 4096-bajtów) tworzonych na dysku w obszarze SPOOL. Dysk ten służy jako pamięć pośrednia między rzeczywistym urządzeniem podstawowym, a maszyną wirtualną. Jeżeli miejsce na urządzeniu o bezpośrednim dostępie przeznaczonym na obszar SPOOL zostało w całości wykorzystane, operacja SPOOL jest wstrzymywana i urządzenie wirtualne wprowadzane jest w stan "niegotów". Na operację składowania należy wówczas wygospodarować dodatkowe wolne miejsca. Można to osiągnąć usuwając któryś ze zbiorów typu SPOOL lub rezerwując dodatkowe miejsca na urządzeniu o bezpośrednim dostępie.

#### Monitorowy system konwersacyjny - CMS

Monitorowy system konwersacyjny (CMS) jest podstawową częścią systemu VM/JS-P. Wspólnie z programem kontrolnym CP zarządza on podziałem czasu bezpośredniej realizacji prac (obsługa komend) oraz wykonywania programów. Użytkownik tak zarządzanej maszyny wirtualnej może tworzyć, jak również modyfikować, translować i wykonywać programy.

CMS jest systemem operacyjnym, który może działać tylko pod kontrolą systemu VM, to znaczy zarządzać pracą maszyny wirtualnej, a nie rzeczywistej. Wiąże się to z tym, że CMS wykorzystuje instrukcję DIAGNOSE dla obsługi operacji wejścia/wyjścia związanych z urządzeniami bezpośrednio dostępu i taśmą magnetyczną.

#### Konfiguracja CMS

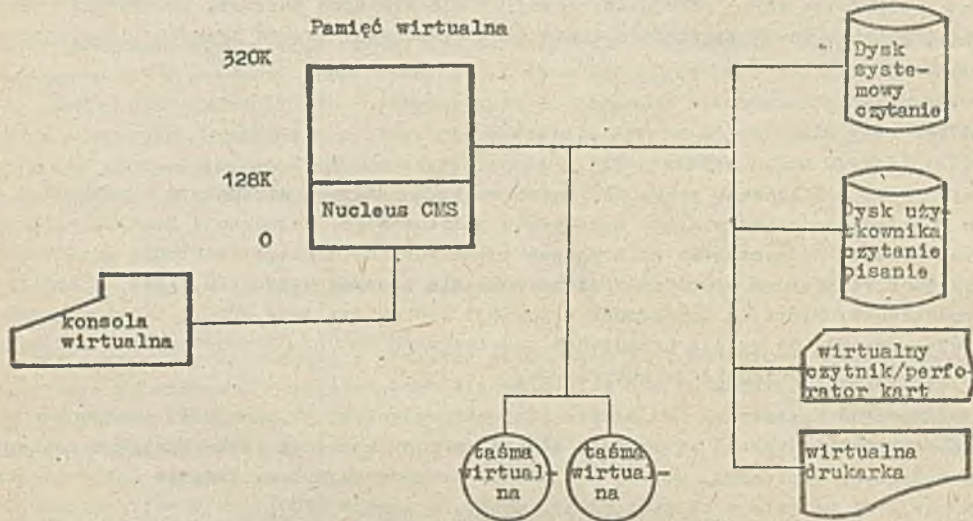
Maszyna wirtualna przeznaczona do współpracy z systemem CMS musi być wyposażona podobnie, jak każda inna maszyna wirtualna w odpowiedni zestaw urządzeń wirtualnych z wymaganiem na pamięć operacyjną 32K, z których 128K zajmuje jądro CMS.

Programy użytkownika wykonywane pod kontrolą CMS mogą zwiększać te wymagania, ale dostępna pamięć nie może przekraczać 16M i musi być wielokrotnością 4K.

#### Tworzenie i wykonanie programu

CMS dostarcza użytkownikowi różnych środków związanych z uruchomieniem i wykonaniem programu. Środki te pozwalają:

- na tworzenie programu źródłowego oraz jego kompilację,
- budowanie plików dla celów testowania,
- wykonywanie i testowanie programów,
- szukanie błędów programu w sposób konwersacyjny.



Rys. 5. Konfiguracja maszyny CMS

W momencie wywołania kompilatora, CMS dynamicznie alokuje potrzebne dla jego działania pliki robocze na jakimkolwiek dostępnym dysku użytkownika, który ma dostatecznie dużo miejsca, a zwalnia je po zakończeniu pracy, jeżeli położenie plików roboczych nie było wyspecyfikowane przez użytkownika. Plik wyjściowy produkowany przez translator (postać binarna programu) oraz plik listingu są normalnie alokowane na tym samym dysku, co wejściowy plik zdań źródłowych lub na dysku systemowym.

