

mikroKLAN 10

10

1984

P. 1877/84

informatyka

Narzędzia inżynierii oprogramowania
Mikroprocesory
MTP'84

Nr 10

Miesięcznik Rok XIX

Październik 1984

Organ Komitetu Informatyki
MNSzWiT oraz Komitetu
Naukowo-Technicznego NOT
ds. Informatyki

KOLEGIUM REDAKCYJNE:

Mgr inż. Zbigniew GLUZA, dr inż. Wa-
claw ISZKOWSKI, mgr Teresa JAB-
ŁONSKA (sekretarz redakcji), Włady-
sław KLEPACZ (zastępca redaktora na-
czelnego), prof. dr hab. Leon ŁUKA-
SZEWICZ (redaktor naczelny), mgr inż.
Andrzej J. PIOTROWSKI, dr inż. Ja-
nusz ZALEWSKI

STALE WSPÓLPRACUJĄ:

Mgr inż. Witold ABRAMOWICZ (Szwaj-
caria), mgr Adam B. EMPACHER, mgr
Katarzyna ISAAK, dr Jacek OWZAR-
CZYK, mgr Marek SOBCZYK, dr An-
drzej SZALAS, dr Jakub TATARKIE-
WICZ, mgr inż. Teresa WILCZEK

**PRZEWODNICZĄCY
RADY PROGRAMOWEJ:**

Prof. dr hab. Tadeusz PECHE

Materiałów nie zamówionych redakcja
nie zwraca

Redakcja: 00-041 Warszawa, ul. Jas-
na 14/16, pok. 243 i 244, tel. 27-71-40 lub
26-82-61 w. 184

Zakł. Graf. „Tamka”. Zam. 2249. Obj.
4,0 ark. druk. Nakład 4250 egz. T-44.

INDEKS 36124

Cena egzemplarza zł 75,—
Prenumerata roczna zł 900,—



00-950 Warszawa
skrytka pocztowa 1004
ul. Biała 4

W NUMERZE:

	Strona
Narzędzia inżynierii oprogramowania (1) <i>Roman Żelazny</i>	1
Mikroprocesory lat osiemdziesiątych <i>Jerzy Szyller</i>	5
CHILL — język programowania systemów komutacyjnych (2) Budowa programu i wykonanie współbieżne <i>Władysław Udrycki, Wiesław Wilczyński</i>	9
Sterowanie alfaskopem z zastosowaniem układu SY 6545 <i>Józef de Mezer</i>	21
mikroKLAN	13
— Jednopłytkowy mikrokomputer VU68K	
— Sterowanie napędami dysków elastycznych (1)	
— Jak zainstalować FORTH (3)	
ZE ŚWIATA	23
— System LEXIS	
— Ewolucje układów	
Z KRAJU	25
— 56 MTP. Mikroinformatyki coraz więcej	
— 56 MTP: Szlaban dla mikro	
TERMINOLOGIA	31
— Dokumentacja oprogramowania (3)	

W NAJBLIŻSZYCH NUMERACH:

- Uwe Petermann i Andrzej Szalas o przerzaniach
- Emanuel Czyżo i Teresa Latusek o prekursorach informatyki
- Rafał Zieleniewski o problemach programowania mikroprocesorów
- Wiesław Wilczyński o języku PLZ
- Ryszard Rybus o programach uruchomieniowych dla mikroprocesorów
- Ryszard Kott i Krzysztof Szwed o uniwersalnych asemblerach dla mikrokomputerów
- Artur Krępski o PASCALU MT+
- Zbigniew Banasik o normie IEEE — arytmetyka zmiennoprzecinkowa
- Ryszard Kott o BASICU dla mikrokomputerów
- Waldemar Kapuścik o projektowaniu sieciowych baz danych
- Konrad Jabłoński o języku BCPL
- Waclaw Iszkowski o mechanizmach komunikacji
- Marek Dziedzic, Krzysztof Perycz i Jerzy Wiliński o emulatorze E6RM-E



Narzędzia inżynierii oprogramowania (1)

... narzędzia, których próbujemy używać, i język lub notacja, których używamy do wyrażenia lub zarejestrowania naszych myśli, są głównymi czynnikami określającymi to, co w ogóle możemy myśleć lub wyrazić!

E. W. Dijkstra

Odkrycie i użytkowanie narzędzi pracy „stworzyło” człowieka. Z każdym zawodem wiąże się posiadanie narzędzi umożliwiających świadomą działalność twórczą czy wykonawczą człowieka. Są one rezultatem długiego procesu rozwoju, ewolucji. Podobna sytuacja istnieje w dziedzinie produkcji i rozwoju oprogramowania.

Po początkowym okresie zachwyty i oczarowania sukcesami programowania i zastosowań komputerów w wielu dziedzinach, przyszedł okres refleksji i krytycyzmu. Okazało się, że istnieją objawy świadczące o pewnym kryzysie w dziedzinie oprogramowania, polegającym na występowaniu takich zjawisk, jak [9]:

- przekraczanie założonych kosztów i terminów realizacji
- ograniczenie możliwości wglądu w stan zaawansowania prac programistycznych
- zawodność oprogramowania
- trudności w użytkowaniu oprogramowania i jego pielęgnacji
- niezadawalająca skuteczność weryfikacji i testów kontrolnych
- brak lub nieadekwatność dokumentacji.

Kolejne zmiany zadań i celów stawianych przed programistami i twórcami systemów programistycznych przekształcały stopniowo tę dziedzinę w dyscyplinę zwaną inżynierią oprogramowania.

Bardziej systematyczna analiza procesu produkcji oprogramowania i jego użytkowania wykazała, że można go podzielić na fazy z dość dobrze określonymi wejściami i wyjściami.

Prof. ROMAN ŻELAZNY jest profesorem zwyczajnym fizyki teoretycznej (teoria transportu cząstek — neutronów i plazmy, metody matematyczne fizyki, fizyka komputerowa). W roku 1950 ukończył Politechnikę Łódzką z tytułem mgr. inżyniera mechanika w 1952 Uniwersytet Warszawski z tytułem magistra filozofii w zakresie fizyki. W roku 1958 obronił doktorat z fizyki statystycznej. Jest twórcą polskiej szkoły transportu neutronów i obliczeń reaktorowych. Współorganizował międzynarodową współpracę w dziedzinie fizyki reaktorowej. Był twórcą Środowiskowego Centrum Obliczeniowego CYFRONET b. IBJ i Systemu Abonenckiego (1972—1983), b. wicedyrektorem IBJ ds. Fizyki i Techniki Jądrowej (1971—1981), b. członkiem Zarządu Grupy Fizyki Komputerowej Europejskiego Towarzystwa Fizycznego (w latach 1980—1981 — przewodniczącym Zarządu).



Mimo że istnieją pewne spory i różne podejścia, to jednak coraz bardziej ugruntowuje się pogląd, że proces ten dzieli się na następujące fazy tworzące model cyklu produkcyjnego:

- sformułowanie i analiza wymagań użytkowych
- projektowanie ogólnej struktury systemu
- projektowanie szczegółowe składników systemu
- realizacja programów
- sprawdzenie i weryfikacja systemu i jego składników
- użytkowanie, pielęgnacja i ewentualnie modyfikacje systemu.

Pierwsza faza powinna zakończyć się — na podstawie sformułowania i zdefiniowania wymagań użytkowych przez zamawiającego — przedstawieniem specyfikacji oprogramowania zarówno pod względem funkcjonalnym, jak i mierzalnych parametrów użytkowych. Projektowanie ogólnej struktury systemu powinno zakończyć się określeniem globalnej architektury systemu (modelem systemu), który pozwoli na realizację wymagań (specyfikacji) użytkownika. Faza projektowania szczegółowego winna być poświęcona analizie i sformułowaniu (specyfiki) zadań w stosunku do składników (modułów) systemu wraz z określeniem ich wzajemnych powiązań (oddziaływań), czyli tzw. sprzężeń (ang. interfaces). Następną fazą, programowanie lub implementacja, polega na tym, że na podstawie dokumentacji (specyfikacji) z poprzednich faz wytwarza się w określonym języku programowania wszystkie składniki systemu, a w końcu — cały system. Faza sprawdzania i weryfikacji polega na sprawdzeniu zgodności poszczególnych modułów i całego systemu ze specyfikacjami użytkowymi i projektowymi pod względem funkcjonalności i wartości parametrów użytkowych. Produktem końcowym jest dokumentacja obejmująca sprawdzony, zweryfikowany program (system) oraz opis jego użytkowania i pielęgnacji.

Dla osiągnięcia tego celu niezbędny jest staranny nadzór nad całym cyklem produkcyjnym każdego większego zamierzenia projektowego w dziedzinie oprogramowania. Aczkolwiek zespoły pracujące nad oprogramowaniem powinny być jak najmniejsze oraz złożone z najlepszych fachowców, to jednak tworzenie współczesnego oprogramowania ze względu na jego wielkość i skomplikowanie wymaga pracy kolektywnej. Oprogramowanie jest produktem trudno mierzalnym (przynajmniej przed zakończeniem jego realizacji), praca nad nim musi być więc nadzorowana oraz organizowana w sposób szczególnie staranny. Niestety, zasady organizacji tego procesu i jego nadzoru nie są jeszcze w pełni opracowane i opanowane pod względem koncepcyjnym i praktycznym. Wymagają specjalnych narzędzi i metod pracy. Część z nich realizowana jest przez nałożenie na zespół określonej dyscypliny pracy, zalecenie standardów dokumentacji oraz metod weryfikacyjnych. Dla ułatwienia realizacji wspomnianych zasad, w całym cyklu produkcyjnym — w ramach organizacji pracy zespołowej przy projektowaniu i budowie oprogramowania — opracowano wiele narzędzi. Można je ogólnie podzielić na narzędzia poznawcze, wspomagające oraz opisowe (notacyjne).

Narzędzia poznawcze rozszerzają intelektualne możliwości twórców oprogramowania przez dostarczanie im metod i technik ułatwiających rozwiązywanie problemów, takich jak dekompozycja hierarchiczna, lokalizacja informacji, programowanie strukturalne itp.

Narzędzia wspomagające zwiększają produktywność programistów. Zazwyczaj obejmują one takie narzędzia implementacji, jak kompilatory, edytory tekstów, pakiety uruchomieniowe (ang. debugging packages), jak również narzędzia dla faz przedimplementacyjnych (definiowanie wymagań użytkowych i projektowych) oraz poimplementacyjnych (weryfikacja, sprawdzanie i pielęgnacja).

Narzędzia opisowe (notacyjne) — jak np. języki, formalizmy modelujące — dostarczają środków dla wyrażania i komunikowania idei, pojęć, struktur, procesów czy zależności.

Najbardziej znane narzędzia inżynierii oprogramowania, uporządkowane od prostych, do bardziej skomplikowanych, zautomatyzowanych lub wspomaganych komputerów przedstawiono w zamieszczonym zestawieniu [7].

NARZĘDZIA I METODY DEFINIOWANIA I SPECYFIKACJI WYMAGAŃ UŻYTKOWYCH

Niesformalizowane specyfikacje w języku naturalnym
 Niesformalizowane diagramy i struktury
 Sformalizowane diagramy i struktury (np. HIPO, SADT, DATA-FLOW)
 Niesformalizowane plany testów i weryfikacji
 Skomputeryzowane specyfikacje
 Skomputeryzowane testy i weryfikacje wymagań użytkowych
 Zautomatyzowane archiwa specyfikacji wymagań użytkowych
 Narzędzia weryfikowania specyfikacji
 Metoda stopniowego rozbudowywania specyfikacji użytkowych
 Formalne przeglądy specyfikacji użytkowych

NARZĘDZIA I METODY SPECYFIKACJI PROJEKTOWYCH

NiefORMALNE funkcjonalne specyfikacje projektowe
 Niesformalizowane plany testów i weryfikacji projektowych
 Zautomatyzowane skorowidze danych
 Sformalizowane metody specyfikacji projektowych (np. PDL, metoda Jacksona, projektowanie strukturalne)
 Skomputeryzowane specyfikacje projektowe
 Skomputeryzowane metody testów i weryfikacji projektowych
 Weryfikatory sprzężeń modułowych
 Skorowidze modułowe
 Zautomatyzowane archiwa specyfikacji projektowych
 Formalne przeglądy projektowe
 Metoda stopniowych przeglądów i projektowania
 Plany techniczne projektu
 Zautomatyzowane narzędzia symulacji (np. SIMSCRIPT, GPSS)

NARZĘDZIA I METODY PROJEKTOWANIA

Procesor tekstów (edytor, system zapisów)
 Zautomatyzowany procesor programów źródłowych
 Skorowidz programu
 Niesformalizowane plany testów i weryfikacji programów
 Zautomatyzowany zarządca konfiguracji programów (nadzór nad wersjami)
 Generator diagramów przepływu danych
 Weryfikator programów źródłowych
 Archiwizator programów

NARZĘDZIA WERYFIKACJI

Narzędzia do porównywania zapisów
 Analizator narzędzi weryfikacyjnych (ich zakresu)
 System sprawdzania i weryfikacji (testów kontrolnych)
 Weryfikator modułów (składników) programistycznych
 Generator danych, dla celów weryfikacji
 Monitor parametrów użytkowych
 Analizator przepływu sterowania
 Analizator przepływu danych
 Archiwizator planów, sprawozdań i weryfikacji (testów kontrolnych)

NARZĘDZIA I TECHNIKI NADZORU I ORGANIZACJI PRACY

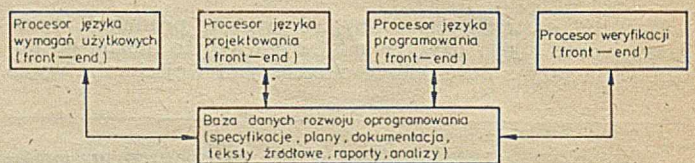
Systematyczne metody nadzoru ręcznego (np. diagramy celów, karty Gantta)
 Zautomatyzowane systemy nadzoru i kontroli przebiegu prac (np. PERT, ścieżki krytycznej, Gantta)
 Generatory raportów stanu prac
 Narzędzia do przedstawiania planów budowy systemów

Taka różnorodność narzędzi i metod wymaga ich integracji w postaci metodologii, a więc stworzenia środowisk rozwoju (budowy) systemów oprogramowania. Takie metodologie składają się zazwyczaj z metod technicznych, procedur

nadzoru i organizacji oraz zautomatyzowanych narzędzi pomocniczych, zintegrowanych w środowisku rozwoju (budowy) oprogramowania. Istnieje pewien zbiór wymagań lub pożądanych charakterystyk, jakie winny spełniać te metodologie — po to, by pozwalały one zwiększać produktywność i efektywność czynności związanych z budową oprogramowania. Według [12] metodologia powinna m.in.:

- obejmować pełny cykl produkcyjny oprogramowania
- ułatwiać przejścia między wszystkimi fazami cyklu w różnych kierunkach
- ułatwiać określanie i osiąganie poprawności oprogramowania w ciągu całego cyklu produkcyjnego — zarówno z funkcjonalnego, jak i wykonawczego punktu widzenia
- wspierać organizację zespołu programistycznego oraz jego nadzór przez zwiększenie komunikacji zarówno wewnątrz zespołu, jak i na zewnątrz, przez lepsze rozczepienie postępów prac i stanu realizacji zadań oraz przez znajomość innych charakterystyk oprogramowania — nawet tych, które nie były formułowane explicite
- pozwalać na zastosowanie do szerszej klasy zadań programistycznych
- być łatwa w użyciu i nie wymagać od wszystkich zainteresowanych zbyt dużych wysiłków przy jej opanowywaniu
- być wspierana przez zautomatyzowane narzędzia, zintegrowane w jednorodnym, elastycznym i zgodnym środowisku komputerowym.

Na to, by te koncepcje miały szansę realizacji, baza środowiskowa rozwoju (budowy) oprogramowania musi być — zgodnie z powszechną opinią — bazą danych z właściwym systemem zarządzania, czyli tzw. bazą danych inżynierii oprogramowania. W tej bazie winny być ulokowane wszystkie istotne informacje dotyczące całego procesu budowy oprogramowania, a wszystkie czynności poszczególnych faz cyklu produkcyjnego winny być przeprowadzane za pomocą stosownych metod i narzędzi wykorzystujących efektywnie zawartość bazy danych. Architektura takiego systemu produkcji (budowy) oprogramowania przedstawiona jest schematycznie na rysunku 1. Opiera się ona na koncepcji czterech procesorów wysuniętych (ang. front-end), odpowiadających czterem odrębnym fazom procesu produkcyjnego: definicji wymagań użytkowych, projektowaniu, programowaniu i weryfikacji. Procesory te pozwalają twórcom oprogramowania formułować koncepcje i zamierzenia w języku mniej lub bardziej sformalizowanym oraz przeprowadzać stosowane analizy, tak aby sfinalizować je mniej lub bardziej sformalizowanymi specyfikacjami wymagań użytkowych, specyfikacjami projektowymi i programami. Są one podstawą procesu weryfikacji i testowania całego cyklu produkcyjnego.

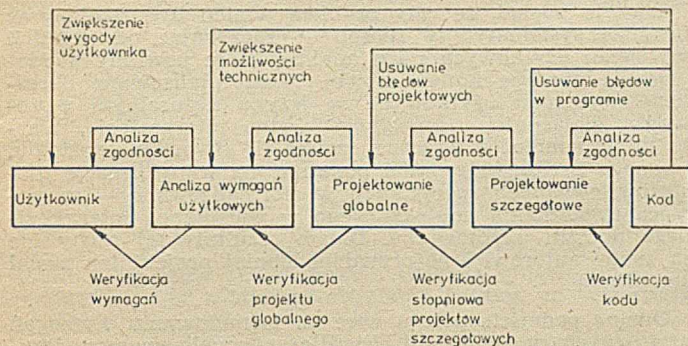


Rys. 1. Schemat architektury systemu produkcji oprogramowania

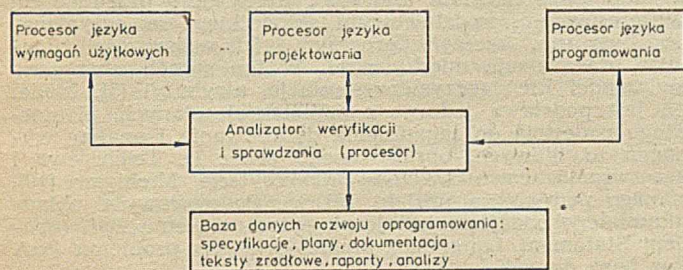
Według innych poglądów [9] proces weryfikacji i testowania winien przenikać wszystkie etapy (fazy) procesu produkcyjnego. W szczególności — przejścia z jednej fazy do drugiej wymagają szczegółowej weryfikacji i testowania (sprawdzania), po to, by przekonać się czy w danej fazie zrealizowano cele, które zostały sformułowane w stosunku do niej w fazach poprzednich.

Możliwość pełnego przeanalizowania uzyskanych rezultatów oraz weryfikacja parametrów technicznych, osiągniętych w budowie oprogramowania, jest we wszystkich zadaniach w tej dziedzinie problemem bardzo ważnym — z punktu widzenia nadzoru i organizacji pracy zespołu programistycznego. Uważa się więc, że weryfikacja i sprawdzanie nie mogą być traktowane jako oddzielne fazy cyklu produkcyjnego, lecz powinny być przeprowadzane w każdej fazie przy przejściu do fazy następnej. Wnioski należy wyciągać nie tylko w odniesieniu do specyfikacji i sformułowań fazy poprzedniej, lecz również w stosunku do kolejnych poprzednich faz aż do definiowania wymagań

użytkowych wyłącznie. Jest to ewolucyjny proces dochodzenia do końcowego kształtu systemu — przez twórców i zamawiającego — dzięki zdobywanemu doświadczeniu oraz uczeniu się w trakcie tego procesu (rys. 2 i 3).



Rys. 2. Weryfikacja, sprawdzanie i korygowanie w cyklu produkcyjnym



Rys. 3. Użycie bazy danych w cyklu produkcyjnym oprogramowania

ŚRODOWISKA PROGRAMISTYCZNE

Przedyskutujmy teraz problemy związane z integracją narzędzi oprogramowania w jedno środowisko lub system, a także idee i przykłady ich realizacji, obejmujące w jakimś stopniu cały cykl produkcyjny oprogramowania.

Historycznie rzecz biorąc, pierwsze wysiłki mające na celu stworzenie środowiska narzędzi programistycznych były oparte na koncepcjach i środkach związanych z fazą implementacji (kodowania, programowania). Wykorzystując system operacyjny oraz możliwości języków programowania, wraz z takimi narzędziami jak edytor tekstowy czy system formatujący, można kontrolować różne wersje kodów źródłowych, zarządzać nimi oraz tworzyć odpowiedni system do produkcji dokumentacji.

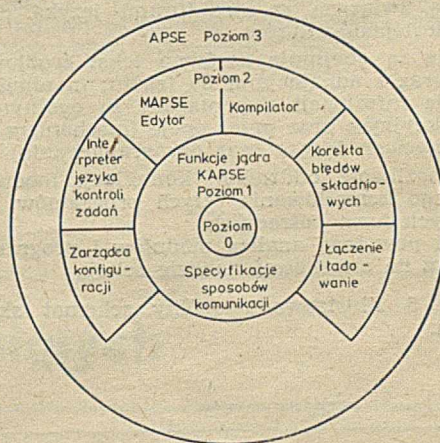
Szczególnie dobrym przykładem jest tu środowisko programistyczne oparte na systemie operacyjnym UNIX, jak również tzw. warsztat programisty powstały w oparciu o ten system operacyjny [8]. Wielką zaletą systemu UNIX jest możliwość łączenia procesów za pomocą mechanizmu rurowego (ang. pipe mechanism) oraz — koncepcja standardowego wejścia i wyjścia. System operacyjny nadzoruje i kontroluje buforowanie oraz czynności związane z wprowadzeniem standardowego wejścia pierwszego procesu jako standardowego wejścia do drugiego procesu. Ta właściwość daje możliwość łączenia wielu istniejących lub nowo przygotowanych narzędzi (programów) oraz budowania większych systemów w łatwy i wygodny sposób. Taka metoda „budowania z klocków” jest podstawą niezwykle efektywności i dużych możliwości rozważanego środowiska. Należy jednak przyznać, że system operacyjny UNIX został zbudowany z myślą raczej o doświadczonych i dobrze wykształconych programistach, którzy są zdolni wykorzystać go bardzo efektywnie, realizując dobrze zdefiniowane zadania techniczne.

Łatwość tworzenia i integracji narzędzi programistycznych i dokumentacyjnych na gruncie systemu UNIX jest widoczna m.in. na przykładzie systemu SOLID (systemu budowy systemów informatycznych działających w trybie bezpośrednim) [4]. Umożliwia on archiwizację wszystkich tekstów programów i dokumentów, jak również przekształca-

nie za pomocą generacji odpowiednich procedur źródłowych wszystkich rodzajów w produkty końcowe. Czynności faz przedimplementacyjnych, oparte o różne metodyki czy metodologie, muszą być przeprowadzone ręcznie i włączane w system w sposób zależny od osobistych preferencji, doświadczenia i poglądów użytkowników.

Analogiczne plany oparte o język ADA, znacznie wyższego poziomu, są dobrze znane pod kryptonimem STONEMAN [2]. Środowisko programistyczne ADY (APSE, Ada Programming Support Environment) ma być stworzone dla komputerów zanurzonych w bardziej złożony system techniczny. Ten fakt jest szczególnie interesujący np. dla współczesnych eksperymentów fizycznych oraz dla systemów kontroli cyfrowej akceleratorów, gdzie komputery są również tylko fragmentem większego systemu (środowiska) fizycznego. APSE powinno dostarczyć dobrze skoordynowanych zbiorów użytecznych narzędzi posiadających jednorodne sprzężenia oraz komunikujących się między sobą przez wspólną bazę danych. Baza ta służy jako źródło informacji oraz jako archiwum produktów końcowych dla wszystkich narzędzi.

Środowisko takie winno ułatwiać tworzenie i integrację nowych narzędzi, ich ulepszenie, modernizację oraz wymiennność. Powinno ono stwarzać użytkownikowi możliwość instruowania się i uzyskiwania pomocy, spełniając wymagania współpracy między człowiekiem a maszyną — zarówno w systemie interakcyjnym, jak i wsadowym. Komunikacja pomiędzy użytkownikiem a narzędziami winna być oparta o jednorodne konwencje. Jeśli tylko możliwe, koncepcje i pojęcia języka ADA winny być wykorzystywane przez APSE, tak by służył on również jako język poleceń.



Rys. 4. Poziomy zdefiniowania wewnątrz APSE

Wymienione cele można osiągnąć przez pokazany na rysunku 4 podział APSE na poziomy [2]:

- poziom 0:** sprzęt i oprogramowanie maszyny
- poziom 1:** jądro środowiska programistycznego ADY (KAPSE — Kernel APSE), które obsługuje bazę danych, komunikację i funkcje wspomagające w czasie wykonania programów (łącznie z MAPSE — Minimal APSE) oraz zapewnia sprzężenia niezależne od maszyny
- poziom 2:** minimalne środowisko programistyczne ADY (MAPSE), które zapewnia minimalny zestaw narzędzi wystarczający i koniecznych dla tworzenia i wspomagania programów; narzędzia te są napisane w języku ADA i wspomagane przez KAPSE
- poziom 3:** środowisko programistyczne ADY (APSE), które jest zbudowane przez rozwinięcie MAPSE dla uzyskania pełnego wspomagania szczególnych zastosowań lub szczególnej metodologii.

Model ten zapewnia spójną komunikację z użytkownikiem przez KAPSE, które z kolei określa współpracę z maszyną, na jakiej jest on zainstalowany. Dodatkowe narzędzia, napisane w języku ADA, można łatwo dołączyć, a następnie przenieść do innego APSE.

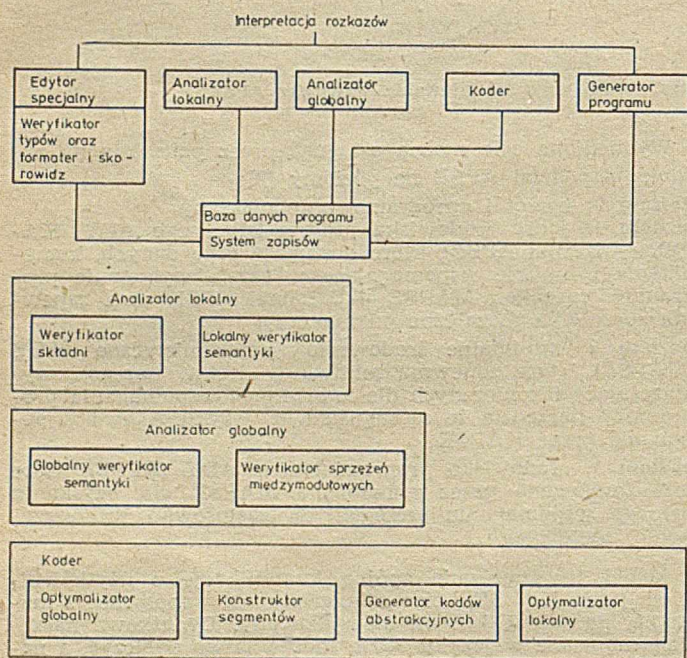
Tak więc KAPSE można traktować jako wirtualną maszynę dla programów napisanych w ADZIE, włącznie z narzędziami napisanymi w języku ADA. Zaprezentowany model ma zastosowanie przede wszystkim w fazie implementacji zadania programistycznego; zastosowania we wcześniejszych

niejszych fazach mogą być oparte wyłącznie na testach. Siła języka nie jest więc wykorzystywana w należyty stopniu. Niemniej pojęcie pakietu czy też modułu oraz rozróżnienie między ich specyfiką a implementacją umożliwia twórcy oprogramowania zrobienie przynajmniej kilku koncepcyjnych kroków w kierunku fazy projektu globalnego i szczegółowego. Możliwość generowania w tym języku elementów strukturalnych systemu (modułów) stwarza dodatkowe możliwości, które mogą być użyte do budowy nowych narzędzi, jak np. narzędzia komponowania i generacji w systemie GANDALF [6].

Zbliżonym podejściem, mającym na celu stworzenie środowiska programistycznego opartego na jednym języku, tym razem języku implementacji systemów oprogramowania CDL2, jest tzw. Laboratorium CDL2. Zostało ono opracowane na Politechnice Berlińskiej [3], zaś rozpowszechnione jest przez firmę EPSILON. Język i laboratorium mają szereg właściwości i cech, które są ważne z punktu widzenia niniejszej prezentacji:

- programy w CDL2 tworzy się przy użyciu hierarchii PROGRAM/MODUŁ/SEKCJA/PROCEDURA/WEZWIANIE/PARAMETR/OBIEKT
- sprzężenia EXPORT/IMPORT istnieją na poziomie modułu, a sprzężenia ROZWINIĘCIE/UOGÓLNIENIE na poziomie sekcji
- ten sam język służy jako język poleceń (bardziej złożone polecenia mogą być przedstawione jako procedury w CDL2)
- inicjowanie, tworzenie zapisów, redagowanie, kompilowanie oraz korygowanie programów zawarte jest w jednolitym procesie produkcyjnym, kontrolowanym wyłącznie przez jeden język wysokiego poziomu
- informacja zawarta w bazie danych tworzy strukturę hierarchiczną związaną ze strukturami języka oraz stopniowanym dostępem autorów
- informacja ta obejmuje zarówno tekst źródłowy, jak i wyszukiwanie wraz z adnotacjami do tekstu; manipulacja oraz wyszukiwanie informacji wspomagane są ogólnym mechanizmem selekcyjnym oraz przez opcje (mechanizm ten opiera się na pojęciach języka oraz oprogramowania)
- istnieje możliwość oddzielnej kompilacji modułów oraz zintegrowanego kompilowania całych programów, jak również dowolnych form pośrednich
- przegląd i powtórna analiza modułów i programów może być oparta na stopniowej analizie sekcji.

Na rysunku 5 zilustrowano ogólny schemat składników laboratorium.



Rys. 5. Ogólny schemat składników laboratorium CDL2

Przedstawione przykłady ilustrują podejście do problemu środowiska od strony implementacji (kodowania) i przedstawiają próby rozszerzenia zakresu ich wykorzystania w

kierunku zagadnień projektowych (ewentualnie w kierunku czynności weryfikacji i testowania). Konstrukcje języków programistycznych wysokiego poziomu umożliwiają przynajmniej częściową realizację tych rozszerzeń.

OGÓLNIJSZE UJĘCIA

Istnieje pogląd, że cały problem winien być zbadany w bardziej systematyczny sposób — zaczynając od definicji wymagań użytkowych i ich analizy, które powinny poddawać się obróbce komputerowej. Niesformalizowane określenie wymagań użytkowych w języku naturalnym wprowadza niejasności i umożliwia błędną interpretację. Warto tu wspomnieć o wczesnej próbie [1] stworzenia systemu, zwanego SAFE, który miał pozwalać na transformację niesformalizowanych specyfikacji w sformalizowane — za pomocą uzgadniania kontekstowego w warunkach interakcyjnego udziału użytkownika. Dalsze rezultaty tej próby nie są znane autorowi i najprawdopodobniej projekt ten został zaniechany.

Ogólne podejście polega więc na formułowaniu wymagań użytkowych w mniej lub bardziej sformalizowanym języku, ze specyficzną składnią i semantyką. Na to, by sformułować co ma być osiągnięte przez system informatyczny, trzeba zastosować jakiś rodzaj formalizmu modelującego, który pozwala opisać tę część „rzeczywistego świata” lub „wszechświata zdarzeń”, która jest przedmiotem zainteresowania. Podstawowym składnikiem są egzystensy albo byty (np. „rzeczy”, „procesy”) i ich właściwości (atrybuty) oraz relacje między nimi. W ten sposób utworzono tzw. model ERA (egzystensy, relacje, atrybuty) [5]. Model ten jest podstawą jednego z najbardziej udanych, pionierskiego podejścia do zagadnienia definiowania i analizy wymagań użytkowych, opracowanego przez D. Teichroewa i jego współpracowników na Uniwersytecie Michigan [10], znanego pod nazwą projekt ISDOS. Podstawowymi składnikami tego podejścia jest język opisu problemu PSL (Problem Statement Language), analizator opisu problemu PSA oraz baza danych.

Język PSL składa się z takich elementów (typów), jak „obiekty”, „relacje” i „właściwości”. Zależnie od liczby typów w ramach trzech składowych, można generować automatycznie (komputerowo) różne wersje PSL za pomocą tzw. systemu META (wersja ta nazywa się czasami językiem opisu systemu SXL według metodologii X). Wersje te lub metodologie mogą być użyte do szczególnych obszarów zastosowań (różnych części „rzeczywistego świata”).

Można stworzyć dowolną, lecz ustaloną liczbę typów obiektów, takich jak: wejście, wyjście, proces, procesor, zbiór, grupa, element itp. Użytkownik nadaje szczególnym przypadkom takich obiektów nazwy według własnego uznania. Obiekty te mogą być powiązane między sobą przez różne relacje, których liczba i rodzaj muszą być ustalone. Opisują one różne aspekty systemu przetwarzania informacji, takie jak: granice systemu („odbiera”, „generuje”), struktury systemu („część czego”), struktury danych („zawiera się w”, „identyfikuje”), wyprowadzenie danych („wykorzystuje”, „wyprowadza”, „uaktualnia”), wielkości („wartości”) oraz jego dynamiki („uruchamia”), itp. Dozwolone relacje między obiektami zależą od ich typów. W konsekwencji można wykryć automatycznie wiele niezgodności tego rodzaju. Ponieważ większość relacji zachodzi w obu kierunkach (relacje binarne), możliwe jest produkowanie na żądanie takich list dla dowolnego obiektu. Daje to użytkownikowi możliwość uzyskania spisu wszystkich obiektów, które mogą podlegać zmianom — w przypadku gdy określony obiekt zostanie zmodyfikowany. Każdy obiekt można opisać przez nadanie wartości różnym jego atrybutom (właściwościom). Do nich należą synonimy, słowa kluczowe i opisy (dowolne teksty służące do opisu obiektu). Można też definiować nowe właściwości (atrybuty).

Analizator PSA analizuje stwierdzenia w języku PSL umieszczone w bazie danych i tworzy znaczną liczbę różnego rodzaju analiz, statystyk i raportów, takich jak:

- sformatowane przedstawienie problemu
- skrowidze i indeksy słów kluczowych
- opisy struktur hierarchicznych
- graficzne przedstawienia przepływów i relacji
- podsumowania statystyczne.

Niektóre z tych raportów nadają się dobrze do wspomaganie czynności projektowych i analitycznych, wykraczając poza zakres wymagań użytkowych (ich definiowania i analizy).

Stwierdzenia w języku PSL można wprowadzać do bazy danych stopniowo. Po ich wprowadzeniu, PSA może dokonać analizy poprawności i zgodności aktualnej zawartości bazy danych. W każdej chwili analizator PSA może dokonać analizy całości lub tylko wybranej części danych. Dane analizowane i przedstawiane w dowolnym raporcie mogą być wprowadzone do bazy danych przez różnych analityków i w różnym czasie. Jest to szczególnie ważne z punktu widzenia kolektywnego wykorzystywania PSL/PSA przez zespół analityków i programistów. W PSA stosuje się wyspecyfikowany zbiór poleceń za pomocą których można formułować różne zapytania w celu produkcji raportów, analiz i zestawień użytecznych dla twórców systemu oraz nadzoru ich pracy w zespole.

Podsumowując, PSL/PSA stwarza możliwość sformalizowanego przedstawiania stwierdzeń dotyczących struktury i funkcjonalności projektowanego systemu oraz daje zestaw cennych narzędzi do ich analizy i przedstawienia jej warunków. Wymaga to szkolenia i nabycia odpowiednich umiejętności, lecz ostateczne rezultaty sownie wynagradzają ten trud.

Niedopracowane są zagadnienia opisu parametrów użytkowych. Jest to w chwili obecnej najtrudniejszy aspekt, wymagający uwzględnienia przy rozpatrywaniu zagadnień projektowych systemów informatycznych. Projekt ISDOS proponuje sposób częściowego rozwiązania tego problemu. W oparciu o zawartość bazy danych, automatyczny generator może generować pakiet symulacyjny, który służy do szacunkowego porównania uzyskanych parametrów użytkowych ze stawianymi wymaganiami. Wersja PSL, którą można użyć dla tego celu, nazywa się czasami DSL (Dynamiczny Język Stwierżeń) [11].

Pierwotna wersja tego artykułu została przedstawiona na konferencji „Zastosowanie komputerów w projektowaniu i eksploatacji akceleratorów”, 20–23 września 1983, w Berlinie Zachodnim.

JERZY SZYLLER

Centrum Informatyczne
Uniwersytet Warszawski

LITERATURA

- [1] Balzer R., Goldamn N., Wile D.: Informality in Program Specifications. IEEE Trans. on Software Engineering, SE-4, p. 94, 1978
- [2] Buxton J. N., Druffel L. E.: Requirements for an ADA programming support environment: rationale for STONEMAN. W: Software Engineering Environments, North-Holland, 1981
- [3] Bayer M., Böringer B., Dehottay J. P., Feuerhahn H., Jasper J., Koster C.H.A., Schmiedecke U.: Software Development in the CDL2 Laboratory. W: Software Engineering Environments, North-Holland 1981
- [4] Bianchi M. H., Glushko R. J., Mashey J. R.: A software/documentation environment built from the UNIX toolkit. W: Automated tools for information system design, North-Holland, 1982
- [5] Chen P. P.: The entity-relationship model — toward a unified view of data. ACM Transactions on Database Systems. Vol. 1, 9, 1976
- [6] Haberman A. N., Perry D. E.: System Composition and Version Control for ADA. W: Software Engineering Environments, North-Holland 1981
- [7] Howden W. E.: Contemporary Software Development Environments. Comm. of the ACM, Vol. 25, 318, 1982
- [8] Mitze R. W.: The UNIX-System as a software engineering environment. W: Software Engineering Environments, North-Holland 1981
- [9] Osterweil L. J.: A software lifecycle methodology and tool support. W: Software Development Tools, Springer Verlag, 1980
- [10] Teichroew D., Herhey E. A.: PSL/PSA: A computer-aided technique for structured documentation and analysis of information processing systems. IEEE Trans. on Software Engineering, Vol. SE-3, p. 41, 1977
- [11] Teichroew D., Spewak S., Herhey III E. A., Yamamoto Y., Starner G.: Computer-aided Modelling of Information Systems, Proc. CompSoc 79
- [12] Wasserman A. I.: Automated tools in the information system development environment. W: Automated Tools for Information Systems Design, North-Holland 1982.

Mikroprocesory lat osiemdziesiątych

Na przełomie lat sześćdziesiątych i siedemdziesiątych, na skutek znacznego udoskonalenia półprzewodnikowych technologii unipolarnych, powstały możliwości wytwarzania układów zawierających ponad tysiąc tranzystorów na jednej strukturze. Wynikiem miniaturyzacji półprzewodnikowych układów elektronicznych było powstanie w 1971 roku w firmie INTEL pierwszego mikroprocesora oznaczonego symbolem 4004. Był to układ wykonany w technologii p-MOS, zawierający cały procesor 4-bitowego systemu komputerowego. Mikroprocesorowi składającemu się z 2300 tranzystorów towarzyszyły trzy inne mikroukłady: 4001 — zawierający 256 bajtów pamięci ROM oraz 4-bitowy port

wejścia-wyjścia, 4002 — zawierający cztery 20-cyfrowe rejestry oraz 4-bitowy port wyjściowy i 4003 — zawierający 10-bitowy rejestr przesuwany, do którego można wprowadzać i wyprowadzać dane szeregowo oraz wyprowadzać równoległe całą zawartość.

Od pojawienia się pierwszego mikroprocesora rozpoczął się trwający do dzisiaj wyścig, polegający na konstruowaniu coraz lepszych i szybszych układów procesora oraz układów pomocniczych. W połowie lat siedemdziesiątych największą popularność na świecie uzyskał 8-bitowy mikroprocesor firmy INTEL oznaczony symbolem 8080. Stał się on konstrukcją wzorcową powielaną następnie przez wielu innych producentów (w tym nasze CEMI), jak również ulepszaną dalej przez różne firmy (np. Z-80 firmy ZILOG).

Podstawowe cechy i uwarunkowania architektury mikroprocesorów połowy lat siedemdziesiątych, wykonywanych głównie w technologii n-MOS, umożliwiającej upakowanie od 5 do 10 tysięcy tranzystorów na jednej strukturze, są następujące:

- długość słowa 4 lub 8 bitów (1 bajt)
- lista rozkazów zwykle obejmująca nie więcej niż 100 rozkazów
- rozkazy o długości nie przekraczającej 3 bajtów
- liczne i zręczne metody adresowania, mające na celu skrócenie części adresowej rozkazu
- istnienie wielu rejestrów bazowych, indeksowych, roboczych i akumulatorów
- niewielka liczba rozkazów systemowych
- istnienie rejestru wskaźnika stosu, umożliwiającego korzystanie ze stosu umieszczonego w pamięci

Mgr inż. JERZY SZYLLER jest absolwentem Wydziału Elektroniki Politechniki Warszawskiej (1969 r.). W początkowym okresie pracy zawodowej zajmował się projektowaniem i wdrażaniem oprogramowania systemowego dla komputerów ODRA 1204, MERA 300 i PDP11 (SM-4). W latach 1976–1979 w Instytucie Technologii Elektronicznej zajmował się analizą zagranicznych systemów mikroprocesorowych, pod kątem przygotowania konstrukcji krajowej. Od 1979 roku pracuje w Centrum Informatycznym Uniwersytetu Warszawskiego na stanowisku Głównego Specjalisty, prowadząc prace z zakresu oprogramowania systemowego mini- i mikrokomputerów.



- możliwość wykonywania prostych operacji arytmetycznych i logicznych na argumentach jedno- lub dwubajtowych
- istnienie mechanizmów umożliwiających reakcję na 1—4 sygnały przerwań
- możliwość adresowania pamięci o pojemności nie większej niż 64 KB
- istnienie prostych mechanizmów blokowania i odcinania procesora od szyn zewnętrznych
- istnienie wielu specjalizowanych mikroukładów otaczających mikroprocesor.

Po roku 1975 można zanotować następną przełom w rozwoju technologii półprzewodnikowych, którego efektem jest powstanie wielu zaawansowanych konstrukcji mikroprocesorowych, w tym pierwszego jednoukładowego mikroprocesora 8-bitowego. Liczba tranzystorów w tych mikroukładach osiągnęła wówczas poziom kilkunastu tysięcy.

Przełom lat siedemdziesiątych i osiemdziesiątych to okres nowego przyspieszenia technologicznego, które zaowocowało szeregiem konstrukcji mikroprocesorowych 16- i 32-bitowych o architekturze znacznie odbiegającej od wypracowanej podczas projektowania układów 8-bitowych. W tym okresie powstały najbardziej znane mikroprocesory, takie jak: INTEL 8086, Z8000, MC 68000, NS 16000 oraz najbardziej zaawansowane technologicznie, jak — TMS 320, iAPX 186, iAPX286, iAPX 432, NCR/32, HP 32 czy BELL-MAC 32. Architektura tych mikroprocesorów, zwłaszcza 32-bitowych, w których liczba tranzystorów sięga kilkuset tysięcy, uległa znacznej ewolucji, tak by gwarantowały one dużo większą moc obliczeniową i uniwersalność zastosowań.

CECHY ARCHITEKTURY

Obecnie konstrukcja mikroukładów elektronicznych rozwija się w dwóch kierunkach.

Pierwszy polega na projektowaniu i wytwarzaniu specjalizowanych układów w seriach o różnej długości na zamówienie klientów. Tę możliwość osiągnięto przez znaczne ulepszenie metod projektowania wspomaganego komputerowo (ang. Computer Aided Design — CAD). Efektem tej działalności jest konstruowanie szeregu specjalizowanych mikroprocesorów 4- i 8-bitowych, stosowanych np. w sprzęcie powszechnego użytku.

Drugi kierunek polega na konstruowaniu 16- i 32-bitowych mikroprocesorów o dużej mocy obliczeniowej, stosowanych jako elementy średnich i dużych systemów komputerowych różnego przeznaczenia, poczynając od komputerów osobistych, a kończąc na sieciach komputerowych i telekomunikacyjnych. Poniżej określono podstawowe cechy charakteryzujące mikroprocesory nowej generacji, cechy które będą zapewne występować również w przyszłych rozwiązaniach, stanowiąc o ich możliwościach użytkowych.

Brak wymienności programowej

Przy projektowaniu kolejnych mikroprocesorów konstruktorzy starali się, aby programy poprzednika, po naniесieniu jak najmniejszej liczby poprawek (a jeszcze lepiej bez żadnych korekt), były akceptowane przez procesor następny. Z punktu widzenia użytkownika jest to zupełnie oczywiste. Tylko najsilniejsze firmy mogły sobie pozwolić na opracowanie całkiem oryginalnego mikroprocesora. W 1976 roku w ten sposób postąpiła firma INTEL wprowadzając na rynek rodzinę mikrokomputerów jednoukładowych typu 8048, 8049 i 8021, które realizowały listę rozkazów zupełnie odmienną od mikroprocesorów rodziny 8080. Firma INTEL miała jednak powody, aby tak postępować; uniwersalny mikroprocesor typu 8080 z funkcjonalnego i ekonomicznego punktu widzenia nie nadawał się bowiem do tworzenia niewielkich, minimalnoulkowych sterowników.

Wszyscy konstruktorzy zakładali natomiast, że okleiny mikroprocesor musi współpracować z poprzednio wytwarzanymi mikroukładami wejścia-wyjścia i pomocniczymi. Takie wymagania są stawiane również większości obecnie produkowanych mikroprocesorów.

Wymiennosc oprogramowania współczesnych mikroprocesorów z poprzednikami zwykle nie jest zachowywana, ponieważ ich architektura (a więc i lista rozkazów, która jest jej odzwierciedleniem) musi gwarantować lepszą sprawność przetwarzania. Jednym z parametrów charakteryzujących sprawność przetwarzania jest zwiezlość programów. Ten zaś parametr jest wprost uzależniony od symetrii architektury mikroprocesora.

Symetria architektury

Realizacja postulatów symetrii architektury oznacza dążenie do osiągnięcia stanu, w którym każdy rozkaz mikroprocesora (o ile jest związany z daną) może operować danymi źródłowymi dowolnego typu lub tworzyć wynik dowolnego typu przy użyciu dowolnego sposobu adresowania.

Mikroprocesory 4- i 8-bitowe zwykle nie charakteryzowały się symetrią architektury, ponieważ ich konstruktorzy starali się optymalizować wykorzystanie zadanej z góry powierzchni mikroukładu, unikając przede wszystkim wszelkich redundancji układowych. Na przełomie lat siedemdziesiątych i osiemdziesiątych, wraz z dalszym doskonaleniem technologii, ta sytuacja ulega zmianie. Możliwość umieszczania na strukturze kilkudziesięciu tysięcy tranzystorów pozwala na pewną „rozzrutność” podczas procesu projektowego, z drugiej zaś strony stwarza poważne problemy projektowe, jeśli działanie układu nie jest sterowane mikroprogramem. Dlatego większość obecnie wytwarzanych mikroprocesorów ma sterowanie realizowane na zasadzie mikroprogramowej, które umożliwia znaczną symetryzację architektury.

Znakomitymi przykładami mikroprocesorów charakteryzujących się dużym stopniem symetryzacji architektury są NS 16000 (NATIONAL SEMICONDUCTOR) oraz MC 68000 (MOTOROLA). Możliwość rozbudowy układów sterowania mikroprocesora owocuje w szereg rozwiązań, które zwiększają szybkość działania mikroprocesora oraz mogą znacznie usprawnić pracę systemów operacyjnych. Poniżej zostaną omówione najciekawsze, choć nie nowe koncepcje stosowane w mikroprocesorach lat osiemdziesiątych.

Sterowanie „z wyprzedzeniem”

Jest to pomysł, który po raz pierwszy zastosowali w roku 1961 konstruktorzy firmy IBM w maszynie STRETCH. Polega on na pobieraniu rozkazów z wyprzedzeniem i ustawianiu ich w kolejce, gdy procesor nie kontaktuje się z pamięcią operacyjną. W ten sposób procesor może wykonywać jeden rozkaz, dekodować następny i w tym samym czasie pobierać kolejny z pamięci. Dzięki takiemu zabiegowi można zwiększyć szybkość działania mikroprocesora nawet dwukrotnie. Sterowanie „z wyprzedzeniem” zastosowali konstruktorzy mikroprocesorów MC 68000 i INTEL 8086. Możliwość pobierania kilku kolejnych bajtów z pamięci znakomicie usprawnia (zwiększając nawet do 90%) wykorzystanie szyny danych systemu mikrokomputerowego. Jednostka sterująca mikroprocesora składa się w tym wypadku z dwóch bloków: pobierającego i ustawiającego w kolejce szereg bajtów z pamięci oraz interpretującego znaczenie kolejnych bajtów.

Rozbudowa mechanizmów systemowych

Większość złożonych architektonicznie mikroprocesorów ma dwa wyróżnione stany działania — systemowy i użytkowy. Wyróżnienie dwóch stanów procesora znacznie ułatwia projektowanie oprogramowania systemowego, które jest wtedy wyraźnie i logicznie oddzielone od warstwy użytkowej. Wykonywanie szeregu rozkazów, których niewłaściwe użycie w programie mogłoby spowodować zniszczenie systemu, jest niemożliwe w stanie użytkowym. Wykonanie jednego z takich nielegalnych rozkazów powoduje — jak w tradycyjnych systemach komputerowych — generację sygnału przerwania i wejście procesora w stan systemowy. Również dostęp do pewnych rejestrów procesora lub mikroukładów pomocniczych może być chroniony, gdy procesor znajduje się w stanie użytkowym. Jednym z takich rejestrów są rejestry ochrony pamięci, których zawartość można zmieniać tylko w stanie systemowym.

Mikroprocesory lat siedemdziesiątych miały możliwość adresowania co najwyżej 64 KB pamięci, która była stosunkowo drogim elementem systemu. Obecnie sytuacja ulega całkowitej zmianie ze względu na ciągły spadek cen układów pamięciowych (256 KB kosztuje mniej niż 500 dol.) oraz niezwykle powiększenie się zakresu adresowania (1 MB — 4 GP). Tak znaczny wzrost możliwości adresowania pozwala na tworzenie dużych programów, choć mimo wszystko pojemność fizycznej pamięci w systemie jest nadal ograniczona jej kosztem i nigdy nie jest maksymalna. Znaną koncepcją pamięci wirtualnej umożliwiająca rozwiązanie konfliktu pomiędzy długością programu a nie wystarczającą wielkością pamięci operacyjnej. Obecnie wszystkie liczące się firmy produkujące mikroprocesory wprowadzają na rynek mikroukłady umożliwiające zarządzanie pamięcią. Są to specjalizowane układy współpracujące z odpowiednimi

mikroprocesorami, MC 68451, Z-8010 i Z8015 oraz NS 16082, lub mikroprocesory z wybudowanym mechanizmem translacji adresu wirtualnego na adres fizyczny, jak np. iAPX 286 firmy INTEL.

Zastosowanie praktyczne pamięci wirtualnej skutecznie eliminuje wymieniony konflikt i znakomicie wpływa na rozszerzenie możliwości systemu mikrokomputerowego. Pamięć dla wykonywanego programu, niezależnie od jego wielkości jest przydzielana automatycznie bez ingerencji programisty i oceny wzajemnej wielkości pamięci oraz programu przed jego wykonaniem. Ponadto, program nie wymaga stosowania mechanizmów ręcznego nakładkowania, a mechanizm pamięci wirtualnej umożliwia efektywne tworzenie środowiska dla systemów wieloprogramowych i wielodostępnych.

Rozbudowa arytmometru

Mikroprocesory lat siedemdziesiątych, zwłaszcza 4- i 8-bitowe, były wyposażone w bardzo proste arytmometry, uniemożliwiające wykonywanie złożonych operacji arytmetycznych i logicznych. Regułą był brak operacji mnożenia i dzielenia. Obecnie problemy zwiększenia mocy obliczeniowej mikroprocesorów rozwiązuje się dwoma sposobami. Pierwszy polega na rozbudowie arytmometru, który znajduje się w ramach mikroprocesora. W ten sposób postąpiła firma MOTOROLA, zaopatrując mikroprocesor MC 68000 w trzy arytmometry używane również do niezależnego obliczania adresu. Drugi sposób wykorzystaly firmy INTEL i NATIONAL SEMICONDUCTOR, które wyprodukowały specjalizowane procesory arytmetyczne realizujące operacje stało- i zmiennoprzecinkowe według obowiązującego standardu IEEE. Procesory typu 8087 i NS 16081 współpracują na zasadzie urządzeń autonomicznych z 16-bitowymi mikroprocesorami firmy INTEL oraz mikroprocesorami rodziny NS 16000 firmy NATIONAL SEMICONDUCTOR, wykonując np. dzielenie 64-bitowych liczb zmiennoprzecinkowych w czasie ok. 23 us.

Dystrybucja funkcji systemowych

Początkowo, ze względu na liczbę elementów tranzystorowych sięgającą kilku tysięcy w jednym mikroukładzie oraz niewielką liczbę końcówek układu, sam mikroprocesor nie mógł realizować wielu funkcji systemowych. Był więc otaczany różnymi mikroukładami, które realizowały obsługę przerwań czy współpracę z urządzeniami wejścia-wyjścia. W miarę wzrostu liczby elementów w mikroukładach — równoległe ze wzrostem złożoności procesorów — wzrastała również złożoność układów pomocniczych.

W drugiej połowie lat siedemdziesiątych firma INTEL zdecydowała się wyprodukować mikroukład typu 8041, który służył jako autonomiczny procesor do różnych celów, głównie do realizacji funkcji wejścia-wyjścia. Wówczas niezbędne stało się wyposażenie takich mikroprocesorów w mechanizmy umożliwiające efektywną współpracę z głównym procesorem w systemach wieloprocesorowych. Mikroprocesor 16-bitowy typu 8086 ma taki mechanizm, umożliwiający realizację funkcji TEST AND SET i co za tym idzie — organizację bezkolizyjnego dostępu wielu procesorów do wspólnej pamięci. Dystrybucja funkcji systemowych, ze względu na zmniejszający się koszt mikroprocesorów, jest coraz szerzej praktykowana w projektowaniu współczesnych systemów mikrokomputerowych.

PERSPEKTYWY ROZWOJU ARCHITEKTURY

Podobnie jak w poprzednim dziesięcioleciu, tak i obecnie producenci mikroukładów starają się pogodzić dwa sprzeczne kierunki — minimalizację kosztów wytwarzania z równoczesną maksymalizacją złożoności mikroprocesora. Firmie INTEL udało się ponownie jako jednej z pierwszych pokonać tę sprzeczność i wyprodukować dwa systemy 80186 (iAPX186) oraz 80286 (iAPX286), które w jednym układzie mikroprocesora zawierają elementy tworzące do tej pory złożone pakiety mikrokomputerowe. Zabieg maksymalnego obniżenia ceny udał się zwłaszcza w odniesieniu do mikroprocesora typu 80186, natomiast znacznie bardziej zaawansowany technologicznie układ typu 80286, zawierający 130 tys. tranzystorów jest jeszcze stosunkowo drogi. W tabeli 1 przedstawiono porównanie kosztów i liczby elementów systemów mikrokomputerowych podobnej klasy, opartych na mikroprocesorach typu 80186, 8086 oraz 8085. Tabela 2 natomiast obrazuje zależności między szybkością (liczbą rozkazów wykonywanych w ciągu jednej sekundy) a kosztem odpowiednich mikroprocesorów.

Tabela 1. Porównanie kosztów i liczby elementów systemów mikrokomputerowych opartych na mikroprocesorach typu 80186, 8086 i 8085 (według miesięcznika BYTE, kwiecień 1983)

Użyte elementy elektroniczne	Cena (w dolarach)
System IAPX 186 (przy zamówieniu 10 tys. sztuk)	
80186	30,00
8282 (dwa bufory)	5,90
Razem	35,90
Odpowiadający pakiet procesora 16-bitowego	
8086 (5 MHz)	19,15
8237A-5 + 8212 (sterownik DMA)	15,30
8259A (sterownik przerwań)	3,75
8253 (czasomierz)	3,40
8282 (dwa bufory)	5,90
8286 (dwa układy przejścia)	5,90
Dekodery	3,50
Generator stanu WAIT	3,50
8288 (sterownik szyny)	7,95
8284A (generator sygnałów taktujących)	3,15
Razem	71,50
Odpowiadający pakiet procesora 8-bitowego	
8085 (3 MHz)	2,65
8259A (sterownik przerwań)	3,75
8237 + 8212 (sterownik DMA)	8,80
8253 (czasomierz)	3,40
8282 (bufor)	2,95
8286 (układ przejścia)	2,95
Dekodery	1,75
Generator stanu WAIT	3,50
Razem	29,75

Tabela 2. Zależności pomiędzy szybkością (liczba rozkazów wykonywanych w ciągu jednej sekundy — MIPS) a kosztem odpowiednich mikroprocesorów (w dolarach) — według miesięcznika BYTE, kwiecień 1983

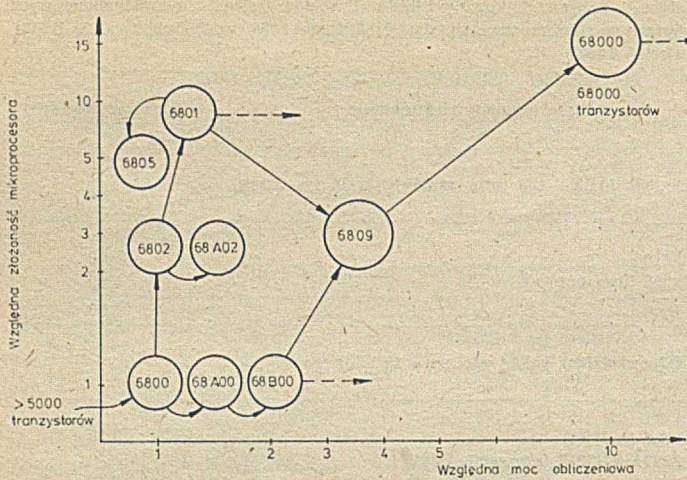
Typ procesora	8085	8086	80186	80286
Szybkość przetwarzania (MIPS)	0,07	0,3	0,7	1,5
Stosunek kosztów do szybkości (MIPS \$)	700	1500	10 000	7500

Niezależnie od starań, mających na celu uzyskanie najkorzystniejszego stosunku szybkości do ceny mikroprocesora, wszystkie liczące się firmy usiłują, bez względu na koszt, wyprodukować — korzystając z zawrotnych postępów technologii — jak najszybszy i najwszechstronniejszy mikroprocesor. Na rysunku 1 przedstawiono ewolucję mikroprocesorów firmy MOTOROLA, której najbardziej zaawansowany mikroprocesor 16-bitowy MC 68000 (z rejestrami 32-bitowymi) składa się z 68 tys. tranzystorów.

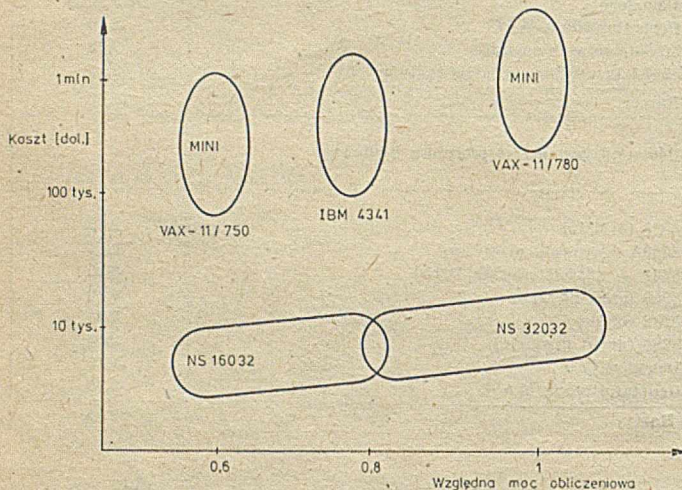
Rysunek 2 przedstawia względne porównanie mocy obliczeniowej mikroprocesorów najnowszej rodziny NS 16000 firmy NATIONAL SEMICONDUCTOR z powszechnie znanymi komputerami firm IBM i DEC.

Ponieważ rozwój technologii przekracza obecnie najsmielsze oszacowania specjalistów — zarówno w zakresie osiągniętej prędkości przełączania poszczególnych bramek logicznych, należy założyć, że mikroprocesory złożone z 1 mln tranzystorów i mające cykl rozkazowy o długości 100 ns nie powinny w drugiej połowie lat osiemdziesiątych należeć do rzadkości.

Możliwości technologiczne będą skłaniały konstruktorów układów mikroprocesorowych do podejmowania eksperymentów w kilku podstawowych, przedstawionych poniżej, kierunkach architektonicznych.



Rys. 1. Ewolucja mikroprocesorów firmy MOTOROLA



Rys. 2. Względne porównanie mocy mikroprocesorów rodziny NS 16000 ze znanymi komputerami firmy IBM i DEC (wg BYTE, kwiecień 1983)

Pierwszy z nich polega na układowym realizowaniu funkcji systemowych i użytkowych, które do tej pory były realizowane na drodze programowej. W najnowszych mikroprocesorach 32-bitowych, takich jak NCR/32, NS 32032,

HP32, BELLMAC 32 czy iAPX 432, znajdujemy wiele funkcji systemu operacyjnego (często odpowiadającego systemowi UNIX) oraz elementów języków wysokiego poziomu (np. języka C) realizowanych sprzętowo. Moc obliczeniowa wymienionych mikroprocesorów 32-bitowych jest porównywalna z mocą procesorów systemu 370 firmy IBM. Przykładowo — mikroprocesor NCR/32 (cztery razy szybszy od MC 68000) może być wykonany w wersji zawierającej zestaw mikrorozkazów, umożliwiających emulowanie procesora IBM 370/138.

Drugim sposobem wykorzystania szybkości i mocy systemów mikrokomputerowych jest przydzielanie różnych funkcji systemowych i użytkowych wielu mikroprocesorom i łączenie ich w jeden system wieloprocessorowy. Najczęściej autonomiczne procesory spełniają rolę specjalizowanych sterowników różnego rodzaju urządzeń, zwłaszcza urządzeń wejścia-wyjścia oraz jednostek arytmetycznych. Dobry przykład mogą tu stanowić rozmaite mikroprocesory firmy INTEL, poczynając od procesora typu 8041, przez 8089, a kończąc na 8087. Podstawowym problemem, który wymaga rozwiązania jest zapatrzenie mikroprocesorów używanych do tworzenia systemów wieloprocessorowych w mechanizm umożliwiające współpracę wielu procesorów przez wspólną szynę i pamięć. Istnienie sygnału LOCK w mikroprocesorze typu 8086/88, który blokuje dostęp do szyny systemowej podczas jednego cyklu rozkazowego umożliwia organizowanie semaforów, a zatem synchronizację dostępu wielu procesorów do wspólnych zasobów różnego rodzaju. Wydaje się, że intensywny rozwój obu wymienionych kierunków umożliwi w niedługim czasie efektywną dystrybucję funkcji systemu komputerowego na poziomie użytkowym to nie tylko w układach wieloprocessorowych, ale także sieciowych.

Trzecim kierunkiem, który nieśmiało zapoczątkował pojawienie się mikroprocesora TMS320 firmy TEXAS INSTRUMENTS, jest projektowanie i wdrażanie układów o architekturze odmiennej od tradycyjnej. Prosty zabieg, polegający na fizycznym rozdzieleniu pamięci z danymi od pamięci z programami oraz szyn, którymi przesyłane są dane i rozkazy, umożliwił znaczne przyspieszenie pracy mikroprocesora (architektura typu HARVARD). Zrównoleżenie cykli pobierania rozkazów i danych, przy podstawowym mikrocyklu 200 ns, umożliwia wykonywanie ok. 5 mln rozkazów/s. Układ ten stosuje się w telekomunikacji, do analizy i syntezy mowy, rozpoznawania obrazów lub obliczania transformat Fouriera. Dążenie do maksymalnej równoległości działania poszczególnych bloków funkcjonalnych mikroprocesora, a następnie elementów mikrokomputera, będzie niewątpliwie stymulować odchodzenie od tradycyjnej architektury. Pewne nadzieje można wiązać — na przykład — z wykorzystaniem koncepcji sterowania przepływem danych (ang. data flow). Oczywiście, zwiększanie liczby elementów w jednym układzie nie może przebiegać w sposób nieograniczony, dlatego też prace nad nietypowymi rozwiązaniami architektonicznymi będą nabierały w przyszłości coraz większego znaczenia.

Kalendarz 1985

Prowadziliśmy do niedawna kalendarz imprez informatycznych, korzystając przede wszystkim z doniesień pism fachowych. Informacje te jednak nie dopływają regularnie — nie uzyskiwaliśmy kompletnych danych. Przeciwnie nawet — były zbyt chaotyczne, by taki kalendarz prowadzić dalej. Zmieniamy więc formułę. Będziemy odtąd zapowiadając tylko te imprezy, o których zostaniemy poinformowani bezpośrednio przez organizatorów. Podajemy tylko główne informacje — szczegółowe otrzymać można w redakcji.

WIOSNA

- **HARDWARE AND SOFTWARE COMPONENTS AND ARCHITECTURES FOR THE 5th GENERATION** — Paryż, 5-7 marca, organizator: AFCET INFORMATIQUE
- **MIKRONIKA-85**, Postęp w budowie precyzyjnego sprzętu elektroniczno-mechanicznego — Warszawa, 15-16 maja, organizatorzy: SIMP, Politechnika Warszawska
- **SM 85** — Warszawa, 21-24 maja, organizator: Klub Użytkowników Minikomputerów SM, Polskie Towarzystwo Informatyczne

- **PROLAMAT 1985**, Software for Discrete Manufacturing — Paryż, 11-13 czerwca, organizator: AFCET

LATO

- **IX KRAJOWA KONFERENCJA AUTOMATYKI**, Stan i perspektywy automatyki w Polsce — Łódź, koniec czerwca, organizatorzy: Polski Komitet Pomiarów i Automatyki NOT, Politechnika Łódzka
- **SIGGRAPH 85**, The Twelfth Annual Conference on Computer Graphics and Interactive Techniques — San Francisco, 22-26 lipca, organizator: ACM
- **WCCE-85**, World Conference on Computers in Education — Norfolk, Virginia, USA, 29 lipca — 2 sierpnia, organizatorzy: IFIP/TC-3, AFIPS
- **EUROMICRO 85**, Eleventh Symposium on Microprocessing and Microprogramming; Microcomputers, usage and design — Bruksela, 3-6 września, organizator: EUROMICRO
- **ZASTOSOWANIE KOMPUTERÓW W PRZEMYSLE**, IV konferencja naukowo-techniczna — Szczecin, 11-13 września, organizatorzy: OW NOT, Politechnika Szczecińska

CHILL – język programowania systemów komutacyjnych (2)

Budowa programu i wykonanie współbieżne

Język CHILL, aczkolwiek projektowany z myślą o systemach komutacyjnych, jest wystarczająco uniwersalny, by służyć do programowania szerokiej klasy systemów działających w trybie uwarunkowanym czasowo. Przy projektowaniu dużych systemów czasu rzeczywistego szczególnie mocno odczuwana jest potrzeba tworzenia właściwej struktury programu. Dlatego projektanci języka wiele uwagi poświęcili opracowaniu konstrukcji oraz strukturalizacji programu.

Specyfika pracy systemów komutacyjnych zmusza do programowania w warunkach współbieżności. W języku CHILL wprowadzono pojęcie procesu jako jednostki programu przetwarzanej współbieżnie i zastosowano bogaty zasób mechanizmów współpracy procesów. Konstrukcje języka są również na tyle ogólne, aby mogły znaleźć zastosowanie do programowania szeroko rozumianych systemów przetwarzania współbieżnego.

STRUKTURA PROGRAMU

Dostępny w CHILLU konstrukcjami strukturalizacji programu są: moduły, rejony i bloki. Dla ograniczenia widoczności (ang. visibility) i okresu istnienia (ang. lifetime)¹⁾ obiektów, procedury i procesy są traktowane jak bloki, podobnie jak instrukcja **DO**, każdy człon alternatywy w instrukcji **RECEIVE CASE** oraz programy obsługi wyjątków (ang. exception handlers). Program w CHILLU składa się z sekwencji modułów i rejonów.