Plik wyjściowy identyfikowany jest przez nazwę pliku wejściowego oraz typ pliku odpowiednio TEXT lub LISTING. Translatory wykonywane pod CMS-em wołane są przez nazwę, a dane wejściowe stanowią plik źródłowy, którego identyfikator typu wskazuje na translator. Wywołują odpowiednią komendę, użytkownik może wyspecyfikować opcje CMS, jak również opcje translatora, które są identyczne z kodowanymi w karcie EXEC dla systemu OS lub VS1. Użytkownik COBOL-u specyfikuje opcje translatora w komendzie OPTION, która poprzedza komendę COBOL. Użytkownik PL/1 musi specyfikować opcje translatora w zdaniu PROCESS, która jest pierwszym zdaniem w programie PL/1.

W CMS symulowany jest program łączący odpowiednik tego, który działa w systemie DOS. Pliki, które mają być łączone, mogą być czytane z biblioteki utworzonej w systemie DOS. Moduł wynikowy umieszczony jest w pliku CMS-owskim o nazwie DOSLIB.

#### Translatory

Assembler VM/JS-P jest dystrybuowany jako część systemu. Wszystkie makrorozkazy potrzebne do jego instalacji rozprowadzania są standardowo dostarczane w bibliotece CMS. Ponadto translatory innych języków programowania są dostępne w CMS po ich dodatkowym zainstalowaniu przez dystrybutora. Językami programowania, o których mowa, są: Fortran, Cobol, PL/1 i inne, których translatory powstaną. Wiele programów opracowywanych w CMS może być wykonanych pod kontrolą tego systemu dla zrealizowania określonego zadania. Programy, które używają funkcji specyficznych dla konkretnego systemu operacyjnego muszą być wykonywane pod kontrolą takiego właśnie systemu.



### Edytor CMS

Edytor CMS składa się z komendy EDIT i jej subkomend. Korzystając z edytora CMS, użytkownik może tworzyć pliki (zbiór rekordów) wprowadzając dane z terminala. Następnie może on przeglądać cały lub część utworzonego pliku oraz dopisywać, zmieniać lub usuwać rekordy.

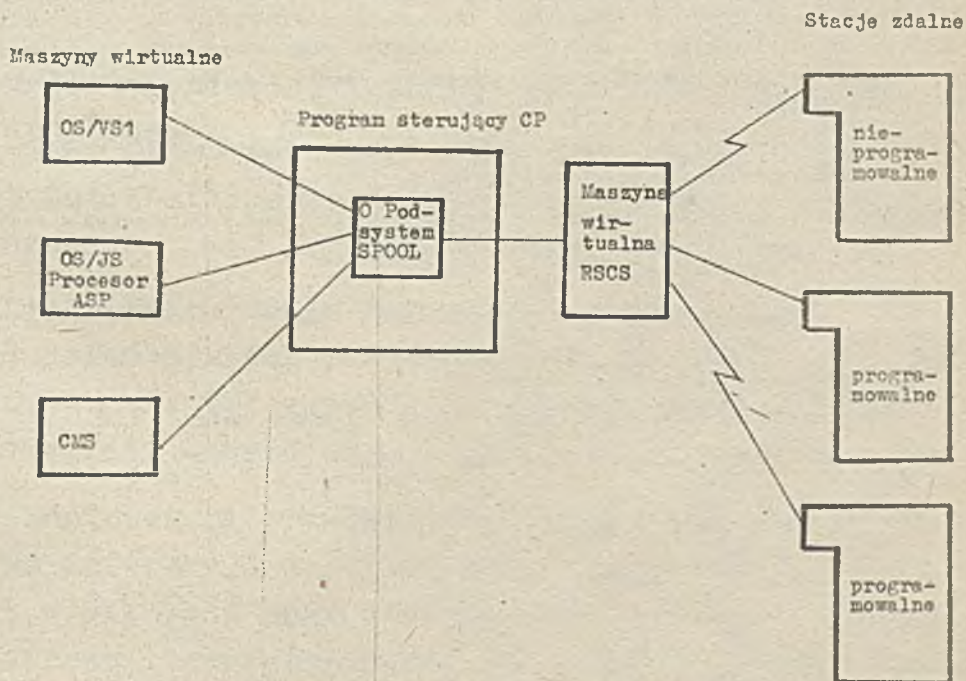
### System RSCS

System zdalnego przesyłania zbiorów (RSCS) jest komponentem VM/JS-P, który pracuje pod kontrolą CP. RSCS składa się z jądra i podsystemu do obsługi linii LINE DRIVER. Jądro jest łącznikiem między podsystemem SPOOL (część CP) do obsługi operacji SPOOL i podsystemem LINE DRIVER dla obsługi linii. LINE DRIVER kieruje pracą i kontroluje specyficzne typy stacji zdalnych. RSCS dostarcza środków telekomunikacyjnych do przesyłania plików pomiędzy stacjami zdalnymi i maszynami wirtualnymi używając linii binarnych synchronicznych (BSC) przełączalnych lub nieprzełączalnych. RSCS wspólnie z podsystemem SPOOL pozwala na przesyłanie plików:

- z maszyny wirtualnej do zdalnej stacji RSCS,
- ze zdalnej stacji RSCS do wsadowej maszyny wirtualnej (CMS).
- z jednej zdalnej stacji RSCS do drugiej zdalnej stacji RSCS,
- z maszyny wirtualnej do zdalnego wsadowego podsystemu wprowadzania zadań.