MODUŁY

Moduły są ograniczone nawiasami **MODULE** i **END**. Nazwy definiowane w module są lokalne w tym module, zaś nazwy globalne, zdefiniowane na zewnątrz modułu, nie są automatycznie widoczne w tym module. Aby globalną nazwę uczynić widoczną w module, należy wymienić ją w deklaracji dzierżawienia **SEIZE**. Aby uczynić nazwę dostępną na zewnątrz modułu, należy wymienić ją w deklaracji udostępnienia **GRANT**.

```
M1: MODULE
    DCL D INT;
M2: MODULE
    DCL C, E, F, INT;
    SEIZE D;
    GRANT C, E; (* C, D, E i F są tu widoczne *)
END M2;
(* C, D i E są tu widoczne *)
M3: MODULE
    DCL G BOOL;
    SEIZE E;
    (* E i G są tu widoczne *)
END M3;
B1: BEGIN
    DCL D BOOL;
    (* C i E są tu widoczne *)
END B1;
END M1;
```

W powyższym przykładzie wszystkie obiekty z modułów M2 i M3 mają okres istnienia rozciągnięty na cały program (przykład). Deklaracje udostępniania i dzierżawienia mają kilka dodatkowych możliwości: **GRANT ALL**; jest to skrócony zapis udostępnienia wszystkich nazw dobrze widocz-

nych²⁾ (ang. strongly visible) w tym module. Nazwa może być udostępniana z atrybutem **PERVASIVE**, np. **GRANT X PERVASIVE**; i wtedy staje się ona widoczna w wewnętrznych modułach programu bez użycia w nich deklaracji dzierżawienia.

Przy udostępnianiu nazwy zmiennej będącej strukturą, istnieje możliwość zasłaniania wybranych pól tej struktury przez użycie atrybutu **FOBRID**, np. **GRANT NODE FORBID (PRED, SUC)**; Atrybut **FORBID ALL**; przesłania wszystkie pola rekordu.

Zapis **SEIZE ALL**; powoduje dzierżawienie wszystkich nazw, które są dobrze widoczne w otaczającym bloku lub module, zaś zapis **SEIZE** (nazwa modułu) **ALL**; oznacza, że dzierżawione są te nazwy, które są widoczne i udostępnione przez wyspecyfikowany moduł.

Ogólnie można stwierdzić, że moduł jest środkiem do kontrolowania widoczności, a nie okresu istnienia. Obiekty zadeklarowane w module mają okres istnienia rozszerzony poza moduł do pierwszego otaczającego go bloku.

Rejon

W CHILLU dostępna jest konstrukcja zwana rejonem (ang. region), służąca do wzajemnego wykluczenia procesów współzawodniczących o wspólne zasoby. Rejon jest ograniczony nawiasami **REGION** i **END**; i może zawierać wewnątrz tylko definicje i deklaracje, łącznie z deklaracjami udostępnienia i dzierżawienia. Wspólne obiekty, o które procesy ubiegają się, są zadeklarowane jako lokalne w rejonie i nie mogą być udostępnione na zewnątrz. Dostęp do wspólnych obiektów może odbywać się tylko za pośrednictwem procedur rejonu. Procedury te są zdefiniowane jako lokalne w rejonie, ale udostępnione na zewnątrz.

Jeżeli jakiś proces wywoła procedurę z rejonu, to inny proces wywołujący tę samą lub inną procedurę z tego rejonu jest opóźniany (ang. delayed) — dopóki nie zakończy się procedura wywołana przez pierwszy proces. Zatem tylko jeden proces może w danej chwili mieć dostęp do wspólnych obiektów zadeklarowanych wewnątrz rejonu.

Rejony nie mogą być zawarte w innych rejonach ani wewnątrz bloków. Procedury rejonu nie mogą wywoływać procedur innych rejonów — ani bezpośrednio, ani pośrednio przez inne procedury.

Przykład

Rozważymy rejon nadzorujący przydział dziesięciu zasobów. Każdy z zasobów może być przydzielany i zwalniany pojedynczo — tylko jeden w danej chwili. Jeśli nie jest dostępny żaden zasób, to proces ubiegający się jest opóźniony, dopóki inny proces nie zwolni zasobu.

1. **ALLOCATE_RESOURCES**;
2. **REGION**
3. **GRANT ALLOCATE, DELLOCATE**;

¹⁾ Nazwa jest dobrze widoczna, jeżeli jest bezpośrednio lub pośrednio dobrze widoczna. Atrybut dobrej, bezpośrednio widoczności jest uzyskiwany przez nazwę: a) w bloku, w którym jest deklarowana, b) wskutek użycia deklaracji udostępnienia i dzierżawienia. Atrybut dobrej, pośredniej widoczności jest uzyskiwany przez nazwę: a) w bloku zagnieżdżonym wskutek deklaracji w bloku otaczającym, b) wskutek deklaracji nazwy wraz z atrybutem **PERVASIVE**.

²⁾ Pojęcia widoczności i okresu istnienia wyjaśnione są w dalszej części artykułu.

```

4. NEWMODE RESOURCE_SET = INT (0 : 9);
5. DCL ALLOCATED ARRAY (RESOURCE_SET) BOOL
   := (: (RESOURCE_SET) : FALSE :);
6. DCL RESOURCE_FREED EVEVT;
7. ALLOCATE;
8. PROC () (INT);
9. DO FOR EVER;
10. DO FOR I IN RESOURCE_SET;
11. IF NOT ALLOCATED (I)
12. THEN
13. ALLOCATED (I) := TRUE;
14. RETURN I;
15. FI;
16. OD;
17. DELAY RESOURCE_FREED;
18. OD;
19. END ALLOCATE;
20. DEALLOCATE;
21. PROC (I INT);
22. ALLOCATED (I) := FALSE;
23. CONTINUE RESOURCE_FREED;
24. END DEALLOCATE;
25. END ALLOCATE_RESOURCES;

```

Numeracja linii nie jest elementem syntaktycznym CHILLA. W tym przykładzie i w następnych wprowadzono tę numerację dla ułatwienia opisu.

Instrukcja **GRANT** w linii 3 udostępnia procesom ubiegającym się o zasoby procedury **ALLOCATE** i **DEALLOCATE**. W linii 4 zdefiniowano zbiór zasobów **RESOURCE_SET** ponumerowanych od 0 do 9. Linia 5 zawiera deklarację tablicy logicznej **ALLOCATED** ze wskaźnikami stanu: zajęty — **TRUE** i wolny — **FALSE** dla każdego zasobu zbioru. Wszystkie elementy z tablicy mają wartość początkową **FALSE**. W linii 6 zadeklarowano zdarzenie **RESOURCE_FREED**, na które proces musi oczekiwać wtedy, gdy żaden z zasobów nie jest dostępny. Linie 7—19 stanowią definicję procedury **ALLOCATE**. W pętli zapisanej w liniach 10—16 przeglądana jest tablica w poszukiwaniu wolnego zasobu. Jeżeli wolny zasób został znaleziony, to jest on zajmowany w linii 13, a procedura zwraca indeks przydzielonego zasobu. Jeżeli brak jest wolnego zasobu, to w linii 17 proces przejdzie w stan oczekiwania do chwili zwolnienia zasobu. Procedura **DEALLOCATED** w liniach 20—24 nadaje odpowiedniemu elementowi tablicy **ALLOCATED** wartość **FALSE** i sygnalizuje ten fakt, w linii 23, procesom oczekującym na dostęp do zasobu. Następuje wywołanie powrót do pętli w linii 9 w celu ponownego przeszukania tablicy **ALLOCATED** i przydzielenia zasobu.

Bloki

Istnieją trzy postacie bloków: blok „begin”, blok procedury i blok procesu. Bloki determinują zarówno okres istnienia, jak i widoczność obiektów w nich zadeklarowanych. Bloki można zagnieżdżać. Wówczas wszystkie nazwy zadeklarowane w zewnętrznym bloku są automatycznie widoczne we wszystkich wewnętrznych blokach, zaś nazwy zdefiniowane w wewnętrznych blokach nie są widoczne w blokach zewnętrznych. Można powiedzieć, że nazwy są lokalne w blokach, w których są zadeklarowane i są globalne we wszystkich blokach wewnętrznych.

Wszystkie zmienne zadeklarowane w bloku i we wszystkich wewnętrznych modułach bloku (z wyjątkiem zmiennych z atrybutem **STATIC**) mają przydzielane miejsca w pamięci podczas wchodzenia do bloku i zwalniane podczas opuszczania go, co determinuje okres istnienia zmiennej. Wielkość przydzielonej pamięci określana jest przez kompilator i użytkownik ma możliwość poznania rozmiarów obiektu przez użycie procedury standardowej **SIZE**.

W przypadku, gdy nazwa jest zadeklarowana z atrybutem **STATIC** okres istnienia zmiennej nie jest ograniczony. Zmienne mogą być tworzone dynamicznie przez wykonanie procedury standardowej **GETSTACK**.

Inicjowanie zmiennej w deklaracji jest dokonywane za każdym razem gdy osiągnie ją sterowanie przebiegiem programu. Dodanie słowa **INIT** przed nazwą powoduje, że inicjowanie jest wykonywane tylko raz podczas umiejscawiania zmiennych.

PROCEDURY

Ważnym narzędziem strukturalizacji w języku CHILL jest procedura. Wejście do procedury jest możliwe przez jej

wywołanie. Po wykonaniu procedury program jest kontynuowany bezpośrednio po instrukcji wywołującej.

Wywołanie procedury składa się z nazwy procedury, po której następuje, ujęta w nawiasy, lista zawierająca zero lub więcej parametrów aktualnych. Parametrem aktualnym może być dowolne wyrażenie przyjmujące wartość odpowiedniej klasy. Wartość każdego z parametrów aktualnych raz określona jest dostępna dla instrukcji znajdujących się wewnątrz procedury.

Procedura może dostarczyć wartość. Wywołanie procedury jest wtedy oznaczeniem wartości i może być używane jako argument w wyrażeniu, np.:

```
RESULT := MAX(X,Y)*2;
```

Poniższa procedura **MAX** z dwoma parametrami:

```

MAX: PROC (I, J READ INT) (INT);
  IF I < J
  THEN
  RETURN I;
  ELSE
  RETURN J;
  FI;
END MAX;

```

dostarcza wartość większego parametru. Część **(I, J READ INT)** jest specyfikacją parametrów formalnych **I** i **J**, obu typu całkowitego o niezmiennych wartościach, co oznaczono przez **READ**. Nazwy te są lokalne w bloku procedury. **(INT)** jest specyfikacją typu wyniku i oznacza, że zwracana wartość jest całkowita. Wartość wyniku zwracana przez procedurę jest wskazywana instrukcją **RETURN**.

Poza powyższymi podstawowymi właściwościami, definicja procedury w CHILLU może mieć punkty wielokrotnego wejścia oraz atrybuty specyfikujące rodzaj procedury. Punkty wejścia są określone przez instrukcję **ENTRY**. Atrybutami mogą być: **RECURSIVE**, **GENERAL SIMPLE** i **INLINE**. Tylko procedura z atrybutem **RECURSIVE** może być wywołana rekurencyjnie. Procedury z atrybutem **GENERAL** mogą być użyte w sposób dynamiczny, w przeciwieństwie do procedur z atrybutem **SIMPLE**. Atrybut **INLINE** powoduje wstawienie procedury w punkcie wywołania, zamiast zwykle stosowanych powiązań z procedurą.

Definicja procedury może mieć dołączony program obsługi wyjątków oraz listę wyjątków formalnych (ang. formal exceptions).

Przekazywanie parametrów

CHILL ma kilka mechanizmów przekazywania parametrów do i z procedury. Podstawowy mechanizm polega na przekazywaniu parametrów przez wartość (ang. pass by value). Wówczas wartości parametrów aktualnych są kopiowane do obiektów lokalnych procedury.

Parametry formalne procedury mogą zawierać atrybuty. Przekazywanie parametrów przez wartość może być jawnie wskazane przez użycie atrybutu **IN**. Atrybut **OUT** wskazuje na kopiowanie wartości z obiektów lokalnych do obiektów określonych przez parametry aktualne. **INOUT** łączy właściwości tych dwóch atrybutów.

Sposób przekazywania przez wartość jest efektywny, jeżeli obiekty są małe; natomiast w przypadku dużych obiektów, np. wielowymiarowych tablic, taki mechanizm przekazywania jest zbyt kosztowny. W CHILLU istnieje możliwość przekazywania parametrów przez obiekty zadeklarowane w otoczeniu procedury. Zmianę wartości tych obiektów przez procedurę nazywa się efektem ubocznym procedury (ang. side-effect). Na ogół unika się efektów ubocznych, gdyż zaburzają one strukturalność programu i stanowią potencjalne źródło błędów.

Powrót z procedury

Gdy wykonywanie procedury osiągnie słowo **END**, sterowanie przebiegiem programu powraca do miejsca wywołania. Powrót może nastąpić również przez użycie instrukcji powrotu z procedury **RETURN**. W przypadku, gdy procedura zwraca wyniki, instrukcja powrotu może także wskazywać jego wartość. Gdy niezbędne jest wskazanie wartości w punkcie różnym od instrukcji powrotu, można użyć instrukcji wskazania wyniku **RESULT**, np. użycie in-

strukcji:

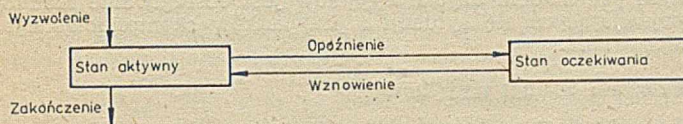
```
RETURN X + S ;
```

jest równoważne instrukcjom:

```
RESULT X + S ;  
RETURN ;
```

PROCES

Proces jest dynamiczną jednostką programu przetwarzaną współbieżnie, tj. równolegle z innymi procesami. Operacje procesu wykonywane są zawsze sekwencyjnie. Proces, który został wyzwolony (instrukcja **START**), a nie jest jeszcze zakończony, może znajdować się w jednym z dwóch stanów: aktywnym lub oczekiwania (ang. active, delayed). Zmiany stanów są powodowane odpowiednimi akcjami programu. Można to zilustrować zamieszczonym obok rysunkiem.



Wpływ akcji programu na zmianę stanów

Definicja procesu stanowi wzorzec, na podstawie którego są tworzone (ang. create) procesy. Wzorzec ten opisuje akcje procesu i jego obiekty lokalne. Instrukcja definicji procesu (ang. process definition statement) jest składniowo podobna do instrukcji definicji procedury — z tym, że zamiast słowa **PROC** użyte jest słowo **PROCESS**. Definicje procesu nie są dozwolone wewnątrz innych procesów, bloków „begin” i procedur.

Wyzwolenie i zakończenie procesu

Na podstawie jednej definicji procesu można utworzyć wiele procesów. Każdy z nich jest tworzony przez wykonanie instrukcji wyzwolenia procesu, która składa się ze słowa **START**, nazwy procesu i listy parametrów aktualnych, np.:

```
START LOCAL_CALL (ACTIVE_LINE1, 100,5);  
START LOCAL_CALL (ACTIVE_LINE2, 50,8);
```

Wyzwolone dwa procesy **LOCAL_CALL** poniżej drugiej linii wykonywane są współbieżnie.

Zakończenie procesu następuje po wykonaniu instrukcji **STOP**, która jest umieszczona zawsze w sposób niejawni na końcu definicji procesu. Jeżeli wymagane jest zakończenie procesu w innym miejscu, używa się jawnej instrukcji **STOP** np.:

```
LOCAL_CALL:  
PROCESS (LINE_NO, INT);  
DO FOR EVER;  
...  
IF LINE_NO > 1000 THEN STOP;  
FI;  
OD;  
END LOCAL_CALL;
```

Instrukcja **STOP** może być umieszczona w procedurze powodując zakończenie procesu wywołującego tę procedurę.

Identyfikacja procesów

Do rozróżnienia poszczególnych procesów używane są identyfikatory procesów, które są obiektami typu **INSTANCE**. Istnieje kilka sposobów nadawania wartości identyfikatorom. Podstawowym z nich jest użycie instrukcji **START**, która dostarcza wartość identyfikującą proces w chwili jej wykonania. Jeżeli przez **CALL_1** oznaczymy identyfikator procesu **LOCAL_CALL**, to po wykonaniu sekwencji:

```
DCL CALL_1 INSTANCE;  
CALL_1 := START LOCAL_CALL (ACTIVE_LINE, 100,20);
```

istnieje możliwość odwołania się do tego procesu przez wartość identyfikatora **CALL_1**. Dla ustalenia tożsamości aktualnie wykonywanego procesu można użyć operatora **THIS**.

Identyfikator nie wskazujący żadnego procesu ma wartość **NULL**.

Identyfikację procesów zilustrowano w poniższym przykładzie:

```
1. DCL CALL_1, CALL_2 INSTANCE := NULL;  
2. CALL_HANDLER:  
3. PROCESS (LINE_NO INT);  
4. ...  
4. CALL_2 := THIS;  
5. END;  
6. CALL_1 := START CALL_HANDLER(1,200,5);  
7. START CALL_HANDLER(2,200,5);
```

Po wyzwoleniu procesu w linii 6 i wykonaniu instrukcji w linii 4 zmienne **CALL_1** i **CALL_2** identyfikują ten sam proces. Na skutek wyzwolenia następnego procesu w linii 7 identyfikujący go **CALL_2** otrzyma w linii 4 wartość różną od **CALL_1**. W przykładzie założono, że pierwszy proces wykona linię 4 zanim zacznie ją wykonywać drugi proces.

Wykonanie współbieżne

Gdy do wykonania jakiegoś zadania wyzwolonych jest kilka procesów lub procesy te współzawodniczą o wspólne zasoby, to pojawia się potrzeba ich koordynacji. **CHILL** ma kilka sposobów koordynacji procesów:

- synchronizacja powodująca wykonanie procesów w trybie krokowym (ang. lockstep)
- komunikacja przez wymianę komunikatów (obiektów danych) między procesami, co implikuje synchronizację
- wzajemne wykluczanie się w celu bezpiecznego dostępu do wspólnych zasobów.

Przy użyciu różnych sposobów synchronizacji tych samych procesów należy mieć na uwadze niebezpieczeństwo ich zakleszczenia.

Zdarzenia

Do synchronizacji procesów używane są obiekty typu zdarzeniowego (ang. event mode). Proces można opóźnić instrukcją **DELAY**, w oczekiwaniu na wystąpienie określonego zdarzenia, które wznowi działanie innego procesu (lub procesów) przez wykonanie instrukcji **CONTINUE**. Użycie zdarzeń przedstawiono w poniższym przykładzie.

```
1. SWITCHBOARD:  
2. MODULE  
3. DCL OPERATOR_IS_READY EVENT;  
4. CALL_DISTRIBUTOR:  
5. PROCESS ();  
6. SYN HANDLING_TIME = 10 (* SEKUND *);  
7. DO FOR EVER;  
8. WAIT(HANDLING_TIME);  
9. CONTINUE OPERATOR_IS_READY;  
10. OD;  
11. END CALL_DISTRIBUTOR;  
12. CALL:  
13. PROCESS ();  
14. DELAY OPERATOR_IS_READY;  
    (* jakieś akcje, wywołanie operatora *)  
15. END CALL;  
16. START CALL_DISTRIBUTOR ();  
17. DO FOR I IN INT(1:100);  
18. START CALL ;  
19. OD;  
20. END SWITCHBOARD;
```

Przykład ten ilustruje działanie łącznicy telefonicznej, która dostarcza zgłoszenia do operatora w stałych odstępach czasu. W liniach 17—19 generowanych jest 100 zgłoszeń przez wyzwolenie 100 procesów **CALL**. Wszystkie te procesy są opóźniane instrukcją **DELAY** w linii 14. Co dziewięć sekund w linii 9 generowane jest zdarzenie **OPERATOR_IS_READY**, które powoduje, że jeden z oczekujących na nie procesów jest wznawiany i przekazuje zgłoszenie do operatora.

Instrukcja **DELAY** użyta w powyższym przykładzie pozwala opóźnić proces w oczekiwaniu na jedno określone zdarzenie. Instrukcja **DELAY CASE** umożliwia odbieranie przez proces jednego lub wielu określonych zdarzeń.

Powyższy przykład można rozbudować wprowadzając drugie zdarzenie **SWITCH_IS_CLOSED**. Instrukcja **DELAY**

CASE umieszczona zamiast instrukcji **DELAY** w linii 14 objętej definicją procesu **CALL**, może być zapisana następująco:

```
DELAY CASE
(OPERATOR_IS_READY): (* akcje *)
(SWITCH_IS_CLOSED):
  DO FOR I IN INT(1:100);
    CONTINUE OPERATOR_IS_READY;
  (* akcje *)
OD;
ESAC;
```

Procesy **CALL** pozostają w stanie oczekiwania, dopóki nie zajdzie jedno z następujących zdarzeń: **OPERATOR_IS_READY** lub **SWITCH_IS_CLOSED**. Wznowienie procesu nastąpi bez względu na to, które ze zdarzeń wystąpi pierwsze.

Ogólnie mówiąc — jeżeli proces jest w stanie oczekiwania, to może odebrać tylko jedno zdarzenie. Jeżeli dwa zdarzenia wystąpią jednocześnie, to jedno z nich jest wybierane arbitralnie.

Instrukcje **DELAY** i **DELAY CASE** mogą mieć dołączone priorytety oznaczone przez **PRIORYTY** (wyrażenie całkowite). Oprócz tego po **DELAY CASE** może być dodana instrukcja **SET** (identyfikator) dla uzyskania wartości identyfikatora procesu, który spowodował zdarzenie.

Bufory

Zmienne typu buforowego (ang. buffer mode) i zdefiniowane na nich operacje używane są do komunikacji i synchronizacji procesów. Są one deklarowane przez słowo **BUFFER**, po którym podany jest typ przesyłanego komunikatu. Komunikaty mogą być przesyłane do buforów i odbierane z nich przez procesy za pomocą instrukcji **SEND** i **RECEIVE**, jak w poniższym przykładzie:

```
1. MODULE
2.   NEWMODE INPUT = SET(ANSWER, CLEAR_BACK, RELEASE);
3.   DCL MAIL_BOX BUFFER(10 INPUT);
4.   RECEIVER:
5.     PROCESS();
6.     DCL MESSAGE INPUT;
7.     MESSAGE := RECEIVE MAIL_BOX;
8.     SEND MAIL_BOX(RELEASE);
9.   END RECEIVER;
10.  START RECEIVER ;
11. END;
```

W linii 2 zdefiniowano typ komunikatów, które mają być wysłane. Linia 3 zawiera deklarację bufora **MAIL_BOX**, który może pomieścić do dziesięciu komunikatów typu **INPUT**. W liniach 4—9 zdefiniowano proces **RECEIVER** z zadeklarowaną nazwą **MESSAGE**. W linii 7 instrukcja **RECEIVE** zatrzyma proces aż do chwili, gdy **MESSAGE** znajdzie się w **MAIL_BOX**. Gdy to nastąpi, komunikat w buforze zostanie przypisany zmiennej **MESSAGE** i proces będzie wznowiony. Wysłany w linii 8 komunikat **RELEASE** do bufora może być z niego odebrany przez inny proces.

Komunikaty przechowywane w buforze są trwałe, w odróżnieniu od zdarzeń — raz w nim umieszczone oczekują aż do odebrania. Proces wysyłający komunikat nie jest opóźniany w oczekiwaniu na umieszczenie komunikatu w buforze. Instrukcja wysyłania do bufora może mieć nadany priorytet.

Sygnały

Sygnały są używane do synchronizacji i komunikacji procesów. Definiuje się je następująco:

```
SIGNAL T TO RECEIVER;
SIGNAL S = (INT,BOOL);
```

Pierwsza linia definiuje sygnał „T” bez komunikatu, przeznaczony tylko do synchronizacji. Druga linia zdefiniuje sygnał „S” wraz z komunikatem złożonym z wartości typu **INT** i **BOOL**. Definicja sygnału może zawierać ujętą w nawiasy listę oznaczeń typów.

Instrukcja **SEND** użyta do wysyłania sygnału może mieć część **TO** wskazującą proces, do którego sygnał jest wysyłany, np.:

```
SEND S (5, TRUE) TO PI;
SEND T;
```

W pierwszej linii sygnał „S” jest wysyłany do procesu o identyfikatorze **PI**. Sygnał „S” w drugiej linii nie jest ukierunkowany, gdyż definicja sygnału może również zawierać

wskazanie, po którym następuje lista nazw procesów umożliwiających jego odebranie; wtedy część wskazująca instrukcji **SEND** jest pomijana.

Sygnał jest trwały. Instrukcja wysyłania sygnału może mieć nadany priorytet. Sygnały umieszczowane są automatycznie — w odróżnieniu od buforów, których umieszczenie jest kontrolowane przez użytkownika.

Odebranie dowolnego elementu ze zbioru buforów i sygnałów umożliwia instrukcja **RECEIVE CASE**. Jest ona podobna do instrukcji **DELAY CASE**, przy czym ma ona dodatkowe środki do obsługi ich części komunikatowej.

OBSŁUGA WYJĄTKÓW

Wyjątek (ang. exception) jest związany z sytuacją, która nie powinna wystąpić podczas poprawnego wykonywania programu. Powiązanie zaistniałych sytuacji z wyjątkiem zależy w dużym stopniu od subiektywnego odczucia programisty. Jednym z zastosowań wyjątków jest obsługiwanie sytuacji, które często nazywa się błędami, jak np. nadmiar arytmetyczny lub próba dostępu przez pusty wskaźnik. Innym przykładem użycia wyjątków jest obsługiwanie rzadko pojawiających się sytuacji i przypadków skomplikowanego sterowania.

Program obsługi wyjątków

Po wystąpieniu wyjątku sterowanie jest przekazywane do programu jego obsługi. Program ten może być dołączony do dowolnej instrukcji, licząc także deklaracje z inicjowaniem oraz definicje procedur i procesów.

Program obsługi wyjątków jest ujmowany w nawiasy **ON** i **END**. Każdy człon zawartej w nim alternatywy składa się z nazwy wyjątku i listy instrukcji. Człon alternatywy **ELSE** obsługuje każdy zaistniały wyjątek, który nie jest wymieniony w pozostałych członach. Alternatywy są wyszczególnione w liniach 2—6 poniższego przykładu.

```
1. DCL A INT(0 : 256), B, C INT := 0;
   (* tekst programu działającego na B i C *)
2. A := B+C ON
3.   (OVERFLOW):   OUTTEXT ("OVERFLOW");
4.   (RANGEFALL): OUTTEXT ("RANGEFALL");
5.   ELSE         OUTTEXT ("OTHER ERROR");
6.   END;
```

Wymuszanie wyjątków

Wyjątki mogą być wymuszane jawnie lub niejawnie. Wymuszenie jawnie następuje wskutek wykonania instrukcji **CAUSE** zawierającej nazwę wyjątku. Niejawne wymuszenie następuje wskutek wykonania pewnych konstrukcji językowych, które spowodują sytuacje wyjątkowe, jak np. nadmiar. Zbiór wyjątków, które mogą być wymuszone niejawnie, jest w **CHILLU** wstępnie zdefiniowany. Wyjątki z tego zbioru mogą być również wymuszane jawnie.

Wyjątki pojawiające się wewnątrz procedury powinny być niekiedy obsługiwane w punkcie wywołania, a nie na końcu definicji procedury. Do obsługi takich przypadków przewidziano w **CHILLU** wyjątki formalne (ang. formal exceptions). Lista wyjątków formalnych jest podawana w definicji procedury.

Poszukiwanie programów obsługi wyjątków

Jeżeli program obsługi wyjątku nie jest dołączony do instrukcji, która spowodowała wyjątek to poszukiwany jest on na zewnątrz przez zagnieżdżone poziomy instrukcji. Sprawdza się również, czy istnieje wyjątek formalny o tej nazwie. W przypadku jego znalezienia sterowanie jest przekazywane do punktu wywołania i poszukiwanie jest kontynuowane. Jeżeli wyjątek, dla którego nie ma programu obsługi, wystąpi w procedurze lub procesie, to kończą się one nieprawidłowo, dowodząc, że program jest błędny.

LITERATURA

- [1] CCITT: CHILL Language Definition. CCITT Recommendation Z 200, November 1980
- [2] CCITT: Introduction to CHILL. CCITT Manual, May 1980
- [3] Wprowadzenie do CHILLA — języka wysokiego poziomu CCITT. Instytut Łączności, Warszawa, sierpień 1983
- [4] Jarociński M., Średniawa M.: Wprowadzenie do języka CHILL. Instytut Telekomunikacji. Politechnika Warszawska, Warszawa, 1983.

mikro KLAN

Procesory 16-bitowe zaczynają już wkraczać do konstrukcji hobbyistycznych (podobno nawet w kraju). Wydawałoby się, że polski amator nie powinien przeżywać rozterek związanych z wyborem typu jednostki centralnej, skoro profesjonalści zdecydowali się na 8086. Wybór ten został jednak zdeterminowany przez spodziewane dostawy tego układu z ZSRR. Dostawy te jeśli faktycznie nastąpią — będą miały, minimalny wpływ na zaopatrzenie hobbystów. Tak więc mogą się oni kierować bardziej zdrowo-rozważkowymi przesłankami. Mikroprocesor 68000 ma bez wątpienia większe możliwości niż 5086. W oparciu o niego zbudowano LISE, MACINTOSHA, a nawet wzbudzający duże zainteresowanie SINCLAIR QL. E.M. Carter i A.B. Bonds zaprezentowali w piśmie BYTE konstrukcję, której całkowity koszt w amerykańskich warunkach nie przekracza 200 dolarów. W ich konstrukcja nabrała życia, trzeba ją podłączyć przez kanał V24 do jakiegoś terminala. Ten słaby punkt może okazać się bardzo użyteczny w naszych warunkach. Wykorzystując poprzednią 8-bitową konstrukcję, mamy wstępnie rozwiązany problem sterownika CRT, obsługi klawiatury, a może nawet współpracy z dyskami elastycznymi. Przez tego typu przystawkę pozwolimy naszemu staremu komputerowi na przeżycie drugiej młodości.

Jednopłytkowy mikrokomputer VU68K

Przedstawione poniżej rozwiązanie powstało na Uniwersytecie Vanderbilt w USA. Jest przykładem minimalnego systemu wykorzystującego mikroprocesor M68000 firmy MOTOROLA. Było jednak optymalizowane pod względem kosztów z perspektywy rynku USA. W naszych warunkach niewątpliwie celowe będzie wprowadzenie pewnych zmian, które zależą jednak w znaczny sposób od indywidualnego pojęcia konstruktora. Dlatego też przedstawimy układ takiej postaci, jak proponują go autorzy.

Mikroprocesor 68000 operuje na słowach o długości 16 bitów. Wynika stąd konieczność „podwójnego” stosowania układów pamięci, które cechuje organizacja bajtowa (8 bitów). Przyłączenie dwóch układów 6116 (CMOS — RAM) i dwóch 2716 (EPROM) do szyny danych daje potrzebne 16 bitów.

Dla komunikacji z otoczeniem zaproponowano dwa zintegrowane konwertery równoległo-szerokowe (6850 ACIA — ang. Asynchronous Communications Interface Adapters). Dokonują one zmiany informacji z postaci szeregowej, przychodzącej spoza systemu, na równoległą — przyjmowaną przez mikroprocesor, i odwrotnie — otrzymywaną z procesora wysyłającą zgodnie ze standardem RS232 (V24) do urządzeń zewnętrznych. Układy te spełniają podobną rolę jak 8251 w systemach Intelowskich. Zastosowanie aż dwóch układów 6850 wynika z faktu, iż jeden jest przeznaczony do współpracy z konsolą operatorską, podczas gdy drugi może być wykorzystany dla wymiany informacji z innym systemem. Jeśli współpracujący mikrokomputer może także spełniać funkcję konsoli, wystarczy zastosować jeden układ 6850.

Układy IC14 i IC15 to konwertery poziomów napięciowych. Zamiast 1488 (nadajnik) i 1489 (odbiornik) można tu z równym powodzeniem wykorzystać

nico bardziej u nas popularne 75150 i 75154, a nawet konstrukcje na elementach dyskretnych. Ponieważ są to jedyne układy wymagające zasilania napięciami +12 V i -12 V, istotnym uproszczeniem całej konstrukcji byłoby ich wyeliminowanie. Jest to możliwe, jeżeli system, z którym odbywa się współpraca, jest oddalony nie więcej jak o kilkanaście centymetrów. Oczywiście połączenie musi wtedy również omijać konwertery poziomów w komputerze współpracującym.

Układ IC13 dostarcza sygnał sterujący częstotliwością transmisji szeregowej. Może on z powodzeniem zostać zastąpiony dowolnym innym rozwiązaniem stosowanym w krajowych konstrukcjach.

Dla prawidłowego funkcjonowania systemu niezbędna jest możliwość jego wyzerowania. W tym celu na wejście RESET (i HALT) przez co najmniej 100 ms musi być podany poziom niski. W rozwiązaniu zastosowano w tym celu specjalny przełącznik, wymagający przyśnięcia zarówno dla włączenia, jak i wyłączenia. Przełącznik został „odkłócony” (drgania zestyków) przez przerzutnik zbudowany z dwóch bramek typu NAND.

ASYNCHRONICZNE I SYNCHRONICZNE OPERACJE NA SZYBIE SYSTEMOWEJ

Mikroprocesor 68000 został zaprojektowany z myślą o wykonywaniu asynchronicznych operacji na szynie systemowej. W rozwiązaniu takim przekazywanie danych nie jest uzależnione od cykli maszynowych mikroprocesora (tak jak to ma miejsce np. dla Z80 czy 8086), lecz od sygnału potwierdzającego, generowanego przez układ współpracujący. W mikrokomputerze VU68K asynchroniczna komunikacja odbywa się między układami pamięci a procesorem.

Uspołecznianie importera

Absurd ograniczeń przywozów komputerów osobistych do kraju wywołuje coraz większe oburzenie. Ale i gromki śmiech. Trudno mu zresztą nie ulec, przyrzawszy się takiemu odpędzaniu czarownic.

Profesor Iwo Białynicki-Biruła pisze do prezesa Głównego Urzędu Cel:

„Postuluję bezzwłoczne zniesienie wymogu uzyskiwania specjalnych zezwoleń na przywóz do Polski komputerów osobistych. Można to po prostu uczynić przez zaliczenie tych urządzeń do elektronicznego sprzętu powszechnego użytku (póz. 58 pt 4 taryfy celnej). Nie ma, moim zdaniem, żadnej trudności z podaniem definicji taryfowej komputera osobistego, ponieważ urządzenia te są zawsze sprzedawane pod nazwą komputery osobiste (personal computer) bądź komputery domowe (home computer) i we wszystkich dowodach sprzedaży i instrukcjach obsługi używa się wyłącznie tych nazw. Zaliczanie komputerów osobistych do systemów komputerowych, objętych pozycją 23 taryfy celnej, jest anachronizmem i było już niejednokrotnie przedmiotem drwin w prasie. Na świecie produkuje się obecnie kilka milionów komputerów osobistych rocznie i liczba ta gwałtownie rośnie. Sprzedawane są one często w domach towarowych w stoiskach z zabawkami i stanowią niewątpliwie sprzęt powszechnego użytku, przede wszystkim dla młodzieży. Jeżeli Polska ma nadrabiać, a nie powiększać, lukę technologiczną w dziedzinie elektronizacji, to może się to odbywać tylko przy pełnej aktywizacji całego społeczeństwa. (...)

Jak może umotywić swoje podanie do Urzędu Celnego kilkunastoletni chłopiec, któremu wujek chce przysłać na Gwiazdkę komputer domowy? W krajach, które mamy doganiać, tacy właśnie chłopcy stanowią będą za kilka lat kadry wybitnych fachowców elektroniki i informatyki, oswójoną od dzieciennych lat z komputerami.”

Dziękujemy, Profesorze. Inkwizytora trzeba rozbrajać. W końcu weźmiemy go choćby śmiechem.

Postulat jest niepodważalny, jedynie słuszny. Podpisujemy się pod nim wszystkimi redakcyjnymi rękami. Ale jeśli już zakreślać horyzonty, to trzeba by postulat uzupełnić. Po pierwsze — aby przy okazji uproszczenia procedury przywozów nie podwyższono cła; po drugie — sprzęt mikrokomputerowy wyższej klasy powinien korzystać (przy obrocie w kraju) z ulg podatkowych. Powinien. Oby nie został tylko śmiechem.

Z ostatniej chwili: Inkwizytor osłabł — już nie wymaga pozwoleń. Szczegóły w tym miejscu za miesiąc.

ZBIGNIEW GLUZA

Ponieważ układy 6116 i 2716 nie generują same z siebie sygnału potwierdzającego, funkcję tę realizuje 4-bitowy licznik (IC4) sterowany sygnałem o częstotliwości 5 MHz. W chwili gdy procesor wystawi sygnał AS (informujący o podaniu adresu na szynę), do licznika wpisywana jest wartość początkowa 1100. Po czterech taktach zegara na wejściu podłączonym do DTACK/ pojawia się stan 0, informując, że można już wprowadzać dane. W rozwiązaniu założono, że transfer danych możliwy jest w ciągu 800 ns.

Mikroprocesor 68000 został zaprojektowany w taki sposób, by możliwe było również wykorzystanie bogatego zestawu układów ze starszej rodziny 6800. W tym celu wykorzystywane jest wejście VPA (ang. Valid Peripheral Address). Z chwilą gdy na szynie adre-

sowej znajdzie się adres przypisany układowi peryferyjnemu z rodziny 6800, na wejście VPA/ powinien zostać podany poziom niski. Spowoduje to, że procesor przejdzie do wykonywania cyklu synchronicznego i wygeneruje stosowane w tym trybie pracy sygnały: VMA/ (ang. Valid Memory Address) oraz E (ang. Enable). Pierwszy z nich bywa wykorzystywany w logice wybierającej układ peryferyjny. Natomiast sygnał E w chwili gdy przejdzie w stan wysoki — aktywizuje wykonywanie operacji przez układ peryferyjny.

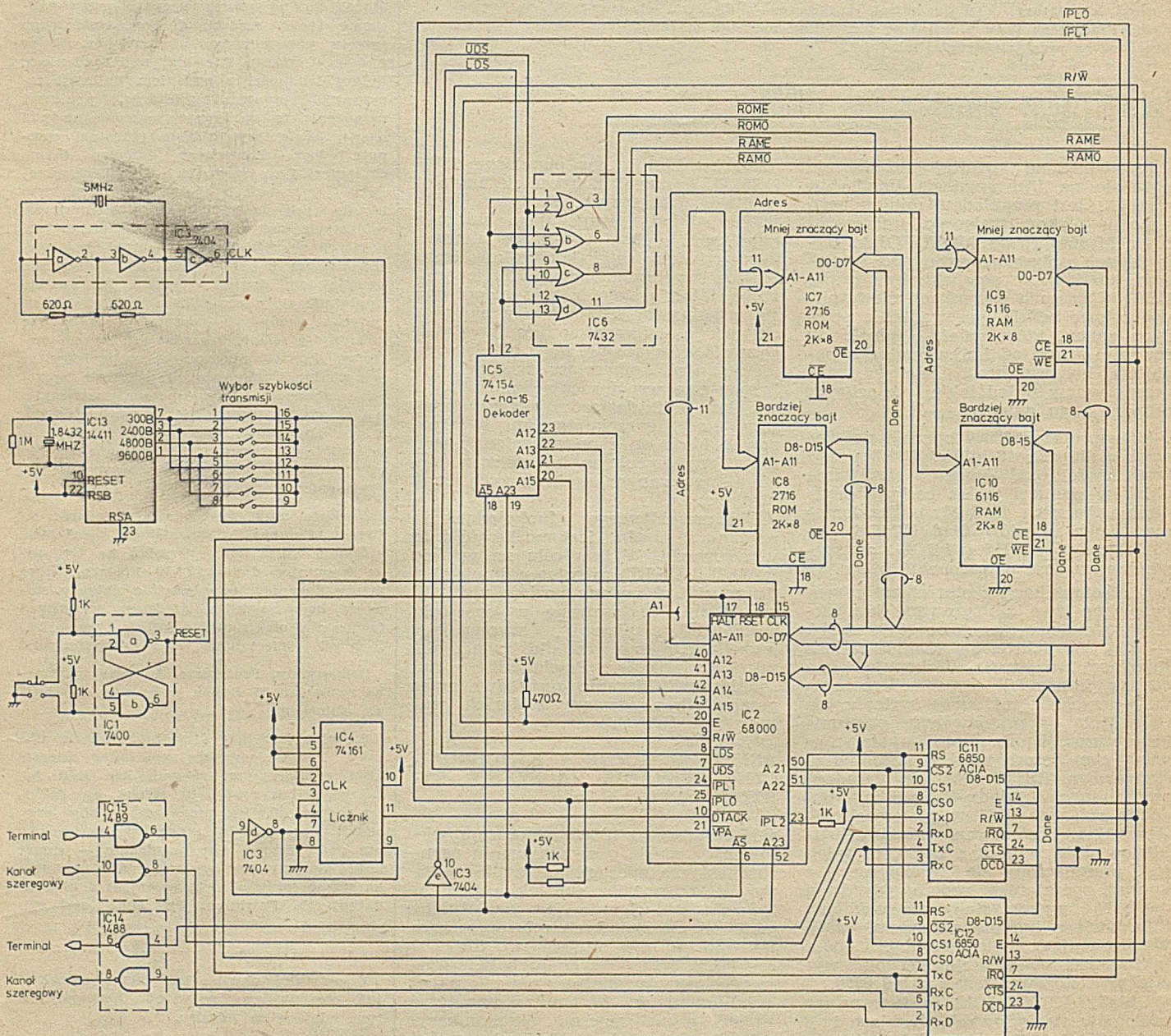
W omawianym rozwiązaniu tryb synchroniczny wykorzystywany jest przy współpracy z układami transmisji 6850¹⁾. Układom tym przypisano adresy: A00000_H-A00002_H i C00000_H-C00002_H. W momencie wystawienia ich przez procesor na szynę adreso-

wą, linia A23 znajduje się w stanie logicznym 1. Wykorzystując ten fakt, połączono poprzez inwerter wyjście tej linii z wejściem VPA/ mikroprocesora. Tak więc z chwilą wybrania dowolnego adresu, dla którego linia A23 jest w stanie logicznym 1, procesor przechodzi do wykonania operacji w trybie synchronicznym. Należy to uwzględnić przy ewentualnej rozbudowie systemu.

PODZIAŁ PRZESTRZENI ADRESOWEJ

Mikroprocesor 68000 ma 23 linie adresowe, co pozwala na uzyskanie po-

¹⁾ Asynchroniczny protokół transmisji realizowany przez układ 6850 nie ma nic wspólnego z synchronicznym trybem współpracy tego układu z procesorem



Schemat ideowy mikrokomputera — przystawki VU 68K

nad 8 mln różnych sekwencji adresowych. W tej olbrzymiej przestrzeni adresowej trzeba wydzielić obszary o rozmiarach 2 K, w których umieszczona zostanie przewidziana w rozwiązaniu pamięć. Ogólnie rzecz biorąc, można by wydzielić ponad 4 tys. takich obszarów. Oczywiście nie ma to sensu — pamięć o takich rozmiarach należałoby realizować z wykorzystaniem układów o znacznie większej pojemności (64 K lub nawet 256 K). Dlatego trzeba przyjąć kompromis, w którym pewna część potencjalnych możliwości 68000 nie zostanie wykorzystana.

Autorzy oferują w proponowanym rozwiązaniu zastosowanie pamięci o pojemności 4 K słów z możliwością rozszerzenia do 32 K słów. Linie adresowe A1...A11 są wykorzystywane do wyboru poszczególnych słów w ramach układów pamięci. Kolejne cztery linie A12...A15 służą do zdekodowania 16 bloków. W tym celu zostały doprowadzone do układu (IC5), realizującego funkcję dekodera 4 z 16. Wyjścia dekodera poprzez bramki typu OR (IC6) sterują wyborem poszczególnych układów pamięci.

Ponieważ szyna danych mikroprocesora 68000 jest 16-bitowa, a zastosowane układy pamięci dostarczają informację 8-bitową, zostały one przyłączone parami do bardziej i mniej znaczącego bajtu szyny danych. Mikroprocesor 68000 generuje dwa sygnały UDS/ (ang. Upper Data Strobe) i LDS/ (ang. Lower Data Strobe) sterujące rodzajem transmisji na szynie danych. Jeśli na obie linie podany został

poziom niski, powinien nastąpić 16-bitowy transfer. Jeżeli — przykładowo — tylko LDS/ jest w stanie logicznym 0, to transmisja odbywa się jedynie po mniej znaczącym bajcie szyny. Sygnały LDS/ i UDS/ wykorzystywane są w omawianym rozwiązaniu do kluczowania sygnałów wyboru bloku przychodzących z dekodera (IC5).

OBŚLUGA PRZERWAŃ

Mikroprocesor 68000 ma dwa tryby obsługi zewnętrznych przerwania. W pierwszym — urządzenie zgłaszające przerwanie powinno na mniej znaczącym bajcie szyny danych podać tzw. wektor przerwania, czyli informację identyfikującą przyczynę przerwania i wykorzystywaną przez procesor do wyznaczenia adresu początkowego procedury obsługi. Przez wymuszenie niskiego poziomu na wejściu DTACK/ mikroprocesor jest informowany, że urządzenie wprowadziło swój wektor na szynę danych.

W drugim trybie — wektor przerwania generowany jest automatycznie przez mikroprocesor w zależności od poziomu zgłoszonego przerwania. Poziomów przerwań jest siedem i są one określane na podstawie stanu logicznego trzech linii zgłoszeń: ILP2/, ILP1/ oraz ILP0/. Sekwencja logiczna 000 oznacza, że nie jest zgłaszane przerwanie; 001 oznacza, że jest zgłaszane przerwanie na poziomie pierwszym, itd. W tym trybie urządzenie zgłaszające przerwanie musi podać poziom niski na wejściu VPA/.

W chwili gdy procesor przyjmie zgłoszenie przerwania na liniach ad-

resowych A1...A3, pojawi się sekwencja logiczna odpowiadająca poziomowi przyjętego przerwania. Na pozostałych liniach adresowych procesor wymusza poziom wysoki.

W mikrokomputerze VU68K wykorzystywany jest drugi tryb obsługi przerwania. Niski poziom na wejściu VPA/ powstaje automatycznie, gdyż jest ono sterowane poprzez inwenter linią A23. Przerwania są zgłaszane przez układy 6850 z wykorzystaniem poziomów 1 i 2. W tym celu wykonane zostały podłączenia do ILP0/ i ILP1/, natomiast na ILP2/ jest stale podawany poziom wysoki (uwaga: sekwencja 000 oznacza, że na wszystkie wejścia podawany jest poziom wysoki). Przyjęte rozwiązanie daje możliwość niezależnej obsługi przerwania dla obu kanałów szeregowych.

Jak wynika z powyższego omówienia, mikroprocesor 68000 jest wyjątkowo „elastyczny” i bez większych problemów VU68K można zaadaptować do naszych krajowych warunków. Chyba warto spróbować!

AJP

LITERATURA

- [1] Bonds A. B., Carter E. M.: The VU68K Single Board Computer. BYTE, January 1984
- [2] MC68000 16-Bit Microprocessor User's Manual. MOTOROLA
- [3] MOSTEK 1982/1983 Microelectronic Data Book
- [4] MC68000 Technical Features. MOTOROLA
- [5] EF68000 HMOS 16-Bit Microprocessor. THOMSON-EFCIS.

Wielu czytelników zapewne będzie zbulwersowanych ilością miejsca, które poświęcamy dyskom elastycznym. Gdybyśmy jednak uzależniali nasze publikacje od szans zdobycia w kraju określonego sprzętu, to byłibyśmy zmuszeni do opisywania konstrukcji z czasów, gdy komputery budowano na przekaźnikach elektromagnetycznych. Wybierając rozwiązanie, w którym zastosowano nielutowy układ FDC (ang. Floppy Disc Controller) kierowaliśmy się względami ekonomicznymi, a nie płoną nadzieją, że może kiedyś któryś z krajów RWPG zdoła skopiować tego typu układ (nie wiedzieć czemu — wszyscy sądzą, że jeżeli już, to będzie to układ INTELA). Poniżej publikujemy rozwiązanie sprzętowe. Za miesiąc napiszemy trochę o oprogramowaniu.

Sterowanie napędami dysków elastycznych (1)

Na Zachodzie istnieje już standard dla pamięci dyskowych, który pozwala na odczytanie i zapis zawartości plików na sprzęcie różnych producentów. Standard ten dotyczy:

- gęstości ścieżek — 48 tpi (ang. track per inch — ścieżek na cal) lub 96 tpi
- gęstości zapisu — czyli częstotliwości zegara zapisu
- liczby ścieżek — 35, 40, 77, 80 lub 96
- prędkości obrotowej dla minidyskietek (ang. Mini Floppy) 5,25" — 300 obr./min.
- złącza sterownika z układami napędu (typ złącza oraz przydzielenie sygnałów)

— wielkości nośnika pamięci (dyskietek) — 5,25" lub 8" (ang. Maxi Floppy) oraz nowy standard 3,5" (ang. Micro Floppy) wprowadzony przez firmę SONY, a stosowany w mikrokomputerach 16-bitowych: LISA, MACINTOSH, IBM PC.

Opisany sterownik stacji dysków elastycznych projektowany był z myślą o podłączeniu do minidysków oraz dysków 8". Możliwe jest również sterowanie stacją dysków 3,5".

Sterownik umożliwia współpracę z czterema napędami dysków elastycznych, przy czym jedną z cech uniwersalności jest możliwość sterowania jednocześnie dyskami 8" i 5,25". Na przykład:

- dysk o nr. 0 — 8" SD — pojedyncza gęstość
- dysk o nr. 1 — 8" SD — pojedyncza gęstość
- dysk o nr. 2 — 5,25" SD — pojedyncza gęstość
- dysk o nr. 3 — 5,25" DD — podwójna gęstość.

O typie dysku i gęstości zapisu decyduje oprogramowanie.

Prezentowane rozwiązanie pozwala wykorzystać dowolny układ z rodziny WD 179X (lub SAB 179X). Różnice pomiędzy poszczególnymi reprezentantami tej rodziny zaprezentowane są w tabeli 1.

Sposób adresowania poszczególnych rejestrów FDC 179X przedstawiono w tabeli 2.

Jeżeli $A_2 = 1$, to przy wykonywaniu operacji zapisu trafia ona do Rejestru Wyboru (U11). Odczyt następuje natomiast z Rejestru Stanu (U4).

Rejestr Wyboru (ang. SELECT REG) zrealizowany został w oparciu o sześciokrotny przerzutnik D. Podczas zapisu poszczególne bity danych mają następujące znaczenie:

- (warunki: $A_2 = 1, A_1, A_0 = X, CLK = 0, WR/\overline{=0}$)
- b_0 — SEL 0 — wybór sterownika o numerze 0
 - b_1 — SEL 1 — wybór sterownika o numerze 1
 - b_2 — SEL 2 — wybór sterownika o numerze 2
 - b_3 — SEL 3 — wybór sterownika o numerze 3
 - b_4 — SD/DD — 0 — podwójna gęstość
1 — pojedyncza gęstość
 - b_5 — MINI/MAXI — 0 — maxi floppy — 8"
1 — mini floppy — 5,25"
 - b_6 — dowolna wartość
 - b_7 — dowolna wartość.

Uwaga: Tylko jeden z bitów $b_0...b_3$ może być jedynką logiczną (pozostałe muszą być zerem logicznym), bowiem tylko jedna ze stacji dysków może być aktywna w danym momencie.

Znaczenie poszczególnych bitów Rejestru Stanu jest analogiczne do linii FDC przyłączonych do U4 (rys. 1).

Sterownik napędów dyskowych jest wykonany w postaci oddzielnego modułu współpracującego z określonym typem magistrali systemu mikroprocesorowego. Dla innego typu magistrali należy proponowany układ odpowiednio zmodyfikować:

- ustalić, czy układ umieszczony będzie w przestrzeni adresowej pamięci („memory mapping”), czy też w przestrzeni adresowej układów we-wy
- ustalić adres bazowy modułu, rezerwując dla niego przestrzeń adresową ośmiu bajtów.

Układ sterownika pracuje z generatorem kwarcowym o częstotliwości podstawowej 8 MHz. Pozostałe częstotliwości są wytwarzane za pomocą licznika binarnego 7493 (U16). Sygnał $CLK = 1$ MHz dla „Mini” i 2 MHz dla „Maxi” jest wybierany bitem Q5 Rejestru Selekcji (U11). Sygnał RCLK jest wytwarzany w prostym, ale skutecznym układzie licznikowym, zrealizowanym na układach U15 i U14. Sygnał

Tabela 1. Charakterystyka układów z rodziny WD 179 X

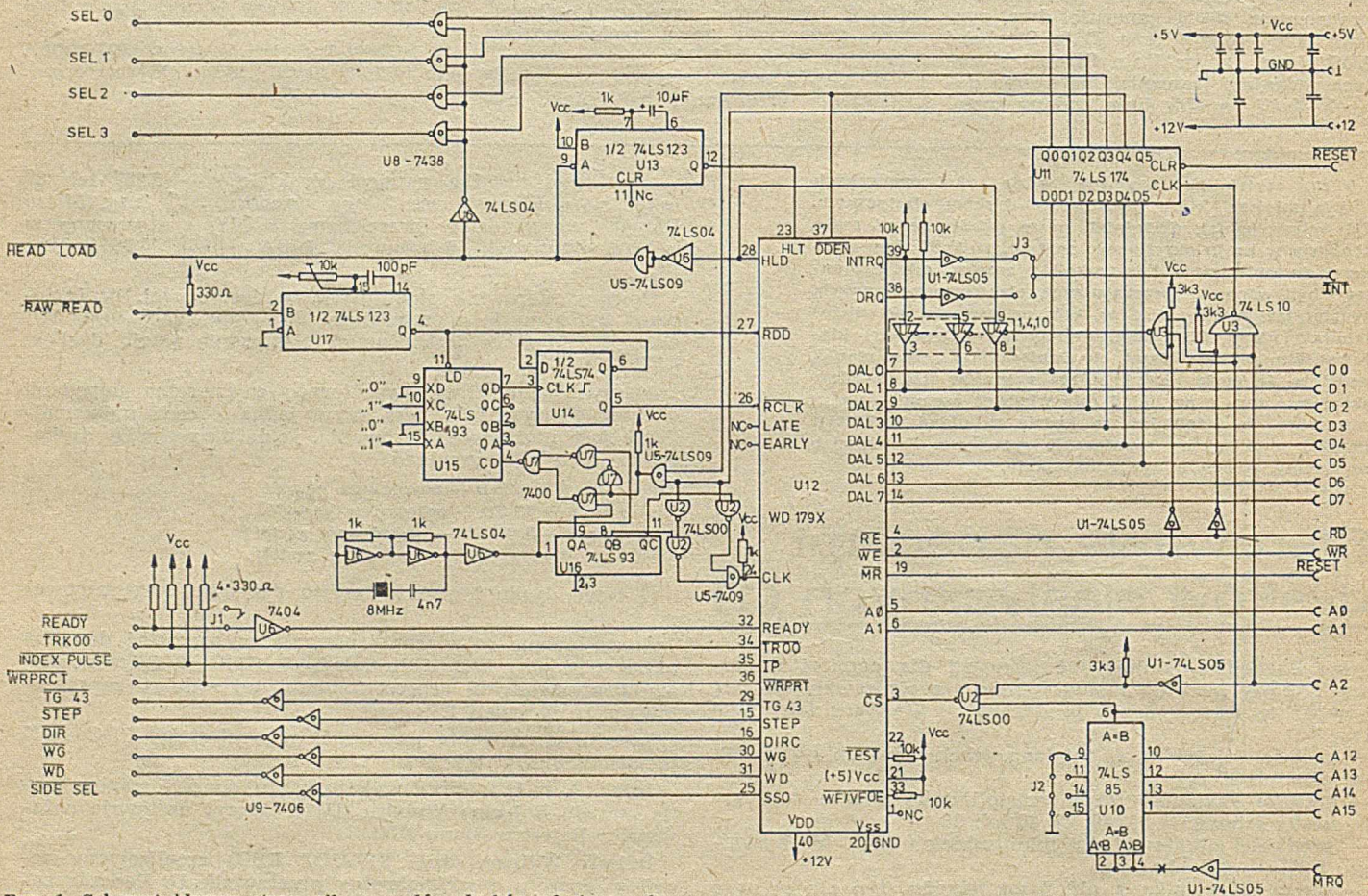
Możliwości \ Typ	1791	1792	1715	1794	1795	1717
FM/SD pojedyncza gęstość	×	×	×	×	×	×
MFM/DD podwójna gęstość	×		×		×	×
Prosta szyna danych			×	×		×
Zanegowana szyna danych	×	×			×	
Kompensacja przy zapisie	×	×	×	×	×	×
Wyjście wyboru strony (SSO)					×	×

FM — modulacja częstotliwości (umożliwia zapis o pojedynczej gęstości) — zgodna z formatem IBM 3740

MFM — zmodyfikowana modulacja częstotliwości (umożliwia zapis o podwójnej gęstości) — zgodna z formatem IBM 34

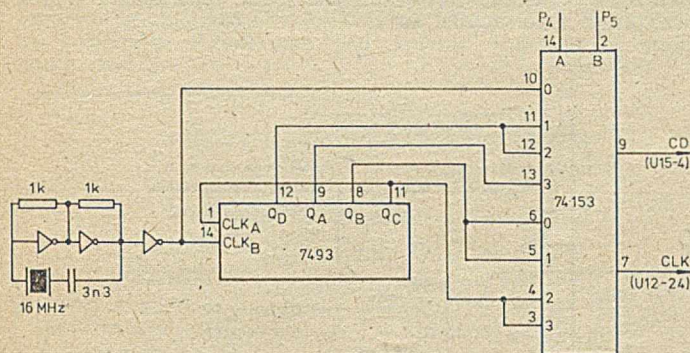
Tabela 2. Sposób adresowania rejestrów FDC 179 X

A_2	A_1	A_0	RE/ $\overline{=0}$	WE/ $\overline{=0}$
0	0	0	STATUS REG	COMMAND REG
0	0	1	TRACK REG	TRACK REG
0	1	0	SECTOR REG	SECTOR REG
0	1	1	DATA REG	DATA REG



Rys. 1. Schemat ideowy sterownika napędów dysków elastycznych

zegarowy jest dostarczany do układu U15 i multipleksowany przez układ bramek U7 i U5.

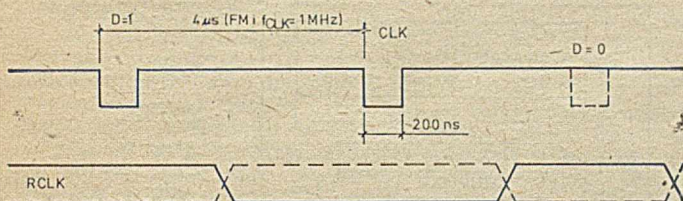


Rys. 2. Modyfikacja umożliwiająca realizację DD

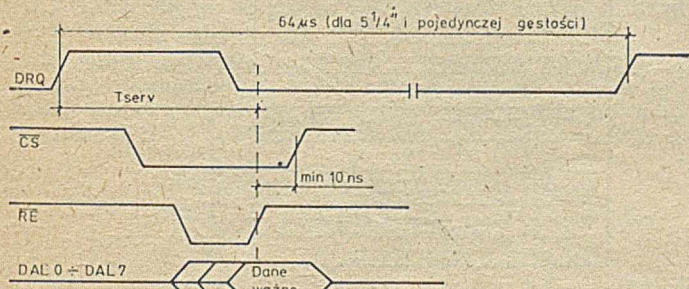
P_4 $\begin{cases} 0 - \text{podwójna gęstość} \\ 1 - \text{pojedyncza gęstość} \end{cases}$
 P_5 $\begin{cases} 0 - \text{dysk 8"} \\ 1 - \text{dysk 5,25"} \end{cases}$

Wersja pokazana na schemacie (rys. 1) nie nadaje się do realizacji odczytu z dysków 8" dla podwójnej gęstości. Problem ten można rozwiązać¹⁾ zmieniając częstotliwość zegara generatora kwarcowego na 16 MHz oraz wstawiając multiplexer. Fragment takiej poprawki pokazuje rys. 2.

Wszystkie wyjścia sterujące sprzęgu z napędem dyskierek powinny być typu: otwarty kolektor. Natomiast wejścia sygnałów przychodzących ze stacji dyskowych powinny być przez rezystor 220 lub 330 przyłączone do napięcia +5 V.



Rys. 3. Sygnał odczytany z dysku wraz z odtworzonym sygnałem zegarowym (polaryzacja sygnału RCLK jest nieistotna)



Rys. 4. Przebiegi przy odczycie danej z FDC; $T_{serv} = \max 47 \mu s$ dla „Mini”-SD; $T_{serv} = \max 23,5 \mu s$ dla „Mini” — DD

Opis Sygnałów FDC 179X

MR/ — zerowanie układu — reaguje na poziom niski¹⁾
 WE/ — niski poziom na tym wejściu (gdy CS/=0) powoduje zapis danych z linii DAL0...DAL7 do rejestru żądanego bitami A_1, A_0

¹⁾ W publikacjach zachodnich do realizacji odczytu MFM z dyskierek 8" zaleca się stosowanie układu z zamkniętą pętlą fazową; proponowane przez autora rozwiązanie działa, trudno jednak ocenić jego niezawodność — przyp. AJP

²⁾ Zapis MR/ jest równoważny z MR. Taką konwencję zapisu negacji zastosowano w tekście całego artykułu.

RE/ — jak wyżej, lecz — odczyt
 A_0, A_1 — linie wyboru rejestru
 CS/ — linia wyboru układu
 DAL0...DAL7 — linie szyny danych
 CLK — wewnętrzny zegar odniesienia dla zapisu lub odczytu informacji z dysku dla stacji 5,25" ($f_{CLK} = 1\text{MHz}$), natomiast dla stacji 8" $f_{CLK} = 2\text{MHz}$
 DRQ — żądanie odczytu lub zapisu danych (aktywne w stanie 1), zerowane po odczycaniu lub zapisie rejestru danych ($A_1 = 1, A_0 = 1$)
 INTRQ — zgłoszenie przerwania — (aktywne w stanie 1), ustawiane po zakończeniu wykonania każdego rozkazu FDC 179X; zerowane po odczycaniu Rejestru Statusu
 STEP — impuls przesuwający głowicę
 DIRC — kierunek przesuwu głowicy; „1” — głowica przesuwana w kierunku osi obrotu dysku; „0” — gdy głowica przesuwana w kierunku przeciwnym do osi obrotu dysku
 EARLY } kompensacja zapisu (tutaj nie wykorzystana)
 LATE }
 HLT — czas dociskania głowicy
 RCLK — sygnał zegarowy wydzielony z sygnału odczytywanych danych
 RD/ — sygnał odczytywanych danych
 HLD — sterowanie dociskiem głowicy do nośnika informacji (dyskietyki)
 WG — bramka zapisu aktywna podczas zapisu informacji na dysk
 WD — zapisywane dane — 250 ns (MFM) lub 500 ns (FM), impulsy zegara i danych; zapisywane dane mają specjalne znaczniki pół (format danych i format zegara)
 READY — gotowość stacji dysków do pracy; przy sterowaniu minidysków, które nie zawsze posiadają wyjściowy sygnał READY, można podłączyć Vcc (+5 V)
 TR00 — informacja o tym, że głowica znajduje się nad ścieżką zerową
 IP/ — informacja o położeniu fizycznego początku nośnika
 WPRT/ — informacja wprowadzona w momencie wysłania rozkazu zapisu („0” — zapis na dysk nie jest dokonywany)
 DDEN/ — wybór trybu dla pojedynczej lub podwójnej gęstości („0” — podwójna gęstość, „1” — pojedyncza gęstość).

Tabela 3. Opis sprzęgu napędów dyskowych „5,25”

Numer masy	Numer sygnału	Opis
1	2	
3	4	[Sterowanie — docisk głowicy] (ang. HEAD LOAD)
5	6	Wybór dysku 3 (ang. SEL 3)
7	8	Impuls odniesienia (ang. INDEX PULS)
9	10	Wybór dysku 0 (ang. SEL 0)
11	12	Wybór dysku 1 (ang. SEL 1)
13	14	Wybór dysku 2 (ang. SEL 2)
15	16	Załączenie dysku (ang. MOTOR ON)
17	18	Wybór kierunku przesuwu (ang. DIRECTION)
19	20	Impuls kroku (ang. STEP)
21	22	Dane zapisywane (ang. WRITE DATA)
23	24	Bramka zapisu (ang. WRITE GATE)
25	26	Ścieżka 0 (ang. TRACK 00)
27	28	Zapis zabroniony (ang. WRITE PROTECT)
29	30	Dane odczytywane (ang. READ DATA)
31	32	Wybór strony (ang. SIDE SEL)
33	34	[GOTOWOŚĆ] (ang. READY)

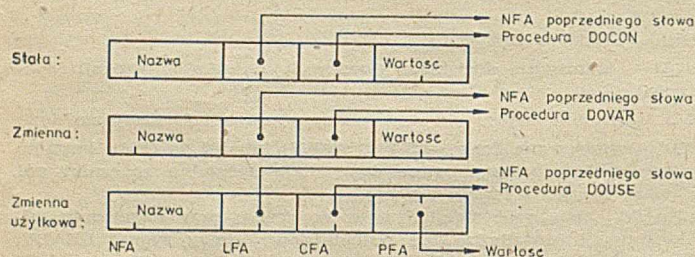
Sygnały opisywane w nawiasach kwadratowych występują nie we wszystkich napędach dyskowych. Przedstawiony zestaw numerów wyprowadzeń — i przyporządkowania im sygnałów — jest zgodny z powszechnie na Zachodzie stosowanym standardem ANSI.

To już ostatni odcinek materiału o implementacji fig-FORTH. Sądzi się, że zamieszczone informacje były dla zwolenników tego języka użyteczne lub przynajmniej ciekawe. Ci zaś, którzy wolą traktować swój komputer jak czarną skrzynkę, z pewnością ucieszy wiadomość, że już niebawem powrócimy do zapowiadanej niegdyś formuły kącika FORTH. Mamy już za sobą problemy natury szkoleniowej, możemy więc skupić się na rzeczach ciekawszych.

Jak zainstalować FORTH (3)

STAŁE, ZMIENNE I ZMIENNE UŻYTKOWE

Na rysunku 1 przedstawiono postać definicji katalogowych dla stałych, zmiennych i zmiennych użytkowych. Różnią się one odwołaniem do procedury wykonawczej, umieszczonej w polu kodu CFA, oraz znaczeniem 2-bajowego pola parametru PFA. Sposób definiowania i wykorzystania tych słów przedstawia tabela 1.



Rys. 1.

Definiowanie i wykorzystywanie stałych i zmiennych

Obiekt	Sposób definiowania	Pobranie wartości	Zmiana wartości
stała	wartość CONSTANT nazwa	nazwa	niedozwolona
zmienna	wartość VARIABLE nazwa	nazwa v	wartość nazwa!
zmienna użytkowa	numer USER nazwa kolejny	nazwa v	wartość nazwa!

Stała służy do przypisywania symbolom wartości, pamiętanych w polu PFA. Procedura DOCON, w przypadku odwołania do stałej, umieszcza jej wartość na stosie. Wartość stałej zostaje ustalona w czasie jej definiowania.

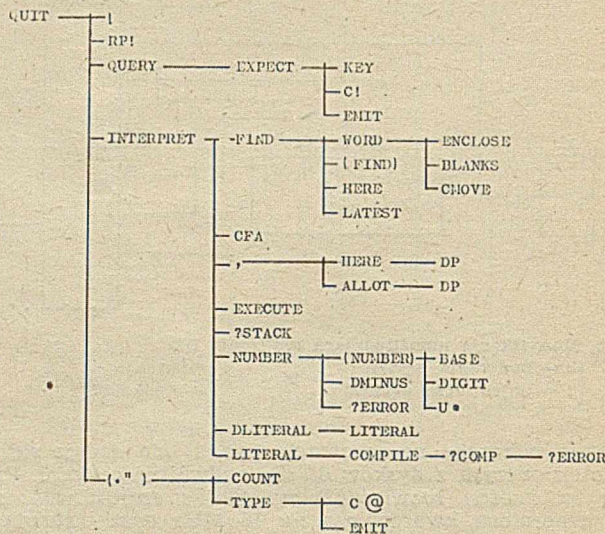
Zmiennej również nadaje się wartość w czasie definiowania. Można ją jednak zmieniać, gdyż odwołanie do zmiennej powoduje umieszczenie na stosie jej adresu, a nie wartości.