Przykładami podsystemów zdalnego wprowadzania zadań są: Hasp, ASP, JES, RES, CRJE. Podsystemy dla wprowadzenia zadań działają pod kontrolą systemów operacyjnych DOS, DOS/VSE, OS, OS/VSE.

Rys. 6 pokazuje zależność między maszynami wirtualnymi użytkowników, podsystemem SPOOL i stacjami zdalnymi.



Rys. 6. Sieć RSCS

## System IPCS

Interakcyjny system obsługi awarii IPCS jest komponentem systemu VM/JS-P, który pracuje w maszynie wirtualnej CMS. Ułatwia on konserwację systemu VM/JS-P gdyż:

- standaryzuje proces zbierania informacji o błędach VM/JS-P,
- identyfikuje powtarzające się błędy programowe (problemy) zlokalizowane na bazie wykorzystywania systemu.

Informacje o błędach, które wystąpiły mają formę raportów problemowych i notowane są w bazie. Również wykryte przez użytkownika awarie mogą być, jako problem, wprowadzane do bazy i uaktualniane, a następnie w postaci zestawienia - wydrukowane. Powyższa baza danych może być przeszukiwana pod kątem wcześniejsz wskazanego problemu. IPCS dostarcza środków dla:

- bieżącego konswersacyjnego szukania błędu (diagnostyka),
- analizy wypisu pamięci CP: wynikowe sformatowane dane są przechowywane na dysku jako unikalny raport,
- notowania awarii (problemów) wykrytych przez użytkownika w postaci unikalnego raportu,
- sprawdzania, czy sygnalizowany problem już wcześniej wystąpił i informowania o tym,
- uaktualniania i wypisywania (wyświetlania) informacji o zanotowanych problemach,
- konswersacyjnego analizowania wypisu pamięci CP.

## Podsumowanie

VM/JS-P pozwala na jednoczesne prowadzenie prac wymagających współpracy z różnymi systemami operacyjnymi lub działających niezależnie od systemu. Jego składowa - CMS jest bardzo wygodnym narzędziem do pisania i testowania nowych produktów programowych. Tak więc omawiany system jest przeznaczony dla dużych konfiguracji sprzętowych opartych na EMC R34 (lub kompatybilnej) w takich ośrodkach gdzie:

- prowadzone są zróżnicowane obliczenia wymagające wykorzystywania różnych systemów operacyjnych (w maszynach wirtualnych działają DOS, OS, VS),
- opracowywane są nowe produkty programowe przeznaczone do pracy w połączeniu z systemem operacyjnym lub działające niezależnie (praca w systemie CMS).

## Literatura

- [1] IBM Virtual Machine - Introduction
- [2] IBM Virtual Machine - Planning and System Generation Guide
- [3] IBM Virtual Machine - System Programmer's Guide

# sprawozdania

I Krajowa Konferencja "Informatyka w szkole"

Wałbrzych 12-13 czerwca 1985 r.

Organizatorami konferencji byli:

- Naczelna Organizacja Techniczna, Rada Wojewódzka w Wałbrzychu,
- Zakłady Elektronicznej Techniki Obliczeniowej w Świdnicy przy współpracy
  - Instytutu Kształcenia Nauczycieli im. Wł. Spasowskiego - Warszawa,
  - Instytutu Kształcenia Zawodowego - Warszawa,
  - Kuratorium Oświaty i Wychowania - Wałbrzych,
  - Instytutu Kształcenia Nauczycieli - Oddział Doskonalenia Nauczycieli - Wrocław

Na konferencji przedstawiono kilka interesujących referatów oraz informacji na temat obecnego zakresu stosowania informatyki w szkołach oraz zamiarów na przyszłość. Prezentowano również sprzęt i oprogramowanie używane w szkołach, jak i propozycje sprzętowe do stosowania w przyszłości. Były również referaty luźno, związane z informatyką.

Z tej ostatniej grupy jeden zasługuje na uwagę, a mianowicie referat "W poszukiwaniu alternatyw edukacji - Wrocławska Szkoła Przyszłości" - doc. dr hab. R. Łukaszewicza, który

w niekonwencjonalny sposób przedstawia metody edukacji młodzieży. Koncepcja ta traktuje szkołę jako instytucję zorganizowanego maksymalizowania okazji do rozwoju i kształcenia się dzieci i młodzieży. Zgodnie z tą koncepcją podana jest organizacja szkoły-laboratorium oraz architektura pomieszczeń szkolnych. W szkole przeznaczonej dla 420 uczniów ma pracować 42 nauczycieli i 8 osób serwisu technicznego.