W polu PFA zmiennej użytkowej (rys. 1) podawany jest adres, pod którym umieszczono jej wartość. Podczas definiowania zmiennej użytkowej, adres ten ustalany jest wewnątrz przeznaczonych na ten cel obszaru USER AREA (na podstawie podanego kolejnego numeru). Wartości zmiennych użytkowych można pobierać i nadawać podobnie jak w przypadku zwykłych zmiennych; można też to zrobić przez wpisanie bloku danych do obszaru USER AREA. Taki sposób stosuje się przy nadawaniu wartości początkowych zmiennym użytkowym po wyzerowaniu systemu.

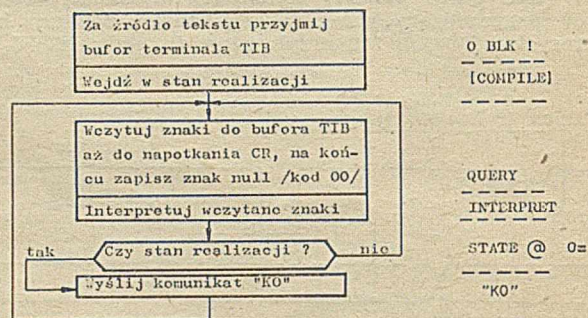
INTERPRETER ZEWNĘTRZNY

Interpreter zewnętrzny jest procedurą organizującą czytanie tekstu źródłowego i jego interpretację. Jest on jednym ze słów FORTH i nosi nazwę INTERPRET. W poprzednim odcinku przedstawiono ciąg definicji prowadzących do zdefiniowania tego słowa. Obecnie omówione zostanie dokładniej jego działanie. Na rysunku 2 przedstawiono hierarchiczne zależności pomiędzy słowem INTERPRET i innymi słowami kluczowymi. INTERPRET może być wywołany przez QUIT (rys. 3) albo przez LOAD (rys. 4).

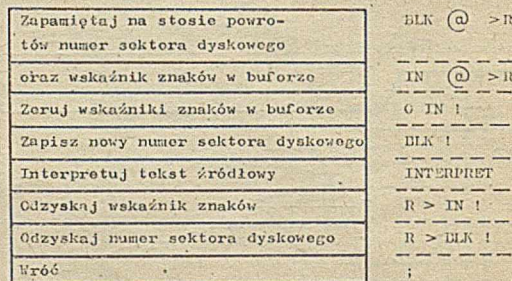
Do QUIT wchodzi się po wyzerowaniu systemu bądź po obsłudze błędu. QUIT jest nieskończoną pętlą; w każdym przebiegu pętli interpretowana jest jedna linia tekstu wprowadzonego z terminala.



Rys. 2.



Rys. 3. QUIT

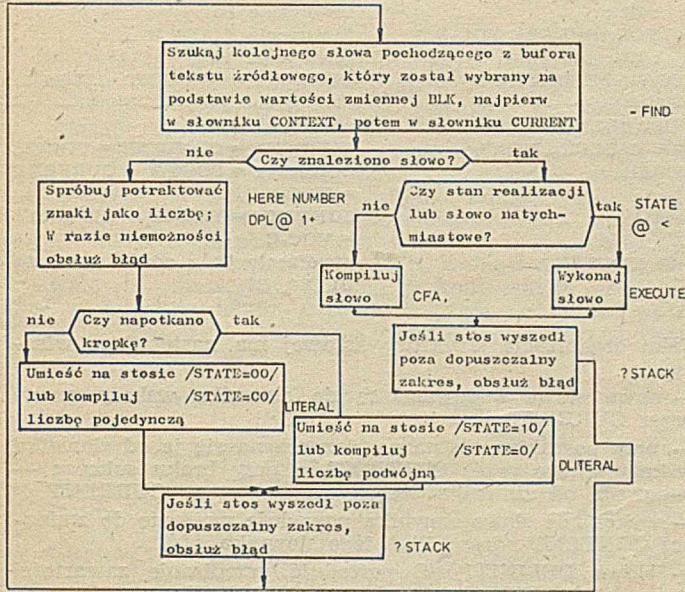


Rys. 4. LOAD

Zmienne STATE decyduje o sposobie interpretacji wczytanego słowa lub liczby. Początkowo system znajduje się w stanie realizacji (zmienne STATE = 0), zaś tekst źródłowy pobierany jest z terminala (zmienne BLK = 0). W stanie realizacji napotkane słowa tekstu źródłowego są wykonywane natychmiast. Stan systemu oraz źródło tekstu mogą ulec zmianie po wejściu do INTERPRET (rys. 5). Napotkanie w tekście źródłowym słowa: powoduje po jego wykonaniu (sterowanym słowem EXECUTE) przejście do stanu kompilacji (STATE = C0), który trwa aż do napotkania słowa; które powoduje przywrócenie stanu realizacji (STATE = 0).

Z kolei napotkanie słów n LOAD (rys. 4), gdzie n jest numerem sektora na dysku, powoduje podstawienie BLK = n, po czym następuje wywołanie INTERPRET i interpretacja słów z bufora pamięci przechowującego zawartość te-

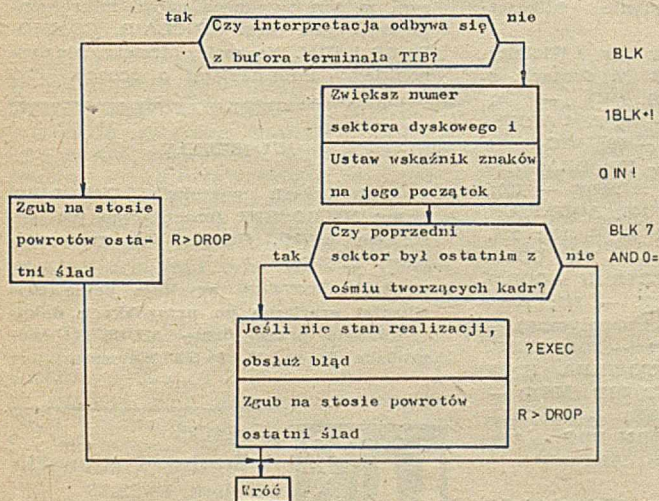
go sektora. Przy wyjściu z procedury LOAD przywracana jest poprzednia wartość zmiennej BLK — i następuje powrót do interpretacji z poprzednio używanego bufora. Dozwolone są zagnieżdżone wywołania LOAD, przy czym liczba poziomów ograniczona jest jedynie pojemnością stosu powrotów.



Rys. 5. INTERPRET

INTERPRET, podobnie jak QUIT, jest realizowany w pętli. Nie jest to jednak pętla nieskończona, choć na pierwszy rzut oka mogłoby się tak wydawać. Zarówno na końcu bufora terminala, jak i każdego z ośmiu buforów dyskowych, znajduje się znak null (kod 00). Jest on nazwą słowa, przedstawionego na rysunku 6. Wykonanie tego słowa powoduje — zawsze w przypadku bufora terminala, a w przypadku bufora dyskowego po zinterpretowaniu ośmiu bloków tworzących jeden kadr — zgubienie ze stosu powrotów ostatniego śladu i powrót na wyższy poziom według przedostatniego śladu, a więc do procedury, która wywoływała INTERPRET — QUIT lub LOAD. Wyjście z INTERPRET i w efekcie przejście do QUIT następuje też w przypadku napotkania błędu w czasie wykonywania procedur NUMBER bądź ?STACK.

W stanie realizacji liczba jest umieszczana na stosie, zaś w stanie kompilacji — kompilowana. Jeśli wśród cyfr czytanej liczby znajdowała się kropka, to jest ona traktowana jako liczba o podwójnej długości, a w zmiennej DPL zo-



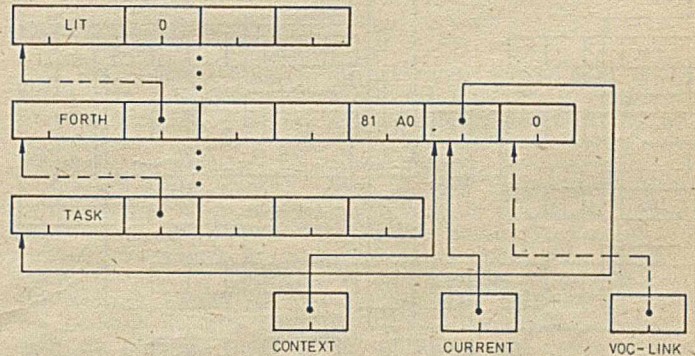
Rys. 6. Słowo null

staje zapamiętana informacja o pozycji kropki. Kompilacja liczby pojedynczej polega na zapisie w obszarze tworzonej definicji katalogowej adresu CFA słowa LIT i wartości liczby, a liczby podwójnej — na dwukrotnym wykonaniu tej czynności.

O sposobie interpretacji znalezionej w słowniku słowa decyduje wartość zmiennej STATE oraz bit poprzedzania (ang. precedence bit), czyli bit 6 w pierwszym bajcie pola NFA tego słowa. Jeśli bit ten ma wartość 1, to słowo jest „natychmiastowe” i — niezależnie od wartości STATE — jest wykonywane. Jeśli bit ten ma wartość 0, to przy STATE = C0 kompilowany jest adres CFA znalezionej słowa, zaś przy STATE = 0 słowo jest wykonywane.

LISTOWA STRUKTURA DEFINICJI KATALOGOWYCH

Wszystkie słowa FORTHA, zapisane w obszarze definicji katalogowych, zorganizowane są w postaci struktury listowej. Rysunek 7 przedstawia strukturę listy dla standardowej wersji fig-FORTH. Pierwszym elementem listy jest LIT, ostatnim — TASK. Lista zorganizowana jest za pomocą wskaźników (zobrazowanych strzałkami). Zmienne CONTEXT i CURRENT służą do znajdowania ostatniego elementu listy, przez wykonanie sekwencji CURRENT bądź CONTEXT



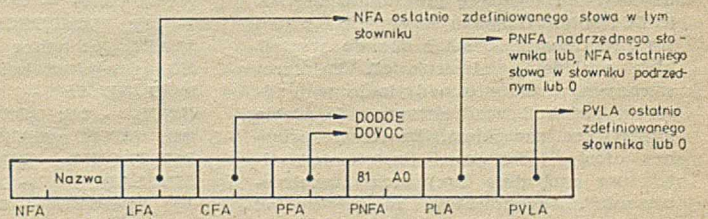
Rys. 7. Struktura listy dla standardowej wersji fig-FORTH

Słowo FORTH jest słownikiem. Strukturę definicji katalogowej będącej słownikiem przedstawiono na rysunku 8. W polu parametrów PFA wyróżniono tu trzy dwubajtowe pola:

- PNFA — ang. Phantom Name Field Address
- PLA — ang. Phantom Link Address
- PVLA — ang. Phantom Vocabulary Link Address.

Pola PNFA i PLA tworzą „widmowe” pole nazwy i pole wskaźnika. Z punktu widzenia procedury przeszukującej słownik (-FIND), bajty 81 i A0 zapisane w polu PNFA kodują pole nazwy 1-bajtowej będącej spacją (sposób kodowania nazwy omówiono w poprzednim odcinku).

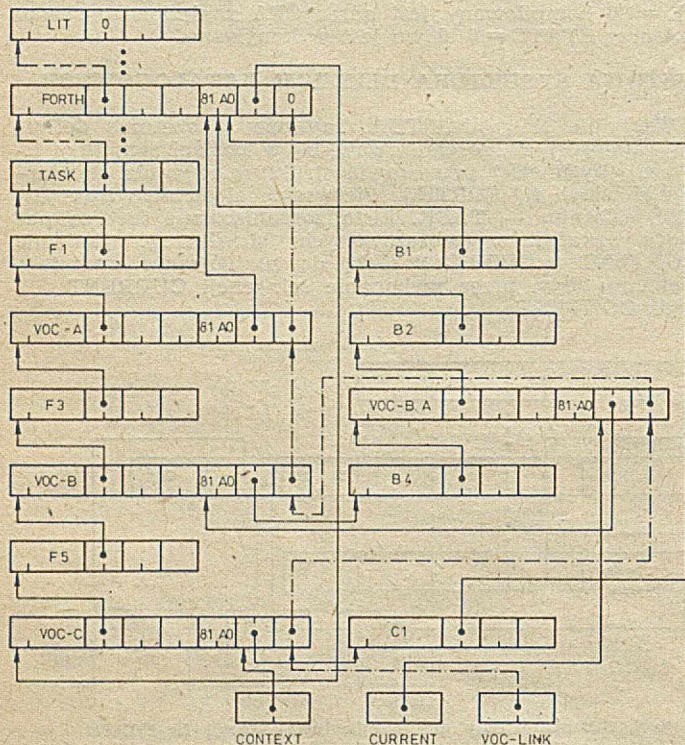
Kompilatory, czyli procedury tworzące nowe słowa (np., CONSTANT) dołączając nowe słowo do struktury listowej zawsze umieszczają je za słowem ostatnio zdefiniowanym, zapisując w polu LFA nowego słowa wskaźnik do CFA słowa dotychczas ostatniego i zmieniając wskaźnik w polu PLA słownika — tak, aby wskazywał adres CFA nowego słowa. Dzięki temu, pole PLA słownika na ogół zawiera wskaźnik do ostatniego słowa listy (patrz słowa FORTH na rysunku 7 i FORTH, VOC-C na rysunku 9).



Rys. 8. Struktura definicji katalogowej będącej słownikiem

* 10/84

Jak wynika z rysunku 9, listy mogą być ze sobą połączone tworząc strukturę drzewiastą. Początkiem nowej gałęzi jest zawsze słowo będące słownikiem. Po wykonaniu sekwencji {VOCABULARY nazwa} do aktualnej tworzonej listy dodawane jest słowo-słownik o podanej nazwie. Pole PLA tego słownika początkowo wskazuje pole PNFA słownika nadrzędnego (pole PLA słownika VOC-A na rysunku 9). Po dodaniu słów do nowego słownika, pole będzie wskazywało NFA ostatniego słowa nowej listy (pole PLA słownika VOC-B na rysunku 9).



Rys. 9.

Struktura przedstawiona na rysunku 9 powstała ze struktury z rysunku 7 przez wykonanie następującej sekwencji:

```
FORTH
DEFINITIONS
: F1 ;
```

```
CONTEXT = FORTH
CURRENT = FORTH
utworzenie F1
```

```
VOCABULARY VOC-A
: F3 ;
VOCABULARY VOC-B
VOC-B
DEFINITIONS
```

```
: B1 ;
: B2 ;
VOCABULARY VOC-B.B
: B4 ;
FORTH DEFINITIONS
: F5 ;
VOCABULARY VOC-C
VOC-C DEFINITIONS
FORTH
```

```
VOC-B
: C1 B1 B2 B4 ;
```

```
utworzenie VOC-A
utworzenie F3
utworzenie VOC-B
CONTEXT = VOC-B
CURRENT = VOC-B
utworzenie B1
utworzenie B2
utworzenie VOC-B.B
utworzenie B4
CURRENT = CONTEXT = FORTH
utworzenie F5
utworzenie VOC-C
CURRENT = CONTEXT = VOC-C
CONTEXT = FORTH, CURRENT
= VOC-C
CONTEXT = VOC-B, CURRENT
= VOC-C
utworzenie C1 z odwołaniami do
B1, B2, B4
```

Taka organizacja struktury listowej ma następujące własności:

- nowe słowa dodawane są do słownika wskazywanego przez CURRENT
- przy poszukiwaniu nazwy rozpoczyna się je od słownika wskazywanego przez CONTEXT, a przy braku sukcesu — ponownie od słownika wskazywanego przez CURRENT
- wykonanie słowa-słownika powoduje wpisanie do zmiennej CONTEXT adresu PLA tego słownika
- słowo DEFINITIONS powoduje przepisanie zawartości zmiennej CONTEXT do zmiennej CURRENT
- przy przeszukiwaniu list dociera się do wszystkich gałęzi nadrzędnych, ale nie dociera się do gałęzi bocznych (np. na rysunku 9 — rozpoczynając od CONTEXT osiągnię się słowa C1 spacja VOC-C F5 VOC-B F3 VOC-A F1 TASK ... FORTH ... LIT, a od CURRENT — słowa: spacja B4 VOC-B.A B2 B1 spacja VOC-C F5 VOC-B F3 VOC-A F1 TASK ... FORTH ... LIT)
- FORTH jest słownikiem nadrzędnym w stosunku do wszystkich pozostałych.

Główną zaletą drzewiastej struktury listowej jest możliwość pisania programów bez konieczności rezerwowania dla nich unikalnych nazw procedur podrzędnych. Różne procedury, umieszczone w różnych słownikach, nie pozostające w stosunku nadrzędna-podrzędna, mogą mieć te same nazwy.

ZBIGNIEW POJMAŃSKI
Warszawa

- Odbędzie się spotkanie organizacyjne KUM. Wyłoniono trzy podgrupy: profesjonalistów (mających coś do zaoferowania hobbystom), konstruktorów (budujących własny komputer) i programistów (posiadających mikrokomputer — najczęściej firmowy, zainteresowanych pisaniem i wymianą oprogramowania). Kontakt: Andrzej Droźniak, tel. w Warszawie: 41-26-01.
- Dzięki zasobom osób współpracujących z redakcją, jesteśmy w stanie umożliwić Czytelnikom wymianę programów dla ZX SPECTRUM. Dysponujemy już ponad 200 programami, chociaż nie do wszystkich posiadamy „instrukcje”. Zainteresowanych prosimy o nadsyłanie pod adresem redakcji zaadresowanej do siebie koperty ze znaczkiem, konieczne własnej oferty (programy, „instrukcje”, literatura zachodnia itp.) oraz „zapotrzebowania”. Przepraszamy za ewentualne opóźnienia w odpowiedzi, jeśli okazałoby się, że podobnie jak na poprzednie

oferty mikroKLANU wpłynęło kilkaset zgłoszeń.

- Ceny mikrokomputerów w Wielkiej Brytanii — listopad 1984 (w funtach, z podatkiem V.A.T.): ZX81 (1K) — 30; MC-10 — 50; SORD M5 — 50; AQUARIUS 1 — 50; ATARI 600XL — 90; VIC-20 — 90; TANDY CoCo (16K) — 100; DRAGON-32 (32K) — 100; ORIC ATMOS — 100; ZX SPECTRUM (48K) — 130; AQUARIUS 2 — 130; COMMODORE 16 — 140; ZX SPECTRUM+ (nowość!) — 180; COMMODORE 64 — 200; ELECTRON ACORN — 200; MEMOTECH MTX — 240; AMSTARD (+monitor monochromatyczny) — 240; (+monitor kolorowy) — 350; COMMODORE 4+ — 250; TOSHIBA HX10 (MSX) — 280; SONY HIT BIT (MSX) — 300; SANYO MPC100 (MSX) — 300; ALPHATRONIC — 347; ACORN BBC — 400; SINCLAIR QL — 400; ADVANCE 85A — 400; EINSTEIN — 500.
- W Budapeszcie przy ulicy Sörház mieści się sklep „VIDEO-COMPUTER”, w któ-

rym za forinty można nabyć mikrokomputery, m.in. firm SINCLAIR, COMMODORE i IBM. Na razie, niestety, nie dysponujemy informacjami o cenach.

OGŁOSZENIA

- Konrad Ladach poszukuje COBOLU-80 firmy MICROSOFT (pracującego pod systemem CP/M), Gdynia, tel. grzecznościowy 21-82-74.
- Drukarka DZM-180 uzyskuje pełne możliwości graficzne po przeróbkach dokonywanych przez firmę MUEL (Nowogrodzka 6a m. 17, 00-513 Warszawa).



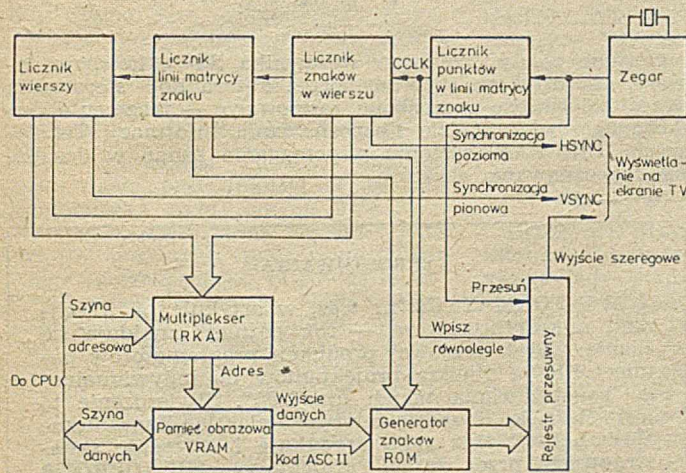
prowadzi:
Andrzej J. Piotrowski
tel. dom. 48-22-85
tel. służb. 43-66-11 w. 98

Sterowanie alfaskopem z zastosowaniem układu SY 6545

W związku z coraz powszechniejszym stosowaniem alfaskopów w mini- i mikrokomputerach wielu producentów opracowało układy scalone LSI sterujące wyświetlaniem na ekranie telewizora. Choć nie wyprodukowano dotąd w pełni „scalonych” alfaskopów, użycie specjalizowanego obwodu LSI pozwala zredukować liczbę układów scalonych potrzebnych do realizacji alfaskopu. Konsekwencją tego faktu są mniejsze wymiary urządzenia, mniejsza liczba połączeń lutowanych, wzrost niezawodności i zmniejszenie mocy potrzebnej do zasilania. Programowalność tych układów pozwala na rozszerzenie funkcji wykonywanych przez alfaskop, przy niewielkim wzroście kosztów. Omawiane układy mogą być także stosowane, choć w ograniczonym stopniu, w systemach graficznych.

ZASADY STEROWANIA ALFASKOPEM

Na rysunku 1 przedstawiono schemat blokowy klasycznego sterowania alfaskopem typu telewizyjnego, realizowany np. w technologii TTL MSI. Procesor lub jednostka centralna (CPU) przesyła do pamięci wizyjnej VRAM (będącej mapą obrazu wyświetlanego na ekranie) dane w postaci 6/8-bitowych słów, gdzie każdy bajt jest kodem (najczęściej ASCII) jednego znaku alfanumerycznego.



Rys. 1. Schemat blokowy typowego alfaskopu

Zespół liczników cyklicznie adresuje pamięć wizyjną tak, że na wejście generatora znaków podawane są kody kolejnych znaków alfanumerycznych. Równocześnie liczniki informują generator znaków o numerze aktualnie kreślonej linii rastra znakowego (rastra jednego wiersza). Generator znaków wpisuje równolegle do rejestru przesuwającego jedną linię rastra danego znaku alfanumerycznego, która jest przesyłana wraz z sygnałami synchronizacji do monitora CRT. Jeden wiersz danych (znaków) jest odczytywany z pamięci wizyjnej tyle razy, z ilu linii składa się matryca znaku.

Jednym z zasadniczych problemów, które stają przed projektantem alfaskopu jest organizacja bezkonfliktowego dostępu procesora (CPU) i sterownika alfaskopu do pamięci wizyjnej. Rolę organizatora współpracy pełni blok (rys. 1) RKA (rozstrzygnięcie konfliktu adresów). Istnieje kilka metod sterowania dostępem do pamięci które omówiono poniżej.

Realizację sterowania alfaskopem zilustrujemy przy użyciu układu scalonego SY 6545 produkowanego przez fir-

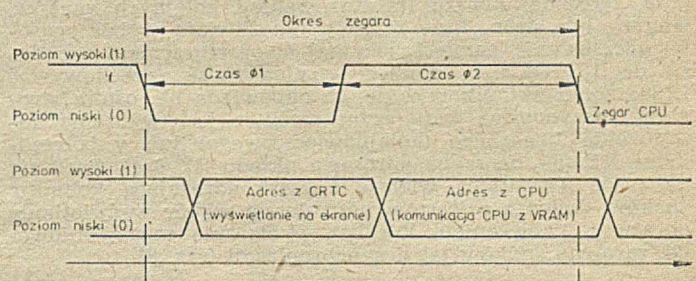
mę SYNERTEK, zbliżonego właściwościami do układu NC 6845 MOTOROLI lecz mającego większe możliwości. Układ SY 6545 steruje wyświetlaniem z rozdzielczością jednego znaku. Spełnia rolę liczników adresujących synchronizator znaków, generuje sekwencję sygnałów synchronizujących i współpracuje z CPU. Jest na tyle uniwersalny, że może współpracować z procesorami różnych firm.

Priorytet CPU

W tej metodzie każda łączność procesora z pamięcią obrazową przerywa arbitralnie dostęp sterownika do tej pamięci, a więc także — wyświetlanie obrazu na ekranie. Blok RKA jest w tym przypadku układem wybierającym źródło adresowania (np. multiplexerem). Ten sposób, choć prosty układowo, ma istotną wadę polegającą na zakłócaniu wyświetlania obrazu na ekranie w trakcie przesyłania danych między CPU i VRAM.

Podział faz zegarowych

W systemach z zegarem dwufazowym każdy cykl zegara składa się z dwóch oddzielnych faz $\Phi 1$ i $\Phi 2$ o zbliżonych czasach trwania. Faza $\Phi 1$ jest używana przez CPU do wygenerowania adresu i sygnału R/W (ang. Read/Write). Faza $\Phi 2$ służy do przesłania po szynie danych. Ponieważ w tych systemach szyna danych nigdy nie jest zajęta w fazie $\Phi 1$ cyklu zegara, może być wykorzystana do przydziału pamięci. W czasie $\Phi 1$ pamięć wizyjna jest adresowana przez sterownik, natomiast w czasie $\Phi 2$ przez CPU (rys. 2). Blok RKA, tak jak w poprzednim rozwiązaniu, może być multiplexerem. Zaletą tej metody jest brak zakłóceń na ekranie podczas dostępu CPU do VRAM. Ponieważ jednak dostęp do pamięci może trwać najwyżej pół cyklu zegara, użyta pamięć musi być w tym przypadku dwa razy szybsza niż w innych rozwiązaniach. Dodatkowym ograniczeniem jest fakt, że procesor i alfaskop muszą mieć jeden wspólny zegar lub dwa oddzielne — ale zsynchronizowane.



Rys. 2. Ilustracja pracy z podziałem faz

Wygaszanie pionowe

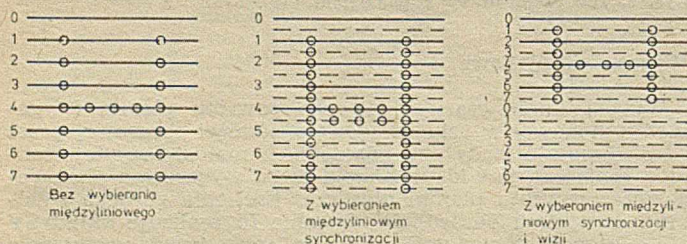
Wykorzystanie czasu pionowego powrotu plamki (wygaszanie pionowe) w celu dostępu do pamięci obrazowej przez CPU eliminuje wady obu poprzednich metod. Przy tej technice konieczna jest identyfikacja czasu wygaszania pionowego (czas powrotu plamki plus czas kreślenia górnego i dolnego marginesu obrazu). Wymaganie to realizuje układ SY 6545, który „zawiadamia” CPU o rozpoczęciu wygaszania pionowego. Tak więc komunikacja procesora z pamięcią wizyjną dokonyuje się podczas nieaktywności ekranu. Blok RKA może być multiplexerem. Wadą rozwiązania jest to, że przesłanie danych (znaków alfanumerycznych) do pamięci wizyjnej następuje tylko raz na 20 ms (jest to okres powtarzania półobrazu TV) i trwa zwykle do 3 ms, co może okazać się niewystarczające przy przesyłaniu dużej ilości danych.

Adresowanie pośrednie i wygaszanie

Adresowanie pośrednie zsynchronizowane z momentem pionowego i poziomego wygaszania plamki umożliwia dostęp CPU do VRAM przez zatrzaszki (ang. latch). Najpierw adres jest ładowany przez CPU do wewnętrznego rejestru adresowego SY 6545, który w czasie pionowego lub poziomego wygaszania zostaje podłączony do szyny adresującej pamięć wizyjną. W tej samej chwili następuje przesłanie kodu znaku z rejestru danych do pamięci obrazowej (lub odwrotnie). Blok RKA na zewnątrz SY 6545 jest reprezentowany tylko przez rejestr danych. CPU ładuje do niego (lub pobiera) dane prawie w dowolnym momencie. SY 6545 informuje CPU o stanie rejestrów (zajęte/wolne) i generuje sygnał zapoczątkowania łączności rejestrów z VRAM. Ze względu na realizację komunikacji CPU-VRAM w czasie nieaktywności ekranu, nie ma zakłóceń wyświetlania obrazu. CPU nie czeka na wygaszanie, bo ma praktycznie zawsze dostępne rejestry pośredniczące (może je zapełniać raz na 64 μ s). Wadą tej metody jest konieczność dodatkowego oprogramowania, gdyż adres musi być przesyłany w postaci dwóch bajtów danych (VRAM, w przeciwieństwie do poprzednich rozwiązań, nie jest włączana w obszar adresowy CPU). Przy przesyłaniu wierszy znaków układ SY 6545 pozwala na uproszczenie programu ładującego. CPU podaje adres pierwszego znaku w danym wierszu, po czym wewnętrzny rejestr adresowy (w SY 6545) wykonuje automatyczną inkrementację adresu po każdym wprowadzeniu kodu kolejnego znaku do pamięci obrazowej.

UŻYCIĘ STEROWNIKA SY 6545

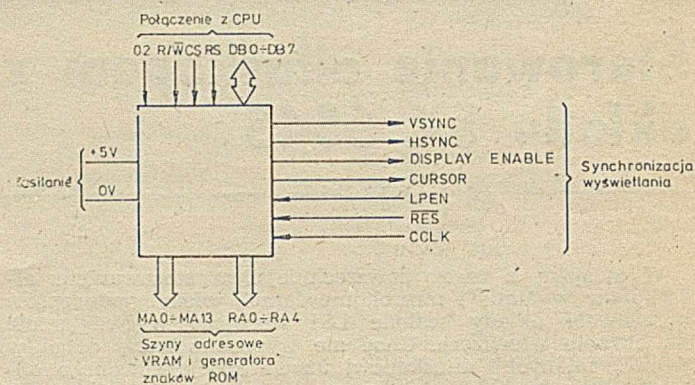
Scalony sterownik SY 6545 pozwala na budowę alfaskopu z zastosowaniem jednej z opisanych metod komunikacji, przez odpowiednią realizacją układową i zaprogramowanie pracy. Zawiera on szereg rejestrów wewnętrznych, dzięki którym CPU zadaje takie parametry obrazu, jak: czas trwania cyklu wyświetlania jednego półobrazu, czas trwania cyklu wyświetlania jednej linii, rozmiary obrazu (w granicach 128 wierszy po 256 znaków w każdym). Rejestry wewnętrzne umożliwiają także zaprogramowanie sekwencji sygnałów synchronizacji (najczęściej zgodnie ze standardem telewizyjnym), przy czym „przesuwanie” tych sygnałów w czasie powoduje zmianę pozycji obrazu na ekranie. Stosownie do użytego generatora znaków, można zadać liczbę kreślonych linii (maksymalnie 32) tworzących wiersz znaków alfanumerycznych. Układ SY 6545 pozwala ponadto na określenie rozmiaru i położenia znacznika (kursora), przy czym może on migać z dwiema częstotliwościami (do wyboru). CPU umieszcza w odpowiednich rejestrach (a także odczytuje) adres odpowiadający danej pozycji znacznika na ekranie. Rejestr, w którym umieszcza się dowolny adres pamięci wizyjnej (traktowany przez liczniki wewnętrzne SY 6545 jako pierwszy wyświetlany na ekranie) pozwala przez cykliczną wymianę jego zawartości na przesuwanie (ang. scrolling) obrazu (tekstu) według kolejnych znaków, wierszy lub stron. Inne rejestry umożliwiają wybór sposobu adresowania pamięci wizyjnej (z ograniczeniem rozmiaru do 16 KB, tzn. pamięć wizyjna może przechowywać 16 384 znaki alfanumeryczne, co odpowiada ok. ośmiu stronom maszynopisu), przy czym możliwe jest wybranie adresowania linearnego (kolejnowierszowego) lub współrzędnymi XY tak, że szyna adresowa układu SY 6545 dzieli się na część adresującą wiersze i część adresującą kolumny znaków. Jeden z rejestrów pozwala na wybór wyświetlania obrazu z wybieraniem lub bez wybierania międzyliniowego, zarówno znaków jak i synchronizacji (rys. 3).



Rys. 3. Wybieranie międzyliniowe

Do rejestrów, które pozwalają na odczytanie zawartych w nich informacji, należy rejestr przechowujący adres znaku wyświetlanego na ekranie w chwili, gdy plamka kreśląca linie obrazu oświetli fotoelement pióra świetlnego.

CPU może odczytywać także rejestr stanu sterownika informujący o aktualnej fazie sekwencji wyświetlania lub o żądaniu obsługi niektórych rejestrów.



Rys. 4. Połączenia zewnętrzne SY 6545

CPU komunikuje się z wewnętrznymi rejestrami układu za pomocą szyny danych oraz standardowych sygnałów sterujących: Φ 2, R/W, CS (rys. 4). Jedynym dodatkowym sygnałem jest RS używany w celu dostępu do rejestru, którego zawartość jest numerem rejestru wybieranego sygnałem CS. Sygnały synchronizacji HSYNC i VSYNC, sygnał CURSOR oraz DISPLAY ENABLE (aktywny podczas wysyłania danych na ekran) służą do utworzenia sygnału wizji. CCLK jest sygnałem zegara znakowego. Sygnał RES zeruje wszystkie wewnętrzne liczniki SY 6545 i może być użyty np. do synchronizowania częstotliwości „ramki” z częstotliwością napięcia sieciowego (synchronizacja pionowa). Można tu zastosować układ generujący RES w momencie przejścia przez zero sinusoidy napięcia sieciowego. Pozwoli to na uniknięcie drgań obrazu wynikających np. z niestałości częstotliwości napięcia w sieci energetycznej.