Inne ciekawe tezy przedstawione w referacie to:

- oświata i edukacja stanowią drugi co do wielkości nakładów "przemysł" świata,
- olbrzymia jest waga społeczna zastosowań mikrokomputerowych (np. przy odpowiednich programach można wywoływać u młodzieży agresywność),
- mikroelektronika jest elementarnym narzędziem, a język informatyki elementarnym językiem, którym ludzie będą się posługiwać w przyszłości.

Dr A. Szejnberg w referacie "Możliwości zastosowania mikrokomputerów w praktyce szkolnej" twierdzi, że rozwój elektroniki, a przede wszystkim wykorzystanie radia, telewizji, magnetofonu i komputerów stanowi czwartą rewolucję w oświacie i wychowaniu. Zastosowanie mikrokomputera humanizuje proces nauczania, pozwalając nauczycielowi na poświęcenie większej ilości czasu na osobisty kontakt z uczniami, na poznanie ich faktycznych zdolności i zainteresowań.

Nauczanie wspomagane przez komputer realizuje takie formy nauczania, jak:

- uczenie się wg założonych programowo strategii i różnego rodzaju ćwiczeń,
- symulacje,
- gry decyzyjne,
- uczenie się sterowane przez komputer,
- demonstrowanie eksperymentów,
- kontrolowanie wyników nauczania.

Stosując mikrokomputery można przejść od "szkoły pamięci" do "szkoły myślenia".

W swoim referacie dr inż. St. Kwiatkowski podkreśla, że mikrokomputer jest tylko jednym z narzędzi używanych w procesie kształcenia, które ułatwia naukę logicznego myślenia.

Referat przedstawia podstawowe wymagania, jakie powinien spełniać minikomputer dla celów dydaktycznych. Są to:

- czas reakcji uwzględniający możliwości percepcyjne ucznia (przyjmuje się, że średni czas reakcji na informację wejściową wprowadzaną przez ucznia nie powinien przekraczać 1,5 s),
- możliwość prezentowania tekstów łącznie z rysunkami i wykresami,
- tworzenie pełnej dokumentacji przebiegu zajęć,
- ochrona zbiorów przed osobami nieupoważnionymi,
- możliwość definiowania znaków specjalnych, np. symboli stosowanych w matematyce, fizyce i chemii,
- wyposażenie klawiatury w specjalne klawisze sterujące,
- możliwości zobrazowania klawiatury na ekranie,
- prowadzenie dialogu za pomocą pióra świetlnego i innych urządzeń wejściowych, np. manipulatorów,
- zastosowanie koloru do kodowania informacji,
- tworzenie, wybieranie i modyfikowanie obrazów,
- generowanie i modyfikowanie obrazów trójwymiarowych,
- generowanie krzywych.

Wymagania ergonomiczne:

- wysoka jakość alfaskopów i grafoskopów ze szczególnym zwróceniem uwagi na:
  - rozmiar i pojemność ekranu,
  - pole zobrazowania,
  - zestaw kolorów umożliwiający długotrwałą pracę,
  - wydzielone pola znaków numerycznych, liter i znaków

- specjalnych,
- zmienny kąt nachylenia klawiatury i monitora,
- zastosowanie powłok antyodblaskowych.

W toku dyskusji podkreślono konieczność zwrócenia szczególnej uwagi na względy zdrowotne.

Spełnienie licznych postulatów natury dydaktycznej i ergonomicznej jest konieczne, ponieważ zgodnie z założeniami, mikrokomputer ma być głównym środkiem dydaktycznym w pakiecie multimedialnym.

Zasługujący na szczególną uwagę był referat dr J. Dunin-Borkowskiego pt.: "Dydaktyczne problemy zastosowania mikrokomputerów w nauczaniu fizyki".

Obecna fascynacja ogromnymi możliwościami komputera prowadzi do dążenia aby zastąpić nim inne znane narzędzia i środki, np. film. Chodzi jednak o to, aby jego zastosowanie służyło podstawowemu celowi nauczania, tj. przygotowaniu do życia i działania w nowoczesnym społeczeństwie. Stąd wnioszek, że w nauczaniu poszczególnych przedmiotów komputery powinny być przede wszystkim stosowane tak, aby odpowiadało to charakterowi danej dziedziny. Np. w nauczaniu fizyki powinno stosować się mikrokomputery do:

- modelowania,
- obliczeń i analizy wyników eksperymentalnych,
- symulacji eksperymentów,
- gier symulacyjnych i symulacji środowiska.

Natomiast minikomputery nie powinny zastępować doświadczeń laboratoryjnych.

W wykorzystywaniu komputerów w procesie nauczania konieczne jest zachowanie umiaru i niewpadanie w przesadną fascynację.

Dr hab. St. Waligórski przedstawił zalety języka LOGO jako, zarówno języka pierwszego zetknięcia się dzieci i młodzieży z komputerem, jak i narzędzia do rozwiązywania bardziej skomplikowanych zadań. Jedną z zalet języka jest dobra

grafika. Dużą przyszłość mają w tym zakresie zastosowań - języki PROLOG i SMALLTALK.