* * *

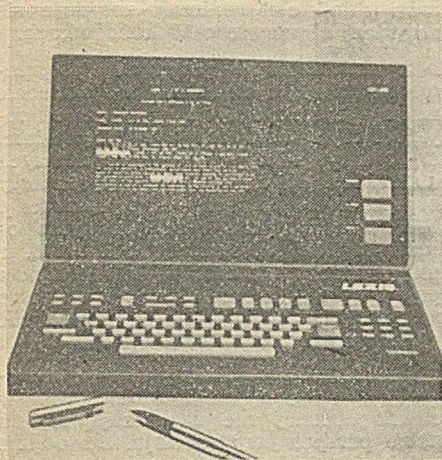
Opisane zalety scalonego sterownika SY 6545 spowodowały, że został on wybrany przez autora do sterowania pracą kolorowego grafoskopu rastrowego — fragmentu budowanego (w Zakładzie Rozpoznawania Informacji Tekstowych IBIB PAN) systemu wspomagania badań w dziedzinie rozpoznawania obrazów.

SPROSTOWANIA

W INFORMATYCE nr 6/84, w artykule pt. „Technika mikroprocesorowa — rok później” znajduje się zdanie: „Brak krajowej produkcji programatorów pamięci PROM zmusza projektantów do indywidualnego konstruowania takich urządzeń”. Stwierdzenie to jest nieścisłe. Przedsiębiorstwo Polonijno-Zagraniczne IMPOL-1 od 1982 roku produkuje i sprzedaje konstruowany przez zespół projektowy MICROPROCA uniwersalny, autonomiczny programator pamięci PROM PP-1. Do chwili obecnej dostarczono na rynek krajowy ponad 60 takich programatorów. Użytkownikami są m.in. MERAMAT, MERA-PNEFAL, MERA-ELZAB, MERA-STER, MERA-REFA IASE Gdańsk, PZL Rzeszów, EMA-APENA, LAMINA. Programator PP-1 to:

- sterowanie mikroprocesorowe, klawiatura i wyświetlacz, sygnał akustyczny
 - wymienne wkładki programujące (1702, 2708, 2716, 2732, 2764 i in.)
 - sprężki: równoległy, szeregowy V24, magnetofonowy
 - symulator pamięci ROM do wspomagania uruchamiania systemów mikroprocesorowych
 - kasownik pamięci UV EPROM.
- Cena zestawu z wkładką 2716 i kasownikiem pamięci UV EPROM wynosi 492 tys. zł.

Przykre niedopatrzenie spowodowało pominienie współautora artykułu „NEPTUN 184” zamieszczonego w INFORMATYCE nr 6/84. Pana ANDRZEJA RAJCE gorąco przeproszamy.



System LEXIS

LEXIS — system usług z zakresu informacji prawnych — obchodzi właśnie dziesięciolecie swego istnienia i w tym czasie rozrósł się — można powiedzieć — też dziesięciokrotnie. Rubin i Woodard podali w 1974 roku w swym raporcie [1], że bank danych systemu LEXIS zawierał wówczas ok. 4 GB informacji źródłowych; obecnie mówi się o zasobie objętości 40 G słów tekstu. Tygodniowy przyrost informacji tekstowej wynosi ok. 3 M słów. Sama baza komputerowa zlokalizowana jest w Dayton (stan Ohio), skąd firma MEAD DATA świadczy usługi w trybie interaktywnym na obszarze Ameryki Północnej, Wielkiej Brytanii, a ostatnio także Francji.

Dla porównania wielkości baz danych warto przypomnieć, że zachodniemiecka baza danych JURIS obejmowała początkowo 150 tys. dokumentów prawnych, przy czym mogło z niej korzystać tylko 40 zdalnych terminali. Obecnie system LEXIS obejmuje ponad 4 tys. terminali. Na terenie Wielkiej Brytanii ok. 5 tys. prawników (przy ogólnej liczbie 40 tys.) — to stali prenumeratorki usług LEXIS. System jest reklamowany jako największy tego rodzaju w świecie.

Opłaty za korzystanie z komputerowego przeszukiwania dokumentów prawnych są zróżnicowane w zależności od liczby prawników zatrudnionych w firmie, jak również od intensywności korzystania z systemu. Godzina taryfowa kosztuje w Wielkiej Brytanii od 40 funtów (przy ponad 60 prawnikach i ponad 35 godzinach taryfowych miesięcznie) do blisko 150 funtów (przy poniżej 20 prawnikach i 2 godzinach taryfowych). Do tego dochodzą koszty instalacyjne (100—300 funtów) oraz szkoleniowe (40—1600 funtów), nie licząc kosztów modemów telekomunikacyjnych, abonamentów oraz kosztów międzymiastowych telefonicznych połączeń do najbliższego miasta węzłowego systemu (Londyn, Paryż, określone miasta amerykańskie). Typowe biuro adwokackie płaci rocznie za usługi LEXIS ok. 6 tys. funtów.

Tam gdzie w grę wchodzi zyski, prenumerata tego rodzaju usług jest dosyć niechętnie ujawniana. Wiado-

mo wszystkie, że brytyjskie ministerstwa korzystają z LEXIS. Wiadomo również, że z LEXIS korzysta 12 największych londyńskich firm prawniczych. Nie wiadomo jednak, które z pięciu brytyjskich uniwersyteckich wydziałów prawa — na łączną liczbę 55 — nie korzysta z LEXIS. Nadal bowiem toczy się ostra walka o pozyskanie klientów; na horyzoncie pojawiła się konkurencyjna firma EUROLEX, a wiele instytucji informacyjnych — ze znaną agencją prasową Reutera na czele — nosi się z zamiarem rozszerzenia swych tradycyjnych usług o poradę prawne. Z drugiej strony — właściciele bazy LEXIS wkraczają na obszar działania agencji; w tym celu utworzyli system NEXIS dotyczący wyszukiwania notek i artykułów ukazujących się w stu najważniejszych dziennikach, czasopismach i biuletynach informacyjnych. Korzystanie z tego ostatniego systemu jest nieodpłatne dla instytucji będących już abonentami LEXIS, jednak upoważnienie do korzystania uzależnione jest od podpisania dodatkowych zobowiązań odnośnie do wykorzystywania zdobywanych informacji zgodnie z prawem autorskim.

Bank danych LEXIS składa się z kilkudziesięciu oddzielnych „bibliotek” — będących właściwie wąskotematycznymi bazami danych. W grupie baz amerykańskich na pierwsze miejsce wysuwa się tu Amerykańska Biblioteka Federalna, obejmująca decyzje Sądu Najwyższego od 1925 r. oraz decyzje Federalnego Sądu Apelacyjnego od 1945 r. Ponadto istnieje 51 bibliotek dla przepisów prawnych każdego ze stanów oraz 10 bibliotek przepisów ogólnych (podatki, ubezpieczenia, prawo cywilne, prawo patentowe, prawo autorskie, prawo pracy itp.).

Grupa baz francuskich to cztery biblioteki. Pierwsza z nich obejmuje decyzje Rady Stanu (od października 1964), Rady Konstytucyjnej (od grudnia 1958) oraz Trybunału Administracyjnego (od stycznia 1964). Druga — obejmuje decyzje Sądu Kasacyjnego (od października 1959). Trzecia — dotyczy umów międzynarodowych (ze skutkami finansowymi) oraz decyzji Europejskiego Trybunału Sprawiedliwości. Wreszcie czwarta — rejestruje akty publikowane we francuskim Dzienniku Urzędowym (od lipca 1955).

Jako stażysta Politechniki w Leicester miałem możliwość zapoznania się z systemem LEXIS od strony jurystyki brytyjskiej. Był to raczej zewnętrzny kontakt z systemem, który — będąc komercyjny — jest praktycznie nie do przeniknięcia od strony projektowej. Ze względu na poufność działań prawnych nie mogłem obserwować pracy konkretnego użytkownika; niemniej Mr Kyle Bosworth, dyrektor handlowy usług LEXIS i NEXIS, zorganizował w swym gabinecie pokazy najważniejszych funkcji systemów.

Abonenci mogą korzystać z usług wyłącznie poprzez specjalne terminale, indywidualnie zabezpieczone hasłami szyfrowymi, zaprogramowanym na stałe identyfikatorem instytucji będącej abonentem, oraz zmiennymi hasłami osób upoważnionych do korzystania z

bazy danych LEXIS. Są to nie tylko prawnicy, ale również księgowi, których często interesują zawiłe kwestie prawne związane z zawieraniem nietypowych umów.

Prawo brytyjskie jest zapisywane w bazie danych LEXIS za pomocą informacji z ponad trzydziestu regularnych wydawnictw prawniczych, takich jak: raporty sędziów pokoju, procesy patentowe, raporty Lloyd'a, raporty władz lokalnych, tygodniowe przeglądy prawnicze, sprawozdania sądów do spraw rodzinnych. W ten sposób baza rejestruje ogół spraw prowadzonych w sądach wyższej instancji Anglii oraz Walii od 1945 roku. Wyjątkowo dla spraw podatkowych — ze zrozumiałych względów niezwykle ważnych — sięgnięto wstecz aż do 1875 r. W bieżącym roku zostanie uwzględnionych w bazie kilka dalszych wydawnictw. LEXIS ma ponadto własne kanały informacyjne o sprawach wnoszonych do Izby Lordów bez publikowania, które są rejestrowane od 1 stycznia 1980.

Dotychczas wydano piętnaście biuletynów informujących o zasobach bibliotecznych, jakimi dysponuje system LEXIS. Oprócz wymienionych zbiorów, składających się na „bibliotekę ogólną” (ENGLISH GENERAL LIBRARY — ENGEN), a obejmujących także akty ogólnoprawne i przepisy sądowe — występują jeszcze: EURCOM (biblioteka Wspólnoty Europejskiej), UKTAX (wydzielona biblioteka podatkowa dla Zjednoczonego Królestwa), ENGIND (biblioteka przemysłu angielskiego), UKIP (biblioteka własności intelektualnych rejestrowanych w Zjednoczonym Królestwie) oraz ENGLG (biblioteka władz terenowych). Rzecz jasna, po tak ogromnym zbiorze bibliotek trudno się poruszać bez przeszkolenia na specjalnym kursie dla abonentów.

Ogólne zasady korzystania z systemów wyszukiwania informacji sprowadzają się do określenia tzw. słów kluczowych — które mogą pojawić się w tytule dokumentu prawnego albo dopiero w jego tekście (streszczeniu). System LEXIS dopuszcza bardzo szeroki repertuar wyszukiwań (ostatecznie przecież klient za to płaci), toteż wspomniane uprzednio szkolenie ma także ustrzec abonenta przed zbyt łapczywym wyszukiwaniem. Oto wzorcowy przykład [2]:

- Użytkownik, po przejściu przez procedurę sprawdzającą jego tajne hasło, zadeklarował zwrot wyszukiwawczy, np. SEARCH WARRANT (rewizja pełnomocnictwa). Po naciśnięciu klawisza TRANSMIT na ekranie ukazuje się liczba wyszukanych przypadków — 36 w tym przykładzie — oraz krótki instruktaż, co robić dalej.

- Widocznie użytkownik uznał to za nadmiar, ponieważ użył określenia dodatkowo precyzującego problem: AND CUSTOMS (oraz cel). Liczba wyszukanych przypadków wynosi już tylko 6.

- Taka liczba już zadawała użytkownika, toteż zdecydował się na przeszerzenie wyimków — naciskając klawisz KWIC. W odpowiedzi uzyskuje wyświetlenie na ekranie fragmentów wszystkich akapitów (do 25 słów przed i po słowie kluczowym) zawierających wskazane deskryptory. U góry ekranu

podane są symbole bibliograficzne, według standardowego wzoru (brytyjskiego), zrozumiałego natychmiast dla każdego prawnika.

- Użytkownik uznał wszystkie fragmenty za interesujące, toteż zażyczył sobie nowej funkcji: FULL — dającej możliwość wyświetlenia na ekranie pełnych tekstów.

- Po sporządzeniu notatek, użytkownik mógł zażądać sporządzenia kserokopii określonych stron z ekranu, a także uzyskać kompletny opis bibliograficzny — indywidualnie ponumerowany od 1 do 6.

Realizacja powyższego przykładu wymaga około 4 minut czasu taryfowego, a abonenta obciąża kwotą niespełna 5 funtów. Może to wydawać się drogą, ale radca prawny w ten sposób jest w stanie już w trakcie pierwszej rozmowy podać swemu klientowi (dyrektorowi) brakujące argumenty — których inaczej musiałby wyszukiwać całymi tygodniami, jeżeliby w ogóle dotarli do właściwych źródeł bibliograficznych.

System LEXIS wyraźnie rozrasta się i tworzy nowe rozgałęzienia (jak np. wspomniany NEXIS), a z czasem może połączyć się z innymi podobnymi systemami. Jest on bowiem wprawdzie największym systemem prawniczym, ale nie największym systemem baz danych. Tutaj palma pierwszeństwa należy chyba do już wymienionej agencji Reutera, która ma obecnie w 78 krajach świata ponad 13 tysięcy abonentów, dysponujących łącznie 50 różnymi systemami informacyjnymi z blisko 35 tysiącami terminali.

Generalnym wnioskiem z obserwacji różnych systemów wyszukiwania informacji jest „niejednoznaczność” rozwojowa. Niektóre z nich — takie jak LEXIS — nadal szybko rozrastają się, ale równocześnie znane są z ostatniego dziesięciolecia przypadki likwidacji nawet bardzo dużych baz danych, i to z bardzo różnych powodów. Najbardziej zaskakującym faktem był zamiar likwidacji przez firmę CDC swych pięciu baz danych prawnych, opracowanych od roku 1967 kosztem ponad 60 tys. robociznat — do celów wspomnianych procesów sądowych przeciw koncernowi IBM [3]. Inaczej mówiąc — nie ma gwarancji, że zakładana w jakimś kraju baza danych prawnych wytrzyma swój „stres rozwojowy. Niemniej — tym większe szanse na sukces ma się wówczas, gdy na początek taką bazę zaczyna się od stosunkowo wąskiej dziedziny prawodawstwa i stopniowo rozbudowuje się ją, sięgając do danych z coraz wcześniejszych lat, a dopiero potem — rozszerzając ją na inne dziedziny prawa.

ALLA A. R. MUHAMMAD
Łódź

LITERATURA

- [1] Rubin J. S., Woodard R. L.: LEXIS — A Progress Report. Jurimetrics Journal, vol. 15/No 2 (Winter 1974), pp. 86—89. Cit. za abstraktem systemu INSPECI No. C7522154
- [2] Legal Research with LEXIS. Butterworth Telepublishing Limited, London 1983
- [3] Retrieval System for Litigants. DATA-MATION, 1978.

Ewolucje układów

Wyprodukowany w 1971 roku pierwszy użyteczny mikroprocesor INTEL 4004, który zapoczątkował rozwój układów tej firmy, mógł przetwarzać zaledwie cztery bity, tzn. mógł dodawać, odejmować oraz wykonywać niektóre funkcje logiczne. Do tego potrzebnych było około tysiąca bramek logicznych na jednym płatku krzemu. Wprawdzie ten pradziad mikroprocesorów nie mógł jeszcze zbyt szybko działać, potrzebował kilku źródeł zasilania o różnych napięciach, to jednak zapoczątkował szybki rozwój mikroelektroniki.

Równoległe z mikroprocesorem 4004 powstawały pierwsze kostki pamięci o dostępie swobodnym, ze skromną pojemnością 256 bitów, oraz pierwsze typy pamięci stałej o tej samej pojemności. By zbudować system mikrokomputerowy, należało użyć wtedy około stu kostek.

Około 1971 roku pojawiły się również pierwsze kalkulatory kieszonkowe. Kalkulator, który „opanował” zaledwie cztery podstawowe działania arytmetyczne, kosztował jeszcze ponad 800 marek zachodnich. Przeprowadzone wtedy badania rynku wykazały nikłe zainteresowanie mikroprocesorem i mikrokomputerem. Wynikało z nich, że należy liczyć się z rocznym zapotrzebowaniem na ok. 10 tys. mikroprocesorów.

Kostki mikroprocesorowe

Popyt na mikroprocesory 4-bitowe wzrósł jednak gwałtownie i to w ciągu zaledwie jednego roku, a planowana roczna wielkość produkcji została sprzedana już po trzech miesiącach. Oprócz tego, coraz liczniejsze głosy domagały się stworzenia możliwości przetwarzania 8-bitowego, chociaż projektanci układów scalonych potrafili wtedy odtwarzać maskę prototypu mikroprocesora 4004 tylko ręcznie. Dzisiaj fachowcy zajmują się mikroprocesorami 32- i 64-bitowymi.

Już teraz spotkać można mikroprocesory 32-bitowe, które wprawdzie używają zewnętrznej szyny danych o formacie 16-bitowym, lecz wewnętrznie działają „na szerokości” 32 bitów. Co do rynku — czterech główni producenci zapowiedzieli już produkcję mikroprocesorów 32-bitowych: INTEL — model 80386, MOTOORLA — 68020, NATIONAL — 32032 oraz ZILOG — Z80000.

Kostki pamięciowe

Nie tylko mikroprocesory podlegają szybkim przemianom. Nowe tendencje zaznaczają się również w rozwoju technologii kostek pamięciowych. I tak, pamięci dynamiczne o dostępie swobodnym (ang. DRAM) są już osiągalne do pojemności 256 Kb, przy czym

cały układ elektroniczny znajduje się w jednej obudowie o 18 końcówkach.

W laboratoriach przemysłu półprzewodnikowego rozwijana jest obecnie nowa generacja tych pamięci o pojemności 1 Mb. Na rynku powinna się ona pojawić w ciągu najbliższych dwóch lat. Budując zatem przykładowy komputer z pamięcią o pojemności 1 MB, trzeba będzie użyć już tylko ośmiu kostek pamięci typu DRAM, a nie — jak dotychczas — 128 kostek o pojemności 64 Kb każda. Pamięć o pojemności 1 Mb będzie się znajdować w obudowie o 20 końcówkach.

Co do pamięci statycznych o dostępie swobodnym (ang. SRAM), to w chwili obecnej nie zaznaczają się tu nowe perspektywy rozwoju. Stosowane są kostki 16-Kbitowe w organizacji albo 2 K × 8 albo 16 K × 1. Dzięki technologii CMOS, następuje istotny spadek pobieranej mocy, a wszystkie nowe kostki wyposażane są w mechanizm określane jako „power down”. Polega on na tym, że jeżeli sygnał CS (ang. chip select) jest na poziomie wysokim (tzw. poziom H), to kostka znajduje się w stanie pogotowia (ang. stand-by), a pobór prądu jest rzędu 20 mA w technologii NMOS lub 0,1 mA w technologii CMOS. Jeśli sygnał CS przełącza się na poziom niski (tzw. poziom L), to kostka przechodzi automatycznie w stan aktywny, a pobór prądu zwiększa się odpowiednio do 160 mA lub 10 mA. To przełączenie ze stanu nieaktywnego (stand-by) w stan aktywny następuje bezpośrednio poprzez wejście CS.

Zaletą kostek DRAM jest niski koszt produkcji oraz małe zapotrzebowanie powierzchni. Przy tego typu pamięciach musi być jednak dodatkowo przeprowadzany proces odświeżania (ang. refresh), którego wykonanie podejmowane jest przez mikroprocesor lub specjalnie w tym celu zaprojektowany układ zwany sterownikiem DRAM.

Pamięć statyczna nie wymaga natomiast odświeżania, a więc posługiwanie się nią jest prostsze, natomiast koszty wytwarzania są stosunkowo wysokie.

Połączenie poszczególnych zalet legło u podstaw opracowania nowej wersji pamięci o dostępie swobodnym tzw. PRAM (ang. pseudo RAM) lub pamięci IRAM (ang. integrated RAM). Kostka taka na zewnątrz działa tak, jak pamięć statyczna o dostępie swobodnym, chociaż tzw. arbiter zarządza komórkami pamięci w oparciu o zasadę dynamiczną. Jej zaletą jest struktura o pojemności 64 Kb w organizacji 8 K × 8, zawarta w obudowie o 28 końcówkach. Wspomniany arbiter nie tylko zarządza pamięcią, lecz także przejmuje jej odświeżanie. Na zewnątrz uwidoczniła się zatem cecha pamięci statycznej. Jedyną wadą pamięci tego typu jest jej stosunkowo wysoka cena, wynosząca ok. 50 marek zachodnich za kostkę 64 Kb.

Pamięci stałe (ang. ROM) muszą się dopasować do 8- i 16-bitowej struktury danych w mikroprocesorach. Kostka ROM o pojemności 512 Kb (w organizacji 64 K × 8) nie stanowi już dzisiaj żadnego problemu technologicz-

nego. 524288 komórek pamięci oznacza ok. 1,5 mln elementów elektronicznych w jednej kostce i w jednej obudowie o 28 końcówkach.

Specjalnie dla rynku japońskiego zbudowany został generator znaków o pojemności pamięci 1082880 bitów w obudowie o 40 końcówkach. Sterowanie odbywa się tu za pomocą 14 przewodów adresowych, natomiast dla wyprowadzania danych istnieje 16 równoległych wyjść. W generatorze tym zapamiętuje się 3760 znaków piersarskich języka japońskiego, a wyjście przystosowane jest do wskazań aparatu pomiarowego o szerokości strumienia informacyjnego 18 × 16 bitów.

Podobna kostka ROM (o pojemności 1 Mb) istnieje również dla języka chińskiego.

Nowe technologie kostek pamięciowych

Rozwój o innym charakterze przebiega na rynku pamięci stałej programowalnej i wymazywalnej elektrycznie (EEPROM lub E²PROM). Okazało się bowiem, że zawartość komórki pamięci można wymazać nie tylko za pomocą — jak to było dotychczas — światła ultrafioletowego, ale również — co jest łatwiejsze w realizacji — za pomocą impulsu napięciowego, umożliwiającego wpisanie do niej nowej

zawartości. Dotychczas trzeba było w tym celu definiować w mikroprocesorze tzw. cykl programowania, natomiast obecnie funkcję tę kostka przejmuje automatycznie.

W magnetycznych pamięciach domonowych technologia scalenia doprowadziła do zadziwiających pojemności (do 4 Mb w jednej kostce). Pamięci te przyczyniają się — jak dotychczas — do powstawania jednak nieprzyjemnych błędów, które nie występują, gdy pojemność nie przekracza 1 Mb.

Oprac.

MARIAN LIPKA

na podstawie „CHIP”, nr 12/1983

Z KRAJU

56 Międzynarodowe Targi Poznańskie

Mikroinformatyki coraz więcej

Wystawianie podczas MTP wyroby stanowią dość obiektywne odbicie stanu naszej myśli technicznej, zwłaszcza konstruktorskiej. Dzięki wystawie poznańskiej można z pewnym przybliżeniem ocenić poziom krajowej technologii; bardzo trudno natomiast ocenić ekonomiczną stronę produkcji. Uważam, że 56 MTP były spotkaniem dość dobrze osądzonym w realiach. Przyjrzyjmy się więc ekspozycji.

Wszystkie dane zamieszczone poniżej pochodzą z materiałów firmowych, udostępnionych przez wystawców.

OFERTA POLSKIEGO PRZEMYSŁU KOMPUTEROWEGO

Krajowi wytwórcy sprzętu i oprogramowania wystąpili pod znakiem „MERA”. Dziś MERA jest Zrzeszeniem Producentów Środków Informatyki, Automatyki i Aparatury Pomiarowej — grupującym 24 przedsiębiorstwa przemysłowe, w tym 11 zajmujących się głównie informatyką. W bieżącym roku wystawa MERY znajdowała się w pawilonach 38/38A i 19. Większa niż poprzednio powierzchnia wystawy znacznie poprawiła warunki pracy obsługi i ułatwiła zwiedzanie. Dominował sprzęt związany z teleprzetwarzaniem:

- system komputerowy EC 1032 M
- procesor teleprzetwarzania EC 8371.01
- punkt abonencki EC 8578
- stacja ESPD 90 — CM 6904
- terminale MERY 9150
- system terminalowy MST 8000.

Ponadto pokazano dobrze znane urządzenia:

- minikomputer SM-4
- minikomputer MERA 60
- mikrokomputer ELWRO 513.

Zakłady Urzędzeń Komputerowych MERA-ELZAB (Zabrze) przedstawiły:

- System wspomagający projektowanie i uruchamianie mikrosystemów typu RTDS-8 (opis: INFORMATYKA nr 1/83, s. 5). Zgłoszono ten system do konkursu o Złoty Medal MTP.

- Monitor ekranowy MERA 7953N (CM 7209) — przeznaczony do pracy w systemach minikomputerowych. Ma on programowalną klawiaturę z wydzielonym polem numerycznym i sterowaniem kursora, odpowiadającą standardom ANSI i GOST. Ekran ma przekątną 15" i podwyższoną rozdzielczość. Jego pojemność wynosi 24 wiersze po 80 znaków; znaki są tworzone z matrycy 5 × 7 kropek. Jest sprzężony z modemu V24 (styk S2) oraz pętla prądowa (IRPS) 20/60 mA. Praca w kodzie ASCII (ISO-7). Szybkość transmisji można wybrać — od 75 do 9600 bitów/s.

- Monitor ekranowy MERA 7978 — przeznaczony jako terminal do dużych systemów komputerowych. Jest kompatybilny z monitorem IBM typu 3278, model 2 i 3. Ma ekran 15" o podwyższonej rozdzielczości. Format wyświetlania: 24 linie po 80 znaków lub 32 linie po 80 znaków + linia systemowa. Liczba znaków 256; są one tworzone na ekranie z matrycy 9 × 12 kropek. Klawiatura programowalna. Transmisja synchroniczna przewodem koncentrycznym z szybkością ok. 2,4 Mb/s.

- ComPAN-8 — mikrokomputer do zastosowań profesjonalnych. Zawiera on procesor oparty na mikroprocesorze 8085, pamięć operacyjną 64 KB RAM oraz 8 KB ROM. Przewiduje się możliwość rozszerzania tej pamięci do 1 MB; wtedy byłaby ona podzielona na strony po 64 KB, a te z kolei na bloki po 16 KB. Dołączona jest pamięć

na dysku elastycznym 8", drukarka mozaikowa z możliwością prezentacji graficznej oraz klawiatura. Wbudowany monitor ekranowy z niezależną pamięcią obrazową (16 KB) umożliwia wyświetlanie znaków i grafiki (200 linii × 640 punktów). Zastosowano płynny wysuw obrazu oraz podział ekranu na pola. Sprzęgi równoległe i szeregowo umożliwiają rozbudowę systemu. Przewiduje się specjalizowane moduły we-wy. Oprogramowanie jest oparte na systemie operacyjnym CP/M 2.2 lub ISIS II. Dostępne są edytory, asemblyery (8080 i Z80) oraz kompilatory i interpretery takich języków, jak: BASIC, FORTH, PASCAL, PL/M, FORTRAN. Rozszerzenie pamięci daje możliwość stworzenia „półprzewodnikowego dysku”, co przyspieszy pracę całego systemu i pozwoli na wykorzystanie wielozadaniowego systemu operacyjnego.

- Komputer osobisty MERITUM I (firmowany także przez Przedsiębiorstwo Polonijno-Zagraniczne ITM). Jest to model wyjściowy dla całej rodziny mikrokomputerów typu MERITUM. Ma on 14 KB pamięci ROM oraz 16/17 KB pamięci RAM. Oparty jest na mikroprocesorze Z-80. Pamięć obrazowa ma pojemność 1 KB; można wyświetlać 16 linii po 64 znaki lub 16 linii po 32 znaki lub semigrafikę o 48 liniach i 128 punktach w linii. Standardowa klawiatura, w obudowie której znajduje się sam mikrokomputer, wydaje dźwięk, potwierdzający wprowadzenie znaku. Sprzęż jest równoległy i szeregowy. Należy podkreślić, że układy we-wy — jak na tego typu komputer — dość rozbudowane. Buforowana magistrala, programowane liczniki (umożliwiające np. stworzenie zegara czasu rzeczywistego) skłaniają do stosowania MERITUM jako prostego sterownika. Pamięć zewnętrzna — na magnetofonie kasetowym. Można przłączyć zwykły odbiornik telewizyjny (kanały 1—6) lub specjalizowany monitor, np. typu NEPTUN 156. Oprogramowanie MERITUM I stanowi dość rozbudowany BASIC, zbliżony do wersji MICROSOFT. Wydaje się, że w pewnym stopniu MERITUM przypomina popularny komputer osobisty TRS-80. Należy podkreślić, że MERITUM zbudowano w oparciu o podze-

spoly elektroniczne dostępne w krajach RWPG, a jego cena jest niewysoka.

Kolejny model, MERITUM II, będzie miał pamięć operacyjną 64 KB, a także pamięć na minidyskach elastycznych z systemem operacyjnym CP/M. Więcej szczegółów na temat MERITUM Cytelnicy mogą znaleźć w artykułach: Z. Korgi, A. Smolińskiego, J. Lipowskiego i P. Podsiadło w Biuletynie Techniczno-Informacyjnym MERA, nr 10/83, oraz Z. Korgi w tymże Biuletynie, nr 5—6/84.

Centrum Naukowo-Produkcyjne Systemów Sterowania MERA-STER (Katowice) pokazało:

- Mikrokomputer MERA 60, który dzięki emulacji może pracować jako terminal inteligenty w sieciach komputerowych z komputerem centralnym typu RIAD lub IBM 360/370. Niezbędny jest system operacyjny RT 60 V.04 i oprogramowanie transmisji zgodne z protokołem BSC. MERA 60, wraz z dodatkowymi monitorami MERA 7953, może realizować funkcje terminala interakcyjnego IBM 3270 lub terminala wsadowego IBM 3780.

- Mikrokomputer 16-bitowy MERA 60-84, zgodny sprzętowo i programowo z MERA 60. Wymagany jest system operacyjny VT05. Opisywana wersja składa się z jednostki centralnej zawierającej na jednym pakiecie procesor, pamięć o pojemności 56 KB, „bootstart” sterowniki: drukarki, monitora ekranowego i pamięci na dysku elastycznym. Możliwe jest przyłączenie dalszych pakietów. Niezbędny jest dysk sztywny o pojemności 4,7 MB (np. MERA 9450), monitor ekranowy i drukarka mozaikowa (np. D-100).

Fabryka Mierników i Komputerów ERA (Warszawa) wystawiała systemy typu SM, z procesorem ELEKTRONIKA 100-25 lub SM-2104.

Zakłady Mechaniczno-Precyzyjne MERA-BŁONIE zaprezentowały:

- Drukarkę mozaikową D-200 (opis: INFORMATYKA nr 9/83, s. 28)

- Drukarkę mozaikową D-100. Jest to nowoczesna, mała (410 × 320 × 120 mm) i lekka (ok. 12 kg) oraz niedroga drukarka stołowa, sterowana mikroprocesorowo i napędzana silnikami skokowymi. Pracuje na typowym papierze obrzeźnie perforowanym o szerokości 10" lub węższym (Od 4 do 9,5"). Można stosować także papier zwykły, z rolki lub arkusza, szerokości do 8,5". Głowica drukująca — dziewięć igłowa; gęstość pozioma druku — 10, 12 lub 15,5 znaku/cal, a pionowa — 6,8 lub 10 wierszy/cal. Wersja „I” („inteligentna”) może dawać różne rodzaje druku: normalny, szeroki, wysoki, gęsty, o podwójnej intensywności, wyrazisty, lub ich kombinacje. Można drukować semigrafikę lub grafikę punktową. Wiele systemów mikrokomputerowych pokazywanych na tegorocznych Targach było wyposażonych w tę właśnie drukarkę. Można przypuszczać, że wkrótce D-100 będzie stosowana masowo.

- Minisystem MIXS TP — modularny i wieloprocesorowy system przeznaczo-

ny do przetwarzania tekstów i grafiki. Wykorzystano w nim graficzny monitor rastrowy MIXS GD oraz graficzną drukarkę mozaikową MIXS MP. System ten jest wyposażony w dyski sztywne i elastyczne, dużą klawiaturę, sprzęg komunikacyjny V24 oraz możliwość pracy w sieci lokalnej ETHERNET.

Krakowska Fabryka Aparatów Pomiarowych MERA-KFAP (Kraków) wystawiła:

- Jednostki pamięci na minidyskach elastycznych ED 501 D i ED 502 D (były krótko opisane w INFORMATYCE nr 9/83, s. 28)

- Pamięć na dyskach elastycznych 8" typu SP 55 DE (SM 5608) do systemów minikomputerowych SM-3, opartych na tzw. wspólnej szynie. Pamięć może również pracować z minikomputerem PDP 11/05.

- Pamięć na dyskach elastycznych SP 60 M do minikomputerów typu ELEKTRONIKA 60, ewentualnie do PDP 11/03.

- Minikomputer MK-45 — oparty na mikroprocesorze typu 8085, z pamięcią operacyjną 48 KB. Oprócz klawiatury ma on monitor ekranowy wyświetlający 24 wiersze po 80 znaków w wierszu, jednostkę pamięci na dyskach elastycznych, drukarkę mozaikową. System operacyjny IMPS (odpowiada znanemu CP/M 1.4) zapewniając możliwość korzystania z makrosemblera, FORTRANU, BASICU oraz oprogramowania komunikacyjnego BSC.

Warszawskie Zakłady Urządzeń Informatyki MERAMAT proponowały:

- Małą pamięć kasetową PK-3 (CM 5214) ze sprzęgiem typu SM. Mieści się ona w kasecie o wymiarach Eurokarty. Zapis według standardu ISO 3407 na kasecie typu „compact” (90 m nośnika o szerokości 3,81 mm) dwuścieżkowo, metodą PE. Prędkość zapisu — 254 mm/s, prędkość przeszukiwania — 1 m/s, gęstość zapisu — 32 bity/mm. Zasilanie + 5 V, - 5 V, + 12 V. Masa — ok. 1,5 kg.

Centrum Komputeryzacji Rynku CEKAR (Warszawa) pokazało kasy rejestracyjne z mikrokomputerami, powtarzając swoją ofertę z ubiegłego roku (INFORMATYKA 9/83 s. 28)

Spośród przedsiębiorstw zajmujących się głównie oprogramowaniem wystąpiły m.in.: MERA-SYSTEM (Warszawa), ZETO-ZOWAR (Warszawa), ZETO Katowice i ZETO Wrocław. Generalnie biorąc — oferowały one oprogramowanie do dużych systemów komputerowych lub do minikomputerów. Systemy mikrokomputerowe widniały wprawdzie w ich reklamie, ale były to jedynie opracowania wąsko specjalistyczne, dla konkretnych zastosowań.

Oprogramowanie ogólnoużytkowe dla mikrokomputerów oferowało ZETO Łódź w postaci biblioteki oprogramowania dla mikrokomputerów PITEM oraz konwersacyjnego systemu kompilowania i uruchamiania programów dla mikroprocesorów, napisanych w języku PL/M-80.

OFERTA NAUKI POLSKIEJ

Osobna, duża i zwarta ekspozycja zgromadziła kilkuset wystawców z Polskiej Akademii Nauk, wyższych uczelni, instytutów przemysłowych oraz placówek zaplecza badawczo-rozwojowego przemysłu. Większość wystawców prezentowała konkretne zastosowania mikrokomputerów, zwłaszcza do automatyzacji pomiarów, zbierania danych czy sterowania urządzeń. Były to rozwiązania specjalistyczne, adresowane do wąskiego, konkretnego kręgu odbiorców.

Warto odnotować dwie oryginalne formy organizacyjne:

- **Warszawskie Centrum Studenckiego Ruchu Naukowego** przy współpracy z Przedsiębiorstwem Polonijno-Zagranicznym KEBA (Warszawa) polecało usługi związane z oprogramowaniem komputerów, w tym projekty przemysłowych systemów informatycznych.

- **Dom Handlowy Nauki PAN** (Warszawa, ul. Miodowa 2; tel. 26-85-86) reklamował spółki cywilne usług informatycznych — oferujące oprogramowanie.

OFERTA FIRM POLONIJO-ZAGRANICZNYCH

Większość przedsiębiorstw „polonijnych” uczestniczących w tegorocznych Targach brała udział we wspólnej ekspozycji organizowanej przez Polsko-Polonijną Izbę Przemysłowo-Handlową. Oceniam, że ok. 10% ogólnej liczby wystawców „polonijnych” przedstawiło ofertę dotyczącą informatyki. Ta grupa wystawców nie jest jednolita; można ją podzielić na dwie części:

— firmy, w których wyroby dla informatyki nie są w zasadzie ukierunkowane na konkretnego odbiorcę, a zatem ich wytwarzanie odbywa się w sposób powtarzalny

— firmy, w których działalność dla informatyki jest związana z konkretnym zastosowaniem, wyrób jest wyspecjalizowany, a więc i wykonywany na jednostkowe zamówienie.

W obu przypadkach charakterystyczne jest wykorzystanie nowoczesnych elementów elektronicznych z krajów zachodnich. Warto przy tym zwrócić uwagę na fakt, że działalność przedsiębiorstw „polonijnych” w dziedzinie informatyki jest najczęściej łączona z produkcją „rynkową” (lampy, zabawki itp.).

A oto przegląd ekspozycji tych przedsiębiorstw:

Przedsiębiorstwo Zagraniczne AME-PROD (Poznań) pokazało:

- Moduły do rozbudowy popularnego systemu mikrokomputerowego SINCLAIR ZX 81, znacznie podnoszące jego właściwości użytkowe. Są to:

- AC 501, będący rozszerzeniem pamięci operacyjnej do 48, 52 lub 56 KB
- AC 502, będący modułem grafiki o wysokiej rozdzielczości; daje możliwość otrzymania na ekranie obrazu składającego się z 248 × 192 punktów; wymaga pamięci operacyjnej o minimalnej pojemności 16 KB, zawiera oprogramowanie własne w pamięci sta-

lej (2 KB); obszar jednej wyświetlanej strony wynosi ok. 6,2 KB

— AC 503, będący standardowym równoległym sprzęgiem typu CENTRONICS; oprogramowanie własne w pamięci stałej 1 KB; kod znaków ASCII; umożliwia kopiowanie ekranu normalnej i wysokiej rozdzielczości przy współpracy z drukarką SEIKO-SHA GP-100A

— AC 101, będący kaseta umożliwiającą używanie trzech wyżej opisanych modułów. Ma on również przycisk „RESET”, którego bardzo brakowało w ZX 81.

• Zestaw laboratoryjny ZLA-01. Jest to jednopłytkowy mikrokomputer — zbudowany z układów scalonych rodziny 8080. Umożliwia on realizację prostych funkcji użytkowych oraz uruchamianie niewielkich programów. Dzięki specjalnemu podręcznikowi, umożliwia prowadzenie zajęć szkolnych z techniki mikroprocesorowej. Zbudowany jest z mikroprocesora MCY 7880, pamięci EPROM obejmującej 2 KB programu monitora oraz 2 KB programu użytkownika, pamięci RAM 1 — 2 KB, sprzęgów równoległych i szeregowych. Urządzenie oparte jest na magistrali MULTIBUS. Zawiera klawiaturę z 25 klawiszami, sygnalizację optyczną i dźwiękową. Przewiduje się wyposażenie zestawu w dodatkowe pakiety: pamięci i wejścia-wyjścia, a także sprzęg dla drukarek i analizator stanów logicznych.

• Komputer osobisty AC-805 — uniwersalny mikrokomputer przeznaczony do obliczeń inżynierskich, automatyzacji laboratoriów, nauczania mikroinformatyki, a także do pracy w roli inteligentnego terminala. Ma on szerokie możliwości przyłączania urządzeń zewnętrznych — programowany uniwersalny 24-bitowy port równoległy oraz dwa szeregowo sprzęgi RS 232 C. Monitor ekranowy typu telewizyjnego wyświetla 24 wiersze po 32 znaki. Dostępnych jest 96 znaków ASCII oraz 32 znaki graficzne. Istnieje możliwość programowanego rozszerzenia repertuaru znaków o dalsze 128. AC-805 wyposażono w interpreter języka BASIC, odpowiadający wersji MICROSOFT. Zastosowano mikroprocesor Z-80, pamięć ROM o pojemności 12 KB oraz pamięć RAM o pojemności 16, 32 lub 48 KB. Zapis na magnetofon techniką FSK z szybkością 300 lub 1200 bodów. Wymiary: 350 × 400 × 90 mm. Typowa klawiatura.

• Większe systemy, oparte na AC-805, wymagają przyłączenia pamięci dyskowych:
— model 915 — sterownik czterech minidysków elastycznych o średnicy 5 1/4" i pojemności po 500 KB
— model 925 — sterownik dysku sztywnego typu WINCHESTER o pojemności 30 MB. AMEPROD wykorzystuje drukarki firmy SEIKOSHA.

Przedsiębiorstwo Zagraniczne API-NA (Zielona Góra) przedstawiło:

• Cyfrową pamięć obrazową typu CPO-1. Jest ona przeznaczona do zapamiętywania obrazu telewizyjnego. Zapamiętany obraz może być odtworzony na monitorze telewizyjnym. Urządzenie wyposażono w układ

do powolnego zapisu i odczytu informacji — aby umożliwić rejestrację obrazu na magnetofonie. Sygnał TV zgodny ze standardem (1 V_{pp} na 75 Ω, rozdzielczość obrazu — 256 × 256 punktów, z których każdy może mieć 16 poziomów szarości.

Przedsiębiorstwo Zagraniczne CA-REX (Warszawa) przedstawiło:

• Ploter (autokreślarka) typu CA X-5000 o modułowej konstrukcji i wymiennych sprzęgach. Zastosowanie mikroprocesora umożliwia umieszczenie w ploterze specjalizowanego oprogramowania. Dzięki temu pojedyncze instrukcje sterujące pisakiem powodują wykonanie powtarzalnych części wykresu, np. układu współrzędnych, skali czy napisów. Powierzchnia robocza pisaka jest formatu A4, można wybierać programowo cztery barwy elementów piszących. Ploter jest napędzany silnikami skokowymi. Najmniejszy krok adresowalny wynosi 0,1 mm, a powtarzalność położenia jest lepsza od 0,1 mm (bez wymiany końcówki piszącej) oraz lepsza od 0,2 mm z wymianą końcówki.

Przedsiębiorstwo COMPUTEX (Warszawa) zaprezentowało:

— mikrokomputer typu CS-80 (opis: INFORMATYKA nr 9/83, s. 31)

— język programowania INDUSTRIAL REAL TIME BASIC (opis jw.)

— pamięci półprzewodnikowe do różnych typów komputerów i minikomputerów

— mikrokomputer CS-305, który ma umożliwiać użytkownikom kontynuację prac prowadzonych w oparciu o minikomputer MERA 305

— pisak typu CS-101, zwany też COM PLOTTER (nowości!); ma on napęd za pomocą silników skokowych i może tworzyć rysunki formatu A3; najmniejszy krok wynosi 0,1 mm; powtarzalność pozycji wynosi 0,1 mm bez wymiany elementu piszącego i 0,2 mm z wymianą; proponuje się sprzęg równoległy typu CENTRONICS lub LOGABAX.

Przedsiębiorstwo DIGICOM-POLAND (Warszawa) oferowało:

— mikrokomputerowe systemy sterowania programowanego urządzeń technologicznych

— indukcyjne i pojemnościowe czujniki położenia do układów sterowania.

Przedsiębiorstwo IMPOL I (Warszawa) przedstawiło:

• Modułowy System Mikroprocesorowy MSM (opis: INFORMATYKA, nr 2/83, s. 17; nr 9/83, s. 31)

• Modułowy komputer osobisty MPC, przeznaczony do m.in. prac inżynierskich i biurowych, badań naukowych i automatyzacji laboratoriów, do zbierania danych i sterowania. Dzięki modułowej konstrukcji stwarza możliwość nadążania za wzrostem potrzeb użytkownika. Zawiera mikroprocesor INTEL 8085A, pamięć o pojemności od 8 KB RAM i 8 KB ROM do 64 KB RAM i 16 KB ROM. Monitor ekranowy 25 wierszy × 80 znaków w wierszu dla informacji tekstowej lub 200 linii × 640 punktów w linii dla grafiki, z możliwością przewijania (ang.

scrolling). Wykorzystuje moduły systemu MSM z magistralą BUSMAT. Oprogramowanie stanowi system operacyjny MSM-MOPS-E zawarty w pamięci EPROM, z oprogramowaniem pomocniczym i interpreterem języka BASIC. Przewiduje się dyskowe systemy operacyjne, odpowiadające CP/M i ISIS II.

Przedsiębiorstwo IMPOL II (Warszawa) przedstawiło:

• Mikrokomputer klasy „business-micro” z pamięcią na dysku elastycznym i systemem operacyjnym IMPS, odpowiadającym systemowi CP/M. Cały zestaw o nazwie IMP-85 był opisany w INFORMATYCE nr 9/83, s. 32.

• Adapter typu TLSA-10, który służy do zwielokrotnienia sprzęgu szeregowego V-24. Jego typowe zastosowanie to przyłączanie do dużego komputera przez jeden modem (i jedną linię telefoniczną) kilku terminali (do 10). Mogą to być np. terminale tej samej firmy oznaczone jako IMP 8502. Urządzenie działa w pełnym duplexie. Może też realizować transmisję synchroniczną z modemem asynchronicznym.

• Mikroprocesorowy adapter teleksowy, tzw. TELEX-BOX typu MAT-1. Jest to urządzenie służące do łączenia komputerów wyposażonych w sprzęg szeregowy z siecią teleksową. Od strony linii teleksowej urządzenie MAT-1 ma parametry zgodne z wymaganiami CCITT, a od strony komputera — spełnia wymagania dla typowego sprzęgu szeregowego. Dodatkowy sprzęg V24 umożliwia przyłączenie np. drukarki. Szybkość transmisji może zawierać się w przedziale od 75 do 19200 bodów. Ze względu na różnicę szybkości transmisji w linii teleksowej i w sprzęgu V24, zastosowano pamięć buforową o pojemności do 4 KB. Omawiany adapter pozwala na zdalny dostęp do komputera z dowolnego teleksu. Można wykorzystywać linię teleksową do komunikacji między komputerami, a także komputer jako „inteligentną” stację teleksową, co pozwala np. usprawnić nadawanie i odbiór korespondencji.

Przedsiębiorstwo ITM (Kraków) zaprezentowało:

• System uruchomieniowy z mikroprocesorem Z-80, oparty na systemie CAMAC, wyposażony w oprogramowanie zbliżone do systemu operacyjnego CP/M.

• Sterownik przemysłowy oparty na mikroprocesorze 8080 do sterowania maszyn i urządzeń. Istnieje specjalny język ułatwiający budowę programu sterującego. Sterownik obsługuje do 512 kanałów we-wy.

• Przenośne (walizkowe) urządzenie do zbierania danych analogowych, o nazwie ADCS-7C. Ma ono 7-kanałowy, 8-bitowy przetwornik analogowo-cyfrowy, o czasie przetwarzania 100 μs na kanał. Okres próbkowania od 2 ms do 1 s. Wyniki pomiarów są zapisywane do pamięci RAM, a następnie na taśmę magnetofonową — w ten sposób, aby taśma ta mogła być później przeczytana przez komputer MERITUM. Można więc pomiary wykonywać „w

terenie", a ich wyniki opracować w domu na komputerze za pomocą własnego lub firmowego programu.

Computer Studio Kajkowski — CSK (Gdynia) określając swoją działalność jako: „produkcja mikrokomputerów — oprogramowanie — doradztwo informatyczne”, specjalizuje się w problemach grafiki komputerowej. Reklama firmy była zamieszczona w **INFORMATYCE** (nr 4 i 5/84). Przypomnijmy w skrócie jej ofertę:

- rozszerzenie języka BASIC o instrukcje graficzne
- oprogramowanie wspomagające tworzenie grafiki trójwymiarowej
- oprogramowanie służące do zapisu cyfrowego dokumentów źródłowych
- system wspomagania projektowania obwodów drukowanych
- system wspomagania kreślenia rysunków technicznych maszynowych
- program redagowania tekstu z ploterem jako urządzeniem drukującym
- system zarządzania relacyjną bazą danych **BANK DANYCH — CSK**
- pakiet do sporządzania planów, zestawień i sprawozdań **TABPLAN-CSK**
- system redagowania oraz wyprowadzania tekstów i programów **TEKST-CSK**.

Sądzę, że firmy polonijno-zagraniczne dobrze wyczuwają słabości wielkiej MERY. Małe, dynamiczne zakłady szybko uzupełniają pojawiające się luki. Doskonałym tego przykładem są monitory ekranowe — oferowane przez firmy polonijne wprawdzie po dwukrotnie wyższej cenie niż monitory z ZUK-Zabrze, ale można je przynajmniej dostać w rozsądnym terminie. Podobnie, zbyt małe pamięci operacyjne do komputerów i minikomputerów dostarczane przez przemysł krajowy spowodowały, że co najmniej trzynaście firm polonijnych (**AMEPOL, COMPUTEX i DANPOL**) oferują półprzewodnikowe moduły pamięci o dużych pojemnościach do **ODRY 1305, R-32, MERY 400, MERY 9150**, a także komputerów **NOVA 840 i 1200**. Usłyszałem, że chętnie wstawiają pamięć również do **SM-0w**. Jak już wspominałem — efektem bezpośredniego współdziałania firmy „polonijnej” z przemysłem państwowym jest komputer osobisty **MERITUM**.

OFERTA KRAJÓW RWPG

BULGARIA

Przemysł bułgarski zaprezentował:
 • Mikrokomputerowy system bazowy typu **ISOT 0260**, połączony z systemem **CAMAC**. Jest on przeznaczony do zbierania danych i sterowania urządzeniami (np. technologicznymi), do automatyzacji prac inżynierskich i badań naukowych itp.

• Minikomputer **ISOT 2104C** — wielostanowiskowy system do wprowadzania i wstępnej obróbki danych, wyposażony w monitory ekranowe **EC 9005**.

• Jednostkę pamięci na dyskach elastycznych 8" — **EC 5074**. Wykorzystuje ona zapis jednostronny metodą podwójnej częstotliwości, ma szybkość transmisji 250 Kb/s i pojemność jednego dysku 3,2 MB.

• Jednostkę pamięci na dyskach elastycznych 8" z powiększoną gęstością zapisu **EC 5082**. Zapis metodą **FM/MFM**, parametry w porównaniu do **EC 5074** zostały podwojone. Oba typy pamięci mają jednostronny zapis dysku.

• Jednostkę pamięci na minidyskach elastycznych (5 1/4") **EC 5088** z jednostronnym zapisem metodą podwójnej częstotliwości, o pojemności 109,4 KB i szybkości transmisji 250 Kb/s. Dysk wiruje z prędkością 300 obr/min, zasilanie +12 V/0,8 A oraz +5 V/0,5 A, masa ok. 2 kg.

• Jednostkę pamięci na minidyskach elastycznych **ISOT 5050E** z podwójną gęstością zapisu o pojemności 218,8 KB i szybkości transmisji 250 Kb/s. Zapis jednostronny metodą **FM/MFM**; 35 ścieżek. Parametry mechaniczne i sposób zasilania — jak wyżej.

• Komputer stołowy z serii „if 800”. Rozwiązanie to jest wynikiem dostępu Bułgarii do zaawansowanej technologii, możliwej m.in. dzięki współpracy z firmą japońską **OKI**. Parametry urządzenia mieszczą się w klasie światowej. Zastosowano procesor **Z-80**, pamięć operacyjną o pojemności do 256 KB, oraz pamięć na dyskach elastycznych o pojemności 2 MB. Monitor ekranowy mieści 25 linii po 80 znaków lub 400 linii po 640 punktów, przy czym każdy punkt obrazu może być zaprogramowany w ośmiu kolorach — **videoRAM** o pojemności do 96 KB. Komputer ma wbudowaną drukarkę mozaikową. Można dołączyć także różne urządzenia we-wy — np. przez sprzęt szeregowy przyłączany jest modem lub ploter. Komputer jest wyposażony w system operacyjny **CP/M** z bardzo bogatą biblioteką programów, liczącą ponad sto pozycji.

CZECHOSŁOWACJA

W bieżącym roku Czechosłowacja nie wystawiała sprzętu komputerowego. Przedsiębiorstwo handlowe **KOVO** przedstawiło różne elementy i podzespoły elektroniczne, w tym również bardzo przydatne dla konstrukcji mikrosystemów.

NRD

Na tegorocznych Targach przedstawiono tylko elektroniczną maszynę do pisania (ze sprzęgiem **V24**) typu **S 6010**.

RUMUNIA

W minionych latach kraj ten raczej nie wystawiał urządzeń informatyki i zaawansowanej elektroniki. Tegoroczna oferta **ELECTRONUM** (bukaresztańskie przedsiębiorstwo eksportujące sprzęt informatyczny) była więc zaskakująca. A oto charakterystyka ważniejszych eksponatów:

• Mikrokomputer **ECAROM 880** do sterowania i nadzorowania przebiegiem procesów technologicznych, zrealizowany w standardzie Eurokart.

• Urządzenie **THETA ROM FD 5010** do automatycznego testowania pakietów z układami **TTL** i **CMOS**, oparte na minikomputerze **FELIX M-18**.

• Drukarka mozaikowa **ISM 150**.

• Monitor alfanumeryczny **VDT 40C**, będący odpowiednikiem **VDT 52** firmy **DEC**.

• Monitor ekranowy **DAF 2020**, odpowiednik **TEKTRONIX 4010**, o 24 liniach po 80 znaków przy wyświetlaniu tekstów lub 288 liniach po 512 punktów dla grafiki oraz prędkości kreślenia 15 μs/punkt.

• **DIAGRAM** — rodzina modularnych systemów graficznych przystosowanych do pracy autonomicznej lub współdziałania z dużym komputerem. Architektura systemu opiera się na dwóch magistralach: szynie głównej i szynie we-wy. W systemie może działać do sześciu procesorów. Zawsze występują procesor we-wy i procesor obsługi ekranu, generujący wektory. Można do nich przyłączyć: podwójny procesor graficzny, procesor arytmetyczny i procesor **Fourierowski**. W modelu **2030** grafika ma rozdzielczość do 512 × 512 punktów, natomiast w modelu **2040** do 1024 × 1024 punkty, z 8-bitowym kodowaniem koloru punktu. Pamięć systemowa **RAM** o pojemności do 2 MB, pamięć dyskowa — do 200 MB. System wyposażony jest w pełną klawiaturę i manipulator „joystick”. Można dołączyć drukarkę graficzną, pióro świetlne, sprzęg do aparatury elektronicznej (np. pomiarowej typu **IEEE 488**), szybkie przetworniki cyfrowo-analogowe (50 ns — do 8 bitów) oraz analogowo-cyfrowe (200 ns — 8 bitów). Całość działa pod nadzorem systemu operacyjnego **DIOS**. Dopuszcza on kontrolę nad 256 współbieżnymi procesami. Można korzystać z takich języków, jak **FORTRAN, LISP, EDISON**. Dzięki mikroprogramowaniu można emulować „**BASIC computer**”, szybko wykonujący programy napisane w języku **BASIC**. Pakiet graficzny jest podzbiorem standardu **SIGGRAPH**.

• Komputer **M-216**. Jest to maszyna 16-bitowa, oparta na mikroprocesorze **8086**. (Budziła ona tak duże zainteresowanie, że już drugiego dnia Targów nie można było otrzymać ulotek z jej danymi).

WĘGRY

Wystawiono:

• Rastrowy monitor graficzny typu **52121**.

• Mały system obliczeniowy z czterema stanowiskami roboczymi typu **VT 20/IV** (krótki opis: **INFORMATYKA** nr 9/83; s. 27).

• Jednostkę pamięci na ośmiocalowych dyskach elastycznych **MF 6400 D**. Liczba ścieżek na dyskietce — 2 × 77, pojemność dyskietki — 6,2 MB, zapis metodą **MFM** i **M²FM**, prędkość transmisji — 800—1600 Kb/s.

• Jednostkę pamięci na pięciocalowych minidyskach elastycznych typu **MF 1300/900**. Pojemność dysku — 109,4 lub 218,8 KB. Szybkość transmisji 125 lub 250 Kb/s. Zapis metodą **FM/MFM**. Liczba ścieżek — 35. Prędkość minidysku — 300 obr/min. Zasilanie: +12 V/1,9 A; +5 V/0,8 A. Wymiary: 83 ×

× 146 × 203 mm, masa — 1,5 kg. Jednostka ta jest zgodna z typem SHUGART SA 400.

• Komputer osobisty (w wykonaniu walizkowym) MINICOMP-C1 lub „transmic 8”. Zbudowany jest na procesorze Z-80, ma pamięć segmentową o pojemności do 512 Kb. Dwie jednostki pamięci na małych dyskach elastycznych. Całkowita pojemność pamięci zewnętrznej — do 1 MB. Ma wbudowany mały monitor ekranowy, wyświetlający 16 linii po 64 znaki. Przewiduje się dla tego systemu kartę z 16-bitowym mikroprocesorem M 68000. Komputer jest wyposażony w zegar czasu rzeczywistego i umożliwia dołączanie sprzęgu IEC-BUS. Istnieje również sprzęg równoległy do drukarek typu CENTRONICS oraz sprzęg szeregowy umożliwiający pracę w sieciach komputerowych. Oprogramowanie jest zgodne z systemami opracowanymi dla komputerów osobistych typu TRS-80. Systemy operacyjne — TRSDOS, NEWDOS i CP/M 2.2.

ZSRR

Przedstawiono:

— minikomputer typu SM 1300 w wersji dwuprocessorowej, przeznaczony głównie do sterowania procesami technologicznymi

— makietę komputera EC 1061; według materiałów informacyjnych jest to model o największej mocy obliczeniowej z serii RIAD o szybkości 1,5 mln operacji/s i pamięci operacyjnej do 8 MB. Istnieje możliwość przyłączenia procesora macierzowego.

* * *

Uważam, że nasze 8-bitowe systemy osiągnęły fazę znacznej dojrzałości technicznej, a niektóre z nich zbliżają się do szczytu tego, co można osiągnąć przy słowie 8-bitowym. Obserwujemy początek rozwoju małych systemów 16-bitowych.

Bardzo krytycznie, wręcz alarmująco, należy ocenić stan krajowej bazy podzespołów elektronicznych, a zwłaszcza układów o dużym stopniu scalenia. Cóż z tego, że polski „8080” dostał nagrodę Mistrza Techniki, skoro z krajowych elementów nie można złożyć żadnego sensownego mikrosystemu? Wobec trudności w dostawach elementów elektronicznych z krajów strefy dolarowej, sytuację ratują układy scalone dostępne w krajach RWPG. O ile wiem, żaden jednak z krajowych mikrosystemów nie jest dziś produkowany w dużych ilościach.

Brak u nas — moim zdaniem — możliwości kupna gotowego mikrokomputera jednopłytkowego (ang. single board computer — SBC). Można byłoby wbudowywać go do różnych urządzeń, wyposażając je w „inteligencję”. Bez tej cechy nie można dziś zbudować nic nowoczesnego i przystającego do europejskiego poziomu. Sy-

stemy modularne, mające swoje zalety, są jednak duże, skomplikowane i drogie. Nie rozwiązują one problemu szerokiej, prostej i taniej aplikacji mikrokomputerów. Zastosowanie mikrokomputerów jednoukładowych (ang. single chip, np. 8048/8748/8035) też jeszcze nie jest masowe, choć znaczącym faktem jest ich wykorzystanie w nowoczesnych drukarkach mozaikowych.

Nie zauważyłem ożywienia w dziedzinie oprogramowania dla mikrokomputerów. Z jednej strony widać tu tak olbrzymią organizację jak ZETO, zainteresowaną „z natury” oprogramowaniem dużych i drogich systemów i — moim zdaniem — nie mającą żadnej perspektywicznej koncepcji. Z drugiej strony — małą firmę Computer Studio Kajkowski, która ma jasny kierunek działania i sprzedaje konkretne oprogramowanie. Pomiędzy tymi biegunami jest kilka propozycji pośrednich, ale oprogramowanie jest w nich wyraźnie dodatkiem do innej działalności. Pozwólę sobie tu na uwagę, że przedsięwzięcie o nazwie MERITUM będzie o tyle udane, o ile zapewni się dla tego komputera dużo oprogramowania podstawowego, a także wiele programów użytkowych (gry edukacyjne, grafika, redagowanie tekstów itp.). Dodatkowym warunkiem jest zapewnienie rozsądnych sposobów obrotu handlowego tym oprogramowaniem.

Wielu użytkowników mini- i mikrokomputerów narzeka na kłopoty z serwisem. Najpoważniejsze problemy są z naprawą i regulacją pamięci na dyskach elastycznych. Dostawca całego systemu najczęściej nie chce nawet dysków dotykać, a producent jednostek dyskowych nie jest zainteresowany sprzedażą części zamiennych czy całych jednostek (zapasowych). Nie mówię już o serwisie tychże pamięci.

Ciekawe, że firmy polonijne wykozystwały jeden ze słabych punktów państwowego przemysłu komputerowego, oferując rozszerzenia pamięci operacyjnych. Nie słyszałem natomiast, aby ktoś oferował naprawy sprzętu nie wytwarzanego przez siebie, a zwłaszcza urządzeń peryferyjnych. Problem napraw nie był do tej pory problemem społecznym, wobec względnie niskiego stopnia komputeryzacji naszego kraju. Upowszechnienie mikrokomputerów, aczkolwiek hamowane przez słabość naszej elektroniki i w pewnym stopniu — mały wybór i wysokie ceny oprogramowania, musi jednak wreszcie nastąpić. Wtedy naprawy sprzętu staną się problemem podstawowym.

Krajowa mikroinformatyka dość doładnie powtarza drogę rozwoju zachodnich mikrosystemów. Czy jesteśmy w stanie iść w tej dziedzinie własną drogą? Pokażą to następne spotkania Targowe.

JACEK ŻEBROWSKI

*

Szlaban dla mikro

Gdyby sądzić według powierzchni, jaką na Targach zajmowała prezentacja polskiej oferty informatycznej — jest to jedna z wiodących branż naszego przemysłu. Zręczna mistyfikacja — pomocna dla reporterów telewizyjnych (hektary komputerów na małym ekranie...), pozwalająca decydom spać snem sprawiedliwych. Oczywiście nie zamierzam nikogo ganić za to, że pościwił informatyce dużo miejsc. W krajowych realiach nawet fakt, że została uznana za dziedzinę godną takiej mistyfikacji napawa optymizmem.

Rozpatrując ofertę państwowych producentów sprzętu mikroinformatycznego, należy jednak umiejętnie rozdzielać fakty od pobożnych życzeń. Abstrahując więc nawet od parametrów oferowanego sprzętu, wielkość produkcji pozwala obiektywnie rozstrzygnąć czy dany produkt jest miłym. Nie mamy jeszcze danych z tego roku, ale za Biuletynem MERY zacytujemy niektóre liczby z ubiegłorocznej produkcji (w sztukach):
— napędy dyskowe (PL × 45D) — 1589
— monitory ekranowe (7957 VGD) — 780

— drukarki (wszystkie razem) — 8376
— systemy RTDS 8 — 15 (!)
Niewtajemniczonym zdradzimy, że jest to produkcja fabryk. Od tego trzeba jeszcze odjąć eksport — aby przekonać się, co pozostaje jako oferta dla krajowej informatyki. A w tym roku nie należy spodziewać się drastycznego zwiększenia produkcji.

Jak wyjaśnił rzecznik prasowy ME-RY, dr inż. J. Dyczkowski, za taką wielkość produkcji należy w znacznej mierze winić... prasę. Otóż lamenty na temat stanu polskiej informatyki wyrobiły w decydentach przekonanie, że nie jest to dziedzina godna inwestycji. Ten paradoksalny osąd ma niestety duże znamiona prawdopodobieństwa, choć nie prasę należy winić za błędy w zarządzaniu. Dodatkowym argumentem przeciwko rozwijaniu informatyki jest przeświadczenie decydentów o niewystarczającym intensywnym wykorzystaniu już istniejącej bazy. Przy tego typu argumentach faktycznie mogą opaść ręce. Temu, że Pan X wykorzystując „układy” zakupił dla fasonu komputer, przeciwdziałała się przez ograniczenie produkcji do takich rozmiarów, że tylko podobnie uprzywilejowani klienci (z reguły to właśnie ci od fasonu) będą je mogli nabyć.

Innej natury rewelacją jest fakt, że państwowy przemysł informatyczny istnieje obecnie jedynie ze względu na... zamówienia z ZSRR. To właśnie one powstrzymują decydentów przed

przebranzowaniem fabryk. Dostawy na rynek krajowy są więc z założenia marginesem.

Bogatsi w wiedzę o realiach dotyczących produkcji sprzętu informatycznego, bez większych emocji możemy zająć się oceną mikrooferty państwowych producentów.

Zaprezentowano w zasadzie trzy mikrokomputery: MERITUM I, MK 45 i ComPAN 8. Z pewnością wiele można by tym konstrukcjom zarzucić. Jeśli jednak potraktować je jako wprowadzenie, na pewno dobrze się stało, że powstały. Produkcja MK 45 praktycznie w całości została zakontraktowana przez NBP. O produkowane w niewielkiej liczbie egzemplarzy MERITUM rozpoczęły się istne boje, tak że prawdziwym cudem by było, gdyby — zgodnie z zapowiedziami — jakieś egzemplarze trafiły do szkół. ComPAN 8 to dopiero prototyp; rozpoczęcie produkcji zapowiadane jest na przyszły rok. Można więc podsumować: na rynku nie jest obecnie dostępny żaden mikrokomputer, przynajmniej dla nabywców „bez pleców”.

W tej sytuacji jakby z innego świata rozbrzmiewa zakończenie wydanego przez MERY-ELZAB prospektu reklamowego o MERITUM I: „Ask about more information, quick delivery, program and demonstration. You will find that when you need us we will be there”.

Nieco większym optymizmem napała oferta MERY-BŁONIE. Otóż — jak twierdził przedstawiciel firmy — jeszcze na Targach można było składać zamówienia na drukarki D-100 z tegorocznej produkcji. Jeżeli jednak wszyscy zainteresowani dowiedzą się, że jest ona sprzedawana, należy się spodziewać, że sytuacja wróci do normy i D-100 nie będzie można dostać — tak samo, jak nie można było liczyć na DZM180.

Tyle można by w zasadzie napisać po Targach o ofercie państwowych producentów w dziedzinie mikroinformatyki. Nie przytłacza ona ani zbyt szeroką inwencją (pięć lat po wprowadzeniu przez firmę INTEL mikroprocesora 8086, którego odpowiednik jest już produkowany w ZSRR, nikt nie przedstawił nawet prototypu mikrokomputera lub sterownika 16-bitowego) ani perspektywami na szybkie upowszechnienie informatyki.

Z tym większą uwagą należy przyrzeć się ofercie przedsiębiorstw „półniwnych” i prywatnych. Ich potencjał produkcyjny, nie mówiąc już o zapleczu badawczo-rozwojowym, jest o kilka rzędów wielkości mniejszy od państwowych gigantów. Tymczasem rozmiary produkcji są w wielu przypadkach bardzo zbliżone. A gdy zsumować ofertę prywatnych producentów — jest ona znacznie bogatsza!

Dla przykładu wymienimy tu AMEPROD z zupełnie nowym mikrokomputerem AC 805, do którego można dokupić nie tylko napędy dysków elastycznych 5,25”, ale nawet dysk Winchestera i to o pojemności 30 MB. Z kolei IMPOL II oferuje dwie kon-

strukcje: IMP 85 oraz IMZ 80. IMPOL I z oferowanego dotychczas modułowego sterownika złożył modułowy mikrokomputer, którego możliwości mogą wzrastać wraz z apetytem użytkownika. Trzeba jeszcze wymienić CS 80 z firmy COMPUTEX, a i tak nie będzie to pełny obraz oferty mikroinformatycznej. Już niebawem zaprezentujemy w INFORMATYCE opisy wielu z tych urządzeń.

Warto jeszcze wspomnieć o poczynaniach tych firm zupełnie nie pasujących do „komerckiego” obrazu wytrwale kreowanego przez mass-media. Firma AMEPROD nie tylko fundowała nagrody w konkursach na grę edukacyjną, ogłaszanych przez Klub ABAKUS, lecz rozpoczęła wydawanie pisemka dla użytkowników ZX81 (produkcję tego mikrokomputera zakończy w tym roku).

Nowe (na mikroinformatycznej mapie) przedsiębiorstwo POLBRIT zamierza otworzyć... Centrum Dydaktyczne, w którym mikroinformatyczny laicy mogliby „własnoręcznie” przekonać się, na czym polega fenomen mikrokomputera. Firma ta prowadzi obecnie rozmowy, które mogą się zakończyć produkcją 4000 (rocznie) egzemplarzy ZX SPECTRUM z przeznaczeniem dla szkół. Zobaczymy, co odpowie Ministerstwo Oświaty na złożoną propozycję.