Język LOGO pozytywnie ocenił w swym referacie dr A. Walat twierdząc, że środowisko LOGO - to celowo skonstruowana Mat-kraina, w której dziecko znajduje duży wybór materiałów do myślenia matematycznego.

St. Waligórski, jako przedstawiciel PTI, zwrócił uwagę na kilka aspektów, które powinny być uwzględnione przy wprowadzaniu komputerów do procesu nauczania.

Szkolnictwo powinno być komputeryzowane sprzętem o wysokiej jakości. Ze względu na duże zapotrzebowanie powinien to być sprzęt krajowy, minikomputery z odpowiednio dużą pamięcią (minimum 64 kb RAM), grafiką, klawiaturą z polskim alfabetem; sprzęt ten ponadto powinien być tani.

Powszechne wprowadzenie sprzętu informatyki do szkół należy poprzedzić przygotowaniem odpowiedniego programu określającego cele dydaktyczno-wychowawcze, przygotowaniem odpowiedniego oprogramowania, dokumentacji użytkowej i wskazówek metodycznych oraz poprzedzić okresem doświadczalnym.

Właściwy przebieg pracy wymaga 4-6 mikrokomputerów i 2 drukarek na klasę, tzn. na jeden mikrokomputer ma przypadać 2-4 uczniów.

Przedstawiciel Ministerstwa Oświaty i Wychowania, dr F. Filipowicz, podkreślił szerokie poparcie Ministerstwa dla wszelkich inicjatyw we wprowadzaniu informatyki do szkolnictwa i podał konkretne działania władz w tym zakresie, jak np. rozpoczęcie szkolenia nauczycieli, zlecenie opracowania programu nauczania informatyki itp. Zwrócił uwagę na konieczność podjęcia przede wszystkim następujących działań:

- wypracowania nowych skutecznych form i metod kształcenia specjalistów i wykładowców w zakresie zastosowań technik mikrokomputerowych w procesie kształcenia dzieci i młodzieży,
- dokonanie wyboru mikrokomputerów, które spełniałyby wymagania edukacyjne i w które zaopatrzone byłyby szkoły różnych typów i stopni.

Program powszechnej edukacji informatycznej i program wdrażania mikrokomputerów do praktyki szkolnej w latach 1986-1990 ma powstać do września 1985 r.

Planowane jest też prowadzenie wspólnych badań naukowych w zakresie wprowadzania informatyki do szkolnictwa i wymiany doświadczeń między krajami socjalistycznymi. Bardzo ważną sprawą jest rozwiązanie problemu opracowania programów edukacyjnych dobrych pod względem rzeczowym i metodycznym.

W czasie trwania konferencji organizowano pokazy pracy na minikomputerach. Jeśli chodzi o sprzęt, to były to głównie mikrokomputery Spectrum, drukarki SEIKOSHA, krajowe magnetofony i telewizory. Niezbyt dobrą opinię miały polskie magnetofony ze względu na trudności w wymienności oprogramowania między poszczególnymi egzemplarzami. Był również wystawiony system ELWRO 523. Prezentowano zestaw laboratoryjny ZLA-01 (prod. AMEPROD) do wyszukiwania i usuwania uszkodzeń w układach mikroprocesorowych, testowania układów, programowania pamięci, itp.

Przedstawiciel firmy NOVATECH z Katowic reklamował systemy mikrokomputerowe oparte na SPECTRUM i SPECTRUM PLUS.

Obeonie w kraju najwięcej jest mikrokomputerów SPECTRUM i są one głównie stosowane eksperymentalnie w szkołach. Mogą one być w przyszłości wypierane przez MERITUM, pod warunkiem, że będzie rozbudowana pamięć do minimum 48 kb i wbudowana grafika. Znaczący wpływ może mieć również produkcja ELWRO (mikrokomputer ELWRO 700).

Z powyższego wynika, że wprowadzanie informatyki do szkolnictwa jest w fazie początkowej. Dlatego też staranne przygotowanie programów i wybór odpowiedniego sprzętu będzie miał olbrzymie znaczenie dla przyszłego kształcenia młodzieży i przygotowania kadr dla nowoczesnego rozwoju kraju.



# nowości techniczne

## Komputery przy projektowaniu i wytwarzaniu

Projektowanie i wytwarzanie wspomagane komputerami rozwija się bardzo dynamicznie, czego dowodem są częste wystawy sprzętu i usług w tej dziedzinie. Jedną z nich odbyła się w Brighton w dniach 26-28 marca 1985 r. Organizowana była przez Reed Exhibitions, a większość wystawców to firmy brytyjskie, które odgrywają obecnie poważną rolę w tej bardzo konkurencyjnej dziedzinie. W przeciwieństwie do początkowego okresu kiedy pionierzy CAD/CAM (Computer Aided Design, Computer Aided Manufacturing - projektowanie wspomagane komputerowo, wytwarzanie wspomagane komputerowo) szukali zastosowań dla opracowanych przez siebie systemów, obecnie dostawcy przystosowują swój sprzęt i usługi do wymagań klientów. Wystawiany sprzęt obejmuje szeroki zakres urządzeń od małych systemów biurkowych po rozbudowane systemy z wieloma końcówkami i różnymi układami kreślącymi, a usługi obejmują projektowanie oprogramowania, analizę systemową i konserwację. Wiadomo, że potencjalny klient szuka obecnie systemów, które są łatwe w obsłudze, spełniają jego specyficzne wymagania i są osiągalne za rozsądną cenę.

Wśród wystawiających firm należy wymienić Norrie Hill, która wytwarza systemy do projektowania urządzeń mechanicznych, elektronicznych i elektrycznych. Głównym jej produktem jest system Soroc 36 z nowym, rozszerzonym oprogramowaniem. Charakteryzuje się on pełnym dostępem do rozkazów systemowych i nieograniczoną powierzchnią kreślenia. Jest to system niezależny, do którego można dołączyć 12 różnych stanowisk, a znacznie więcej przez multipleksery.

Firma Marconi Instruments pokazała system kreślący Quadrant 2. Jest to dwuwymiarowy system o wysokiej wydajności, obejmujący sprzęt i oprogramowanie pozwalające na sprawne wytwarzanie mechanicznej i elektrycznej dokumentacji wszelkiego rodzaju. Jest on oparty na mikroprocesorze i jest wyposażony

w dysk typu Winchester o pojemności 40 Mbajtów pozwalający na przechowywanie 250 złożonych projektów. Rozszerzone oprogramowanie tego systemu pozwala na wskazywanie wyświetlanych zleceń, stosowanie do 10 "okienek" dla różnych programów oraz uzyskiwanie efektów panoramiczności i zmiennej ogniskowej. Cechy wydawnicze obejmują usuwanie, przesuwanie i przenoszenie poszczególnych obszarów rysunków, a także obrót i zmiany typu elementów. System może być kupowany jako niezależny, a następnie rozszerzany do systemu z wieloma stanowiskami. Podobnymi cechami charakteryzuje się system Radraft firmy Radan Computational, której najnowsze stanowisko robocze nosi nazwę Whitechapel MG1 i może być na zasadzie sieci łączone z innymi końcówkami i komputerami w zależności od wymagań. Stanowisko wyposażone jest w ekran czarno-biały o rozdzielczości 1000 x 800 elementów i wielkości przekątnej 432 mm. Jest ono szczególnie przydatne do zastosowań szkoleniowych, gdyż szkolenie w tej dziedzinie jest sprawą istotną.

Firma Ferranti Infographics opracowała w końcu 1984 r. system CAM-X, który może być stosowany, zarówno do szkolenia, badań, jak i projektowania. Jego oprogramowanie pozwala na modelowanie trójwymiarowe. System ma rozległą bibliotekę programów, którą można rozszerzać. Na Uniwersytecie w Glasgow wykorzystuje się go do badań i szkolenia, w firmie F & O Electronics Systems w RFN służy do wytwarzania dokumentacji drukarek elektronicznych, a w laboratoriach BSRA (British Ship Research Association) do projektowania statków.

Nieco później rozwinęło się wspomagane komputerowo wytwarzanie. Wyróżnia się tu firma Micro Aided Engineering, która oferuje programy do obrabiarek sterowanych numerycznie oraz inne sterujące poszczególnymi etapami produkcji. System Maelink pozwala programiście części wytwarzanych na obrabiarce numerycznych czerpać dane dotyczące wymiarów geometrycznych z systemu kreślącego Maedos tej samej firmy, co oszczędza weryfikację tych wymiarów, a także dobór narzędzi i sposobów podawania.

Firma Deltacam Systems oferuje system DUCT, który obejmuje projektowanie, kreślenie i wytwarzanie oraz trójwymiarowe modelowanie powierzchni. Realizowany jest on na małym komputerze 32-bitowym z kolorowym monitorem. Kształty zaprojektowane w tym systemie mogą być wytwarzane na sterowanych numerycznie obrabiarkach wyposażonych w moduł DUCT. Jest to szczególnie użyteczne przy złożonych wypraskach z tworzywa sztucznego, odlewach metalowych i odkuwkach. System ten może być łączony z systemem kreślącym DOGS i obrabiarkowym GNC tej samej firmy. Deltacam oferuje też oprogramowanie o nazwie MOLDFLOW optymalizujące projektowania i wytwarzanie wyprasek z tworzywa sztucznego na podstawie analizy przepływu cząsteczek tego tworzywa. Może on być realizowany na minikomputerze. Firma oferuje dostosowanie oprogramowania do wymagań klientów, co zresztą jest cechą większości wytworów firm brytyjskich.

London Press Service TE 2241/4

### Nowe oprogramowanie szkoleniowe

Britain's National Centre for Information Technology, który dawniej nosił nazwę National Computer Centre wypuścił nową wersję oprogramowania do szkolenia analityków systemowych. Instytucja ta działa już od dwudziestu lat i wyszkoliła tysiące analityków na całym świecie.