Zupełnie innej natury działalność podjęła firma COMPUTER STUDIO KAJKOWSKI (CSK). Nie tylko dostarcza oprogramowanie użytkowe pozwalające na korzystanie z mikrokomputera przez nieinformatyków, ale — pragnąc sprzymierzyć swe siły z ośrodkami naukowymi — zamierza prowadzić seminaria na temat oprogramowania graficznego.

Prywatne firmy informatyczne zostały porozrzucane po terenie targowym, co znacznie utrudnia zestawienie ich oferty ze zblokowaną ekspozycją MERY. Mimo to trudno oprzeć się wrażeniu, że powstał tu swoisty „drugi obieg”. Nazwę tę nasuwa niestety polityka państwa, w wyniku której do jednego worka wrzucono firmy produkujące dla zbitcia fortuny breloczki oraz firmy ambitne podejmujące zagadnienia, z którymi nie bardzo sobie radzą nawet gigantyczne firmy państwowe.

Mimo zadziwiająco bogatej oferty firm prywatnych, perspektywy na przyszłość nie są optymistyczne. Prawdziwy popyt na informatykę dopiero się objawi. To, że produkty firm prywatnych były dostępne w realnych terminach, wynikało tylko z początkowego braku zaufania do urządzeń wykonywanych w pozornie chałupniczych warunkach. Porównując jednak 36-godzinną (maksymalnie) obsługę serwisową firmy IMPOL z wielomiesięcznymi przetargami z MERA-KFAP, trudno nie zmienić zdania. A pierwsze przykłady „odsysania” rynku już są. Na przykład, PKP zaproponowało AMEPRODOWI wykupienie całorocznej produkcji.

Ktoś mógłby więc zapytać, dlaczego firmy „półniwne” nie powiększą pro-

dukcji — tak, by sprostać rosnącemu popytowi. Przyczyn jest kilka. Po pierwsze — nie ma dostatecznej podaży urządzeń peryferyjnych (trzeba je sprowadzać z zagranicy co pokaźnie nadzarpuje skromne zasoby dewizowe firm prywatnych, pozostałe po opłaceniu żądanego przez państwo haraczu). Po drugie — zwiększenie produkcji, a więc i obrotów firmy, pociąga za sobą ostrą progresję podatkową, stawiającą całe przedsięwzięcie pod znakiem zapytania. Po trzecie — kłopoty lokalowe, po czwarte — ... lista jest jeszcze długa.

Podsumowując mikroinformatyczne impresje, należy stwierdzić, że potencjalnie możemy całkiem sporo. Istnieje jednak jakiś wyższej natury czynnik, hamujący rozwój zarówno sektora państwowego, jak i prywatnego. O ile w pierwszym przypadku w szaleństwie (bo blokowanie rozwoju mikroinformatyki trudno określić innym mianem) można dopatrzeć się pewnej metody, o tyle w sektorze prywatnym zasadniczym hamulcem jest „feudalna” polityka podatkowa.

Nie żądamy więc zapalenia zielonego światła, otwórzmy szlaban.

ANDRZEJ J. PIOTROWSKI

NOWE CENY OGŁOSZEŃ

Od stycznia 1985 r. obowiązują następujące ceny ogłoszeń publikowanych na naszych łamach:

● ogłoszenie duże (zależnie od objętości): cała strona — 35 tys. zł; 3/4 — 30 tys.; 1/2 — 25 tys.; 1/4 — 20 tys.; 1/8 — 15 tys.

● ogłoszenia drobne (zależnie od liczby słów): jedno słowo — 30 zł

Dodatki do ceny podstawowej:

— za dodatkowy kolor (na okładce) +30%
— za zamieszczenie ogłoszenia na czwartej stronie okładki +100%
— za zamieszczenie ogłoszenia na trzeciej stronie okładki +50%

Zniżki:

— za ogłoszenie 3–5-krotne —5%
— za ogłoszenie 6–10-krotne —10%
— za ogłoszenie 14-krotne i powyżej —20%
— za artykuły reklamowe i wkładki wykonane przez zleceniodawcę —40%
— za bloki i biuletyny wykonane przez zleceniodawcę —maks. 60%.

Oferujemy bloki CAMACowskie:

pamięć półprzewodnikowe 4K
programator EPROMów
W przygotowaniu:
pamięć dynamiczna 16K
interfejs do minidysków elastycznych
sterownik na bazie 8086
Warszawa, tel.: 28-11-50

Dokumentacja oprogramowania (3)

Nawiązując do omawianych w bieżącym roku terminów dotyczących dokumentacji oprogramowania (INFORMATYKA, nr 3, 5 i 8, 1984), warto zwrócić uwagę na istnienie krajowych norm w tym zakresie. Co prawda, są to tylko normy branżowe, a więc nie obowiązują we wszystkich środowiskach i ich zasięg nie musi być ogólnopolski, jednakże warto porównać zawarte w nich określenia dokumentów z omówionymi na łamach INFORMATYKI.

Spośród dwóch serii norm BN/3109-01 i BN/3102-01 ustanowionych, odpowiednio, w 1977 i 1981 roku, omówimy tylko arkusze tej ostatniej, jako ogólniejsze i nowsze, choć odnoszą się one w zasadzie tylko do dokumentacji programów Jednolitego Systemu i Systemu Minikomputerów (JS i SM) w obrocie międzynarodowym.

W BN/3102-01 wyróżniono trzy podstawowe rodzaje dokumentów dotyczących programów. Są to dokumenty zawierające informacje konieczne do opracowania, rozpowszechniania i eksploatacji programu.

Jedynym dokumentem pierwszej grupy są założenia techniczne, zdefiniowane jako dokument określający: przeznaczenie i zakres stosowania programu, wymagania na program i jego dokumentację, a także dane dotyczące procesu opracowania. W dalszej części tej normy stwierdzono, że w skład tego dokumentu powinny wchodzić również wymagania dotyczące kontroli i odbioru pracy na różnych jej etapach, a także — dotyczące rodzajów badań.

W porównaniu z analogicznymi dokumentami, powstającymi w procesie wytwarzania oprogramowania według omówionego raportu EPRI EL-3089 (INFORMATYKA, nr 5, 1984), wyżej zdefiniowany dokument jest znacznie uboższy. Nie wyróżnia się w nim jawnie żadnej z faz specyfikacji (wymagań, projektu wstępnego i szczegółowego), omawiając tylko ogólnie etapy opracowania. Nie mówi się też dostatecznie wyraźnie o testowaniu programu, nazywając je trybem kontroli i odbioru (stwierdzenie to pominięto zresztą w samej definicji założeń technicznych). Prawdopodobnie przez nieuwagę nie sklasyfikowano też w tej samej grupie ważnego dokumentu nazwanego *metodyką badań*, który zawiera wymagania w zakresie kontroli zgodności programu i jego dokumentacji z założeniami technicznymi danego programu oraz określa tryb i metody kontroli.

Główną grupę dokumentów związanych z programem stanowią dokumenty eksploatacyjne. Opis zastosowania jest dokumentem zawierającym informację o przeznaczeniu programu, zakresie stosowania, klasie rozwiązywanych problemów, stosowanych metodach, ograniczeniach przy realizacji oraz o konfiguracji środków technicznych. Podręcznik programisty systemowego jest dokumentem zawierającym informację wystarczającą do:

- sprawdzenia programu
- zapewnienia jego właściwej eksploatacji
- instalowania programu w danym zestawie komputerowym.

Podręcznik programisty jest dokumentem zawierającym informacje niezbędne do pełnego wykorzystania możliwości programu. Podręcznik operatora jest dokumentem zawierającym informacje niezbędne do wykonania czynności związanych z przetworzeniem programu¹⁾. Opis języka jest dokumentem zawierającym opis składni i semantyki języka.

Podobnie jak w przypadku poprzedniej grupy dokumentów, najtrudniejsze funkcje związane z eksploatacją, tzn. instalowanie, modyfikowanie i testowanie programu, od-

¹⁾ Wydaje mi się, że chodzi tu o wykonanie programu

Zelazny R.: Narzędzia inżynierii oprogramowania (1). Model cyklu produkcyjnego i jego narzędzia

INFORMATYKA 1984, Nr. 10, s. 1

Pierwsza część przeglądu współczesnych narzędzi inżynierii oprogramowania. Scharakteryzowano główne etapy procesu produkcji oprogramowania oraz stosowane w nim narzędzia, a także próby integracji tych narzędzi, ilustrowane przykładami praktycznej realizacji.

Szyller J.: Mikroprocesory lat osiemdziesiątych

INFORMATYKA 1984, nr 10, s. 5

Geneza ewolucji i charakterystyka podstawowych cech architektury współczesnych mikroprocesorów. Omówiono kierunki przyszłego rozwoju tej architektury.

Udrycki W., Wilczyński W.: CHILL — język programowania systemów komutacyjnych (2). Budowa programu i wykonanie współbieżne

INFORMATYKA 1984, nr 10, s. 9

Druga część charakterystyki języka CHILL. Omówiono strukturę programu, funkcje procedury i metody identyfikacji procesów oraz sposoby współbieżnego wykonywania zadań.

De Mezer J.: Sterowanie alfaskopem z zastosowaniem układu SY 6545

INFORMATYKA 1984, nr 10, s. 21

Charakterystyka rozwiązania konstrukcyjnego oraz sposobu działania kolorowego alfaskopu rastrowego z zastosowaniem scalonego układu sterującego SY 6545. Omówiono podstawowe zalety przedstawionego rozwiązania.

Желязны Р.: Инструменты технологии производства программного обеспечения (1). Модель производственного цикла и его инструменты

INFORMATYKA 1984, № 10, стр. 1

Первая часть обзора современных инструментов технологии производства программного обеспечения. Охарактеризованы главные этапы процесса производства программного обеспечения и применяемые в этом процессе инструменты, а также попытки интеграции этих инструментов, иллюстрированные примерами практической реализации.

Шиллер Й.: Микропроцессоры восьмидесятых годов

INFORMATYKA 1984, № 10, стр. 5

Эволюция и характеристика основных свойств архитектуры современных микропроцессоров. Обсуждены направления будущего развития этой архитектуры.

Удрыцки В., Вильчиньски В.: CHILL — язык программирования коммутационных систем (2). Структура программы и параллельное выполнение.

INFORMATYKA 1984, № 10, стр. 9

Вторая часть характеристики языка CHILL. Обсуждены структура программы, функции процедуры и методы идентификации процессов, а также способы параллельного выполнения задач.

Де МЕЗЕР Й.: Управление буквенно-цифровым дисплеем с применением схемы SY 6545

INFORMATYKA 1984, № 10, стр. 21

Характеристика конструкционного решения и способа действия цветного буквенно-цифрового растрового дисплея с применением интегральной схемы SY 6545. Обсуждены основные преимущества представленного решения.

Zelazny R.: Tools for software engineering (1). Production cycle model and its tools

INFORMATYKA 1984, No. 10, p. 1

First part of the survey of contemporary tools for software engineering. Main phases of software production process and applied tools, as well as tools integration attempts, illustrated on practical realization examples, are characterized.

Szyller J.: Microprocessors of the eighties

INFORMATYKA 1984, No. 10, p. 5

Genesis of evolution and characteristics of basic features of modern microprocessor architecture. Future development trends of this architecture are discussed.

Udrycki W., Wilczyński W.: CHILL — a programming language for commutation systems (2). Program structure and concurrent realization

INFORMATYKA 1984, No. 10, p. 9

Second part of the CHILL-language characteristics. Program structure, procedure functions and process identification methods, as well as concurrent task realization are discussed.

De Mezer J.: Alphadisplay control using SY 6545 circuit

INFORMATYKA 1984, No. 10, p. 21

Characteristics of construction and operation of colour alpha-rasterdisplay using SY 6545 integrated control circuit. Basic advantages of this solution are discussed.

Zelazny R.: Hilfsmittel für Software-Technik (1). Produktionszyklusmodell und seine Hilfsmittel

INFORMATYKA 1984, nr 10, S. 1

Erster Teil einer Übersicht über die modernen Hilfsmittel für Software-Technik. Es wurden Hauptetappen der Softwareherstellungsprozess und die dort angewendeten Hilfsmittel charakterisiert, sowie Integrierungsversuche betreffend dieser Hilfsmittel, mit praktischen Realisierungsbeispielen illustriert, besprochen.

Szyller J.: Mikroprozessore der achtziger Jahre

INFORMATYKA 1984, Nr. 10, S. 5

Entwicklungsursprung und eine Charakteristik der Grundeigenschaften zeitgenössischer Mikroprozessorarchitektur. Es wurden Richtungen der zukünftigen Entwicklung dieser Architektur besprochen.

Udrycki W., Wilczyński W.: CHILL — eine Programmiersprache für Kommutationssysteme (2). Programmbau und gleichlaufende Ausführung

INFORMATYKA 1984, Nr. 10, S. 9

Zweiter Teil einer Charakteristik von CHILL-Programmiersprache. Es wurden Programmbau, Prozedurfunktionen und Prozessidentifizierungsmethoden, sowie Lösungen der gleichlaufenden Aufgabenausführung besprochen.

De Mezer J.: Alphasingerätsteuerung mit Verwendung von SY 6545 Schaltung

INFORMATYKA 1984, Nr. 10, S. 21

Eine Charakteristik der Konstruktionslösung und Wirkungsweise eines farbigen Rasteralphasingerätes mit Verwendung von SY 6545 integrierter Steuerschaltung. Es wurden wichtigste Vorteile der vorgestellten Lösung besprochen.

powiadające w przybliżeniu trzem dokumentom zalecanym w omówionym raporcie amerykańskim, opisuje się w jednym dokumencie, tj. podręczniku programisty systemowego. Natomiast istotne, choć znacznie bardziej oczywiste, informacje dotyczące użytkowania programu rozbiła się na dwa dokumenty — podręcznik programisty i podręcznik operatora. Dokument zwany w omawianej normie **opisem zastosowania** odpowiada w znacznym stopniu technicznemu opisowi teorii, zdefiniowanemu w raporcie EPRI.

W grupie dokumentów niezbędnych do rozpowszechniania programu wymieniono: opis programu i tekst programu w języku źródłowym. Opis programu jest dokumentem zawierającym informacje o strukturze logicznej i działaniu programu. Według cytowanego raportu EPRI, dokumentem najbardziej zbliżonym do tak zrozumiowanego opisu programu jest tzw. program reference manual, który nazwałem podręcznikiem programisty (INFORMATYKA, nr 8, 1984). Określenie tekstu programu źródłowego jest dość banalne; mianowicie nazywa się tak dokument zawierający zapis tekstu programu w języku źródłowym (w językach źródłowych) z niezbędnymi komentarzami. Szczepnie mówiąc, nie bardzo widzę sens wyróżniania tej grupy dokumentów, gdyż rozpowszechniane powinny być wszystkie dokumenty eksploatacyjne.

Dla ścisłości należy dodać, że w omawianej normie BN/3102-01 wymieniono jeszcze dwa dokumenty, mające jedynie charakter zestawień: wykaz dokumentów eksploatacyjnych i tzw. specyfikacja, obejmująca wykaz programów wchodzących w skład danego programu wraz z wykazem dokumentów dla tego programu. Należy podkreślić, że tak rozumiana specyfikacja nie ma nic wspólnego ze specyfikacją w znaczeniu użytym w numerze 5, 1984 (por. polemika z czytelnikiem w numerze 5, 1983).

W powyższym omówieniu pominięto wiele szczegółów dotyczących zawartości konkretnych dokumentów według normy BN/3102-01, lecz takie zadanie wychodziłoby już poza zakres tej rubryki.

JANUSZ ZALEWSKI

Zasady prumeraty

dokończenie z IV strony odkładki

PRENUMERATA ze zleceniem wysyłki za granicę jest dwukrotnie droższa.

PRENUMERATA ULGOWA. Uprawnieni są do niej: indywidualni członkowie Stowarzyszeń Naukowo-Technicznych, studenci oraz uczniowie szkół zawodowych. Warunkiem prumeraty ulgowej jest poświadczenie blankietu przekaz (na odwrocie środkowego odcinka) pieczęcią Koła SNT NOT, wyższej uczelni lub szkoły.

UWAGA!

- Dotychczasowi prumeratorzy otrzymają przesyłkę pocztową „Informator tytułowy i cennik czasopism technicznych” oraz blankiet wpłaty-zamówienia z Zakładu Kolportażu.
- Instytucje zainteresowane prumeratą naszego czasopisma mogą otrzymać cennik i blankiet-zamówienie w Zakładzie Kolportażu oraz w:
 - Biurach Wydawniczych:
 - 90-020 Łódź, Pl. Komuny Paryskiej 5a
 - 50-019 Wrocław, ul. Świerzewskiego 74
 - Oddziale Wydawnictwa:
 - 40-014 Katowice, ul. Dąbrowskiego 23
 - Oddziałach Wojewódzkich NOT
 - Zarządach Głównych Stowarzyszeń Naukowo-Technicznych
 - Redakcjach czasopism branżowych.
- Powyższe warunki prumeraty dotyczą czasopism kolportowanych przez Wydawnictwo NOT SIGMA.

Dodatkowych informacji o prumeracie udziela Zakład Kolportażu, tel. 40-00-21 wew. 293, 299 lub 40-35-89.

EGZEMPLARZE ARCHIWALNE czasopism wydawanych przez Wydawnictwo NOT SIGMA (od 1977 roku) można zamawiać w Zakładzie Kolportażu, 00-950 Warszawa ul. Mazowiecka 12 (tel. 26-80-16) lub nabywać w Klubie Prasy Technicznej, 00-950 Warszawa, ul. Mazowiecka 12 (tel. 27-43-65).

■ Cena oprogramowania dla komputerów osobistych spada, ponieważ możliwe jest przegrywanie taśm z danymi za pomocą sprzętu do nagrywania muzyki na kasety magnetofonowe. Pozwala to na nieprzerwaną produkcję podczas sezonowego zmniejszenia popytu na nagrania muzyczne. Jedno z takich przedsięwzięć — ABLEX — wyprodukowało wiosną 1983 roku tyle samo kaset z muzyką, co z programami komputerowymi. Większość programów jest dostosowana z racji braku uznanych standardów tylko do jednego typu komputera, muszą zatem być oddzielnie opracowywane, kopiowane i pakowane. Czynności te są tanie, ponieważ zapis programu odbywa się z częstotliwością fal słyszalnych, a zatem może być kopiowany tak jak muzyka. Podstawą produkcji kaset z muzyką jest nagranie na taśmie 25-milimetrowej, tworzącej zamkniętą pętlę. Taśma ta jest odtwarzana z bardzo dużą prędkością przez głowicę, która przesyła zapis do urządzeń kopiujących, na których z tą samą prędkością przesuwają się wielkie otwarte zwoje taśm do nagrania. W przypadku programów procedura jest identyczna, różnica polega na dokonywaniu nagrań mono na taśmach bardzo wysokiej jakości. Nagrane zwoje, zawierający średnio ponad 500 kopii programu, jest cięty na części, które są umieszczane w standardowych kasetach. Przegrywanie programów odbywa się zwykle tak, jak kopiowanie taśm hi-fi z muzyką, tzn. 32 razy szybciej, od normalnej prędkości odtwarzania. Jednakże firma ABLEX stwierdziła, że ZX SPECTRUM ma tak dużą tolerancję przy odczycie, że kopiowanie może odbywać się znacznie szybciej. Źródłowe taśmy do powielania programów dla tego komputera przesuwane są więc 64 razy szybciej od normalnej prędkości odtwarzania. Pozostaje problem kontroli jakości otrzymanego produktu. Jedyną drogą sprawdzenia poprawności zapisu programu jest jego uruchomienie na komputerze. Wykonanie tego z każdą wyprodukowaną kasetą istotnie zwiększyłoby jej koszt. ABLEX sprawdza więc tylko pierwszy i ostatni zapis ze zwoju. Jeżeli oba są prawidłowe, istnieje bardzo duże prawdopodobieństwo, że pozostałe również są bezbłędne. (I)

*

■ Ministrowie przemysłu krajów EWG uchwalili na sesji w Atenach dziesięcioletni program, którego zadaniem jest doścignięcie przez Wspólnotę Europejską USA i Japonii w rozwoju technologii komputerowej. Program ten o kryptonimie ESPRIT (European Strategic Program for Research in Information Technology) przewiduje łączne wydatki w wysokości 1,2 mld dol. na finansowanie prac badawczych w dziedzinie oprogramowania i mikroelektroniki.

Decyzja powyższa wynika z faktu, że wzajemne obroty w handlu sprzętem informatycznym między krajami EWG i USA, które jeszcze w latach siedemdziesiątych wykazywały saldo dodatnie, w 1981 r. zamknęły się deficytem w kwocie ok. 5 mld dol. Deficyt ten w 1982 r. uległ podwojeniu. W związku z tym jeden z czołowych przedstawicieli brytyjskiego ministerstwa przemysłu oświadczył, że nie podjęcie wspomnianych działań oznacza nieuchronną likwidację zachodnioeuropejskie-

go przemysłu komputerowego jeszcze przed końcem bieżącego stulecia. (K)

■ Rząd szwedzki rozpatrywał ostatnio projekt założeń pięcioletniego planu prac badawczo-rozwojowych oraz szkolenia w dziedzinie mikroelektroniki. Koszty realizacji tego planu określono na kwotę 714 mln koron (ok. 90 mln dol.). Mają być one pokryte częściowo z budżetu państwowego, a częściowo zostaną sfinansowane przez zainteresowane koncerny przemysłowe. Plan obejmuje badania podstawowe w dziedzinie półprzewodników oraz badania stosowane w obszarze elektroniki i optyki, z głównym zadaniem gromadzenia doświadczeń z zakresu stosowania nowych technologii wytwarzania. Część planu, dotycząca prac rozwojowych w przemyśle, ma na celu zwiększenie aktywności możliwie najszerzego kręgu producentów w zakresie wprowadzania nowych rozwiązań konstrukcyjnych. Program szkolenia przewiduje wyposażenie szkół wyższych w sprzęt mikroelektroniczny do celów dydaktycznych. Podobnej wielkości środki finansowe mają być przeznaczone na rozszerzenie wiedzy technicznej w małych i średnich przedsiębiorstwach oraz na organizowanie kursów dokształcających dla nauczycieli i konstruktorów. (K)

*

■ Po zakładach VIDEOTON, drugim najbardziej znanym producentem węgierskiego sprzętu informatycznego są zakłady MOM, mające ponad stuletnią tradycję w dziedzinie wyrobów optycznych. Do największych osiągnięć tego producenta należy uruchomienie w oparciu o licencję francuską produkcji niewymiennych pamięci dyskowych DM 0,8 o pojemności od 0,8 do 2,5 Mb. Jednostka o symbolu FPE-4000/4001, złożona z dwóch pamięci typu DM 0,8 jest eksportowana do NRD, gdzie wykorzystuje się ją w konfiguracji komputera ROBOTRON 4000. Od trzech lat zakłady produkują również pamięci na dyskach elastycznych, m.in. typu MF 3200 w kooperacji z kombinatem ROBOTRON, który jest jednym z głównych odbiorców wyrobów MOM. Niedawno uruchomiono produkcję nowej rodziny tych pamięci o symbolu MF 6400. Przy średnicy 8 cali mają one pojemność 6,4 Mb i metodę zapisu MFM. W br. dostarczany będzie nowy model tej pamięci o symbolu MF 6400D z zapisem obustronnym (podwójna głowica) i pojemnością 12,8 Mb. Zakłady przygotowują się również do uruchomienia produkcji minidyskietek 5 1/2-calowych, oznaczonych symbolami MF 900 oraz MF 1800 o pojemności 0,9 i 1,8 Mb. Oba te modele przeszły z wynikiem pozytywnym międzynarodowe badania Jednolitego Systemu, skutkiem czego otrzymały symbole EC 5088 i EC 5089. W konsekwencji ulepszeń konstrukcyjnych, od ub.r. produkowane są minidyskietki o symbolu MF 4001, których gabaryty zostały zmniejszone do polowy. Jeszcze w bieżącym roku wejdzie do produkcji ulepszony model tej pamięci o symbolu MF 8001 z zapisem dwustronnym i pojemnością 8 Mb. (K)

*

■ W ostrym współzawodnictwie, jakie zaobserwować można również w dziedzinie miniaturyzacji pamięci zewnętrznych, in-

teresującym rozwiązaniem są mikrodyskietki o średnicy 3 oraz 3 1/2 cala w osłonie ze sztywnego tworzywa. Zostały one wprowadzone na rynek przez firmy japońskie i dzięki wspomnianej osłonie bardzo przypominają cienkie kasety z taśmą do nagrywania dźwięku. Nowe rozwiązanie osłony ma na celu bardziej skuteczne zabezpieczenie dyskietki przed ewentualnym uszkodzeniem niż to zapewniają dotychczas stosowane w dyskietkach o średnicy 8 oraz 5 1/2 cala osłony papierowe. Dodatkowym ulepszeniem nowej osłony jest większa szczelność otworu, w którym następuje kontakt głowicy magnetycznej z powierzchnią dyskietki. Zapobiega to zanieczyszczeniu tej powierzchni przez kurz. Szybkość dyskietek 3-calowych wynosi 300 obrotów/min, natomiast 3 1/2-calowych — 600 obrotów/min. (K)

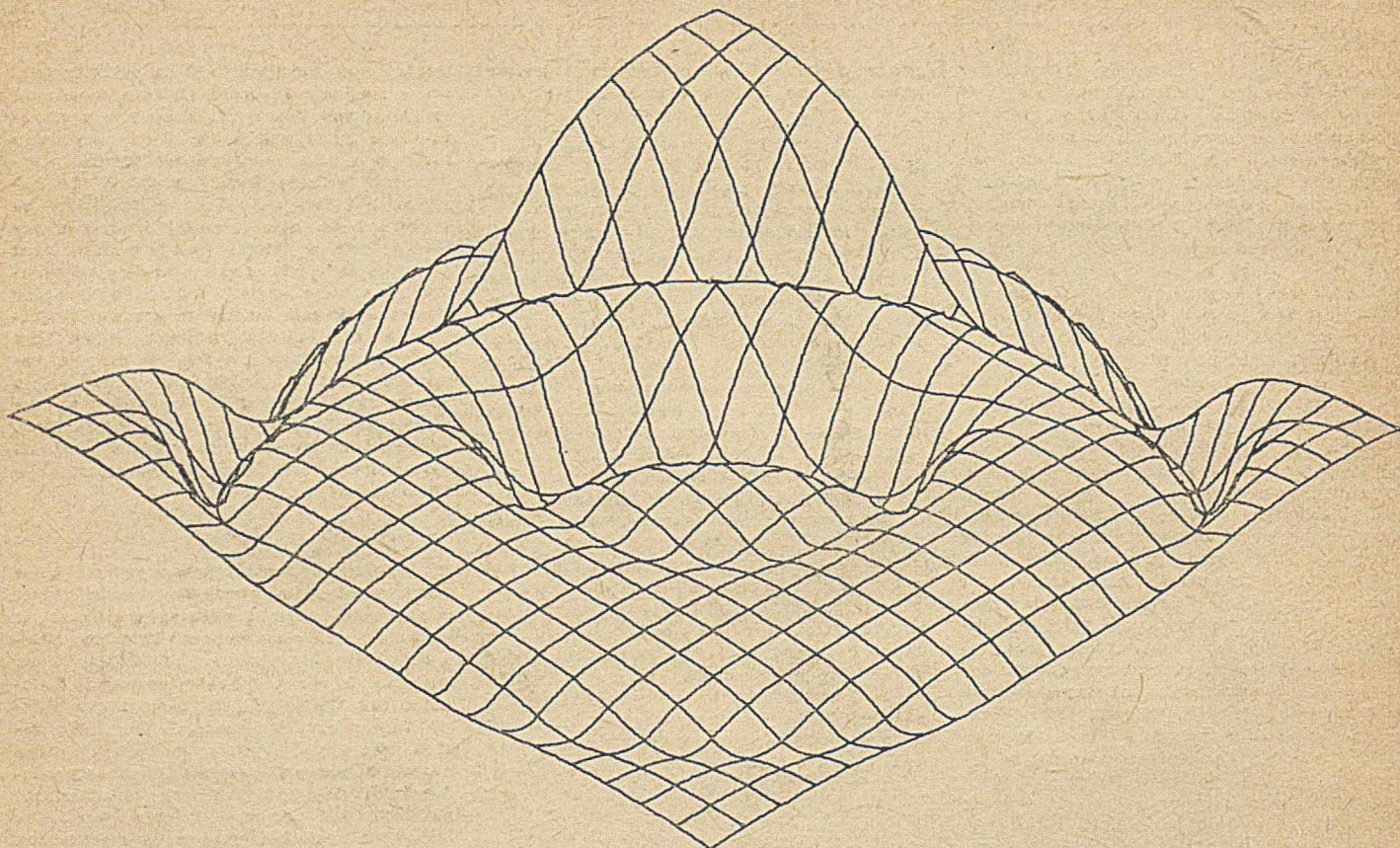
*

■ Według doniesień prasy chińskiej, w ChRL zbudowano pierwszy superkomputer o szybkości 100 mln operacji/s. Osiągnięcie to określono jako wejście Chin do światowej czołówki państw produkujących w konstrukcji sprzętu komputerowego. Komputer ten został uruchomiony i przekazany do próbnego eksploatacji po pięciu latach intensywnych prac badawczych. Jego nazwa o niewątpliwie niechińskim pochodzeniu (GALAXY) świadczy dobitnie, że podobnie jak osiągnięcia tego kraju w dziedzinie techniki jądrowej i satelitarnej, w pracach badawczo-rozwojowych uczestniczyć musieli specjaliści amerykańscy pochodzenia chińskiego, a co najmniej specjaliści wykształceni na uniwersytetach amerykańskich.

Dotychczasowa eksploatacja superkomputera pozwoliła rozwiązać istotne problemy poszukiwań chińskiej ropy naftowej, nierozwiązywalne za pomocą uprzednio posiadanej sprzętu (najszybsze chińskie komputery działały dotąd z szybkością ok. 2 mln operacji/s). Osiągnięcie w budowie superkomputera komentowane jest przez prasę światową jako jeden z najważniejszych elementów nowej polityki gospodarczej ChRL, mającej na celu przyspieszenie rozwoju i unowocześnienie przemysłu oraz potencjału militarnego. (K)

*

■ Kalifornijska firma ATARI postanowiła stworzyć dla studentów zachętę do korzystania z komputerów osobistych w laboratoriach fizycznych i chemicznych. W tym celu wprowadziła na rynek nowy produkt o nazwie ATARI LAB, wyposażony w osiem przegłów typu telefonicznego, umożliwiających dołączanie miernika temperatury. Miernik ten, w kształcie półra, pozwala dokonywać szybkich pomiarów temperatury gazów, cieczy i materiałów sypkich (np. gleby) w zakresie od -5 do +45°C. Odczyt temperatury może być dokonany 60 razy w ciągu jednej sekundy przez dowolnie długi czas, przy czym wyniki są prezentowane na ekranie komputera w postaci barwnego obrazu termometru, wykresu funkcji lub tablicy wartości. Firma zamierza dostarczyć również inne czujniki, np. do pomiarów intensywności strumienia świetlnego. ATARI LAB może współpracować z komputerami APPLE oraz COMMODORE. Cena podstawowego modułu tego produktu wynosiła w kwietniu br. ok. 50 dol. (S)



Oto przykład zapisu powierzchni trójwymiarowej na płaskim rysunku. Prosimy uważnie obejrzeć grafikę, a potem obrócić ją o 180°. Na czym polega zmiana wypukłości? Które części są wklęsłe? Które widzimy od góry, a które od dołu?

RAFAŁ PIETRAK
JAKUB TATARKIEWICZ

Zasady prenumeraty

Zamówienia i przedpłaty na prenumeratę **INFORMATYKI** przyjmuje Zakład Kolportażu Wydawnictwa **NOT SIGMA**. Adres pocztowy: Wydawnictwo **NOT SIGMA** — Zakład Kolportażu, 00-950 Warszawa, skr. poczt. 1004. Konto bankowe: III O/M NBP Warszawa nr 1036-7490-139-11.

PRENUMERATORZY ZBIOROWI — jednostki gospodarki uspołecznionej, instytucje i organizacje — składają zamówienia w formie wpłaty-przelewu na specjalnym blankiecie opracowanym przez Wydawnictwo. Część tego blankietu zawiera listę tytułów czasopism kolportowanych przez Wydawnictwo.

W przypadku większej liczby odbiorców:

- od 2 do 5 — należy dokonać wpłaty oddzielnie dla każdego odbiorcy
- od 6 odbiorców — dokonać jednej wpłaty, z powołaniem się na znak kancelaryjny pisma przesłanego do Zakładu Kolportażu, a zawierającego wykaz adresów poszczególnych odbiorców i zamawianych dla nich tytułów czasopism oraz numer konta bankowego, z którego dokonany został przelew. Na wpłacie-zamówieniu należy podać łączne liczby egzemplarzy poszczególnych tytułów czasopism.

WPŁATY-ZAMÓWIENIA przyjmowane są w terminach:

- do 15 listopada — na I kwartał, I półrocze i cały rok następny
- do 28 lutego — na II, III i IV kwartał

- do 31 maja — na II półrocze i IV kwartał
- do 31 sierpnia — na IV kwartał

PRENUMERATA STAŁA-WIELOLETNIA dotyczy tylko prenumeratorów zbiorowych. Zamawiający będzie otrzymywał z Wydawnictwa potwierdzenie kontynuacji prenumeraty wraz z wezwaniem do zapłaty. Zmiany w prenumeracie należy zgłaszać pisemnie w terminach obowiązujących dla składania zamówień-wpłat.

PRENUMERATORZY INDYWIDUALNI — osoby fizyczne — zamawiają prenumeratę dokonując wpłaty na blankiecie opracowanym przez Wydawnictwo **NOT SIGMA** (dostępnym w urzędach pocztowych) lub na blankiecie przekazu NBP — w terminach j.w. Na odwrocie środkowego odcinka przekazu, przeznaczonego dla adresata-posiadacza rachunku (Wydawnictwa), należy podać tytuł zamawianego czasopisma, okres prenumeraty i liczbę egzemplarzy.

Ceny **INFORMATYKI**

Cena egz.	Prenumerata w złotych					
	Normalna			Ulgowa		
	kwart.	półroc.	roczna	kwart.	półroc.	roczna
100,—						
300,—	600,—	1200,—	105,—	210,—	420,—	