Obejmuje ono 180-godzinny kurs i może być stosowane na uczelniach, ośrodkach szkolenia dla dorosłych oraz w szkoleniach firmowych.

Organizacja ta oferuje 450 różnych materiałów szkoleniowych z różnych dziedzin techniki informacji, prowadzi corocznie 72 kursy w Londynie i 12 zagranicą. Oprogramowanie szkoleniowe rozprowadzane jest poza Wielką Brytanią przez 30 dystrybutorów. Jest ono wykorzystywane przez ośrodki szkoleniowe.

we na Cyprze, w Dubaju, Egipcie, Bahrajnie i Kuwejcie, a za pośrednictwem British Council - w wielu innych rozwijających się krajach. Ponadto ośrodek ten rozwija działalność wydawniczą w dziedzinie informatyki, która w 1983 r. w 40% przeznaczona była na eksport. Wydano wówczas 32 nowe tytuły, a 20% wydanych książek miało wznowienie, tak że liczba tytułów książek w druku wynosiła 126. Wśród książek wydanych przez NCC dobrą opinią cieszyła się pozycja "W stronę komputerów piątej generacji" pióra wydawcy NCC, Geoffa Simonsa. Została ona przetłumaczona na język japoński i portugalski.

London Press Service

BR24A185

Oprogramowanie brytyjskie dla Dowództwa  
Marynarki USA

United States Naval Sea Systems Command (NAVSEA) zakupiła od brytyjskiego zrzeszenia BSRA (British Ship Research Association) systemy B-Lines i HULISURF do obliczeń hydrodynamicznych. Będą one realizowane na komputerze VAX 11/780 w centrali NAVSEA w Crystal City pod Waszyngtonem. Systemy te dostosowane będą do bazy danych opracowanej w NAVSEA, a BSRA podjęła się również przeszkolić na miejscu personel.

System B-Lines jest wspomaganym komputerowo systemem określania powierzchni kadłubów statków. Umożliwia on szybkie i dokładne projektowanie kształtów kadłubów i określanie ich własności hydrostatycznych. System został opracowany przez konstruktorów statków i jest dostosowany do personelu, który nie musi mieć wykształcenia informatycznego. Daje on pełny opis kształtu kadłuba na podstawie danych wejściowych, jak przyjęty profil, płaskość boków, dna itp. Wyliczone krzywe wchodzi do bazy danych do dalszych operacji.

System HULL-SURF jest wspomaganym komputerowo systemem projektowania i manipulowania powierzchniami kadłuba za pomocą grafiki interaktywnej. Powierzchnie te określa się za pomocą krzywych z systemu B-Lines przy założeniu ciągłości na granicach. System ten opracowany jest do projektowania kadłubów statków ale może być również zastosowany do modelowania śrub napędowych, śmigieł i płatów lotniczych. Umożliwia on także łączenie krzywych i powierzchni oraz określanie przejść między nimi, jak również przecięć pomiędzy powierzchniami i płaszczyznami oraz obliczanie pól i objętości. Własności geometryczne tych połączeń mogą być obliczane i wyświetlane.

London Press Service  
5BIN1184

#### Tani system dotykowy

Firma Microvitex z Bradfordu w pñn. Anglii oferuje nowy system o nazwie Touchtech 501, którym można sterować dotykając jedynie ekranu monitora. Umożliwia to pracę dzieciom lub inwalidom, bez potrzeby zapoznawania się z klawiaturą. Koszt tego systemu określony ok. 200 funtów jest znacznie niższy od innych tego rodzaju systemów. Ekran obramowany jest uchwytem, w którym zamocowane są czujniki podczerwieni. Wiązki nieszkodliwych promieni podczerwonych wysyłane są przed ekranem do tych czujników i jeśli ręka lub wskaźnik przerwie obwód tych promieni informacja o tym jest przekazywana do komputera. Taki sposób komunikowania się z systemem jest w pełni bezpieczny.

Podręcznik użytkownika w prosty sposób opisuje istniejące programy i ich działanie. Załączona dyskietka zawiera dziewięć programów demonstracyjnych.

Oprogramowanie opracowane jest w ramach programu oświatowego korzystającego z wsparcia rządowego i pokazuje możliwości systemu, który zawiera procesor sterujący łączności z komputerem głównym i innymi układami przetwarzania. Obciążenie wynosi tylko 0,3A przy 5V, a przesyłanie informacji odbywa się z prędkością 9600 bodów przez szeregowy port.

London Press Service

### Brytyjskie oprogramowanie dla Japonii

Brytyjskie przedsiębiorstwo Disco International Ltd z Taunton w pld.zach. Anglii odstąpiło Japońskiej firmie Sumitomo Corporation prawo do opracowania japońskiej wersji swego systemu Viewbase będącego interakcyjnym systemem wideo-tekstowym działającym pod systemem operacyjnym Eick na szerokim zakresie mikro, mini i dużych komputerów. Daje on pełne możliwości pracy interakcyjnej i przeprowadzania transakcji, przy czym nowe zastosowania mogą być tu szybko opracowywane, a istotne cechy systemów wideo-tekstowych, jak prostota działania i korzystne parametry ekonomiczne są w pełni zachowane. Może on pracować z różnymi międzynarodowymi protokołami wideo-tekstowymi, spośród których najistotniejszy jest NAPLPS (North American Presentation Level Protocol Syntax - Północnoamerykańska składnia protokołów na poziomie przedstawiania) zawierający określenia złożonych znaków graficznych i dostosowany do języka japońskiego. Współpracującą z Sumitomo firma Tau Engineering oferuje SNS 210, pierwszy japoński minikomputer pracujący pod systemem Eick. Negocjowane są porozumienia z innymi wytwórcami m.in. z Prime Computer, która osiągnęła znaczne sukcesy z Viewbase na innych rynkach.

System stosowany jest w handlu, zarządzaniu, gospodarce finansowej, ubezpieczeniach, zakupach, w domu, bankowości i in.

London Press Service 30BIN185

### Przeniesienie operacji bankowych z domu

W wielkiej Brytanii wprowadzone zostało udogodnienie polegające na możliwości zarządzania z domu lub z biura kontami bankowymi, zarówno osobistymi, jak i przedsiębiorstw. Umożliwia to najnowszy system bankowy, bardzo wygodny dla użytkownika. Domowe usługi Banku Szkocji wymagają jedynie odbiornika telewizyjnego, aparatu telefonicznego i połączenia z krajową bazą danych Prestel. Klienci bankowi mają dostęp do swych rachunków w ciągu 17 godzin na dobę i mogą dokonywać operacji natychmiastowych lub wyznaczać określoną datę płacenia rachunków lub składania zamówień.

Bank Szkocji oferuje już obecnie przedsiębiorstwom połączenie ze światową siecią bankową, a w niedalekiej przyszłości będzie to możliwe i dla klientów prywatnych. Bank ten od 1960 r. stosuje elektroniczne przetwarzanie danych i doświadczenia te zostały wykorzystane przy opracowywaniu omawianego systemu.

Zwrócono szczególną uwagę na zabezpieczenie przed niepowołanym dostępem. Każdy użytkownik ma osobisty numer prywatny oraz możliwość wprowadzenia hasła, które znane jest tylko jemu i które może być zmienione w razie potrzeby.

London Press Service 30BIN 185





Informacja o cenach i warunkach prenumeraty na 1986 r.  
- dla czasopism Instytutu Maszyn Matematycznych

● Cena prenumeraty rocznej

Techniki Komputerowe - Biuletyn Informacyjny	1560.-	dwum.
Przegląd Dokumentacyjny - Nauki i Techniki Komputerowe	1260.-	dwum.
Informacja Ekspresowa - Nauki i Techniki Komputerowe	2400.-	mies.
Prace naukowo-badawcze Instytutu Maszyn Matematycznych	660.-	3x w roku

● Warunki prenumeraty

1/ dla osób prawnych - instytucji i zakładów pracy:

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

2/ dla osób fizycznych - prenumeratorów indywidualnych:

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

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

● Terminy przyjmowania prenumeraty na kraj i za granicę:

- do dnia 11 listopada na I kwartał, I półrocze roku następnego oraz na cały rok następny,
- do dnia 1-każdego miesiąca poprzedzającego okres prenumeraty roku bieżącego.

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

# informacja ekspresowa



NAUKI  
I TECHNIKI  
KOMPUTEROWE

Instytut Maszyn Matematycznych zawiadamia, że od 1984 r., po dwuletniej przerwie, wznawia wydawanie miesięcznika "Informacja ekspresowa - Nauki i Techniki Komputerowe". W czasopiśmie zamieszczamy opisy bibliograficzne /wraz z krótkimi notatkami objaśniającymi/ dokumentów źródłowych, które znajdują się w bibliotece IMM - najlepiej zaopatrzonej w branży komputerowej.

Dokumentujemy ok. 600 pozycji książkowych rocznie /krajowych, i zagranicznych/ oraz 184 tytuły czasopism /około 2000 zeszytów/ w językach: polskim, angielskim, rosyjskim, niemieckim, czeskim; katalogi i in.

Informacja ekspresowa NITK informuje o najnowszych publikacjach z zakresu branży komputerowej i dziedzin pokrewnych oraz nauk związanych z branżą /monografie, słowniki, podręczniki, materiały szkoleniowe, artykuły w czasopismach, przyczynki, krótkie notatki o najnowszych zdobyczach techniki komputerowej na świecie itp./ jest więc podstawowym i niezbędnym narzędziem pracy każdego pracownika naukowego, studenta, inżyniera - praktyka, projektanta i in.

Nasi Czytelnicy mogą zamawiać mikrofilmy i kserokopie dokumentów, których opisy znajdują się w Informacji ekspresowej.