

P. 1877/83

2 1983

informatyka

W NUMERZE :
TECHNIKA MIKROPROCESOROWA (2)

Czytelnicy – INFORMATYCE?

Mamy kłopoty. INFORMATYKA przestała być pismem dochodowym, więcej — zaczęła przynosić deficyt. Przyczyną tego jest znaczna podwyżka cen usług poligraficznych, ale także — nagła zmiana zasad zamawiania prenumeraty. Dotychczasowy kolporter — RSW „Prasa-Książka-Ruch” — nie widząc w swym wykazie INFORMATYKI niejednokrotnie informował prenumeratorów, że czasopismo przestało istnieć. Ponadto, ze względu na cięcia budżetowe, wiele instytucji zrezygnowało z prenumeraty bądź zmniejszyło liczbę zamawianych egzemplarzy. W rezultacie nastąpił gwałtowny spadek nakładu.

Wprawdzie pisma naukowo-techniczne w Polsce nie bankrutują, nawet gdy sama redakcja zrezygnowała z wysiłku i robi wszystko na odczepnego — a do takich nie chcemy się zaliczać, niemniej zła budżetowa passa zagraża naszej pracy. Pieniądze, które — jako czasopismo o malejącym nakładzie — mamy do rozdzielenia, nie zachęcają nikogo do rozwijania działalności publicystycznej i zwiększania atrakcyjności łamów, co bez wątpienia jest możliwe, zważywszy choćby rewolucyjne znaczenie informatycznej technologii.

Musimy więc zwrócić się do Czytelników z prośbą o wystąpienie do własnych

instytucji lub bibliotek, z których korzystają, z namową zamówienia prenumeraty INFORMATYKI bądź zwiększenia liczby zamawianych egzemplarzy, a do tych, których mimo wysokiej ceny stać na to — o zamówienie prenumeraty indywidualnej. Jeżeli bowiem nie zwiększy się liczba prenumeratorów, Wydawnictwo SIGMA nie zaryzykuje zwiększenia nakładu.

Ze swojej strony zrobimy możliwie wszystko, by zapewnić Czytelnikom pismo na wyższym niż dotąd poziomie. By dało ono w miarę pełny obraz nie tylko polskiej, ale i światowej informatyki.

Nie jesteśmy finansowo niezależni. INFORMATYKA umieszczona jest w strukturze Wydawnictwa, które zajmuje się nie jednym, ale niemal stoma tytułami prasy technicznej. Trudno się więc dziwić, że mimo znaczenia dziedziny wiedzy, którą się zajmujemy, nie uzyskaliśmy żadnych priorytetów. A na dodatek spada nakład...

Prosimy rozważyć, czy INFORMATYKA jest pismem pożytecznym. Jeśli tak — liczymy na pomoc.

Nowe zasady prenumeraty przedstawiamy na czwartej stronie okładki.

REDAKCJA

KOLEGIUM REDAKCYJNE

Redaktor naczelny: prof. dr hab. Leon ŁUKASZEWICZ

mgr inż. Zbigniew GLUZA, dr Janusz GWIAZDA, Władysław KLEPACZ (zastępca redaktora naczelnego), dr inż. Tomasz PAWLAK, mgr Andrzej SZAEŁAS, mgr Zbigniew ŚWIRSKI, dr inż. Janusz ZALEWSKI

Sekretarz redakcji: mgr Teresa JABLONSKA

RADA PROGRAMOWA

Prof. dr hab. Tadeusz PECHE (przewodniczący), mgr inż. Tomasz BANKOWSKI (sekretarz), mgr inż. Antoni BOSSOWSKI, mgr inż. Roman BURNO, prof. dr hab. Andrzej JANICKI, mgr inż. Jan KRAMARCZUK, prof. dr hab. inż. Juliusz KULIKOWSKI, prof. dr hab. Leon ŁUKASZEWICZ, prof. dr hab. Antoni MAZURKIEWICZ, gen. dr inż. Marian PASTERNAK, dr inż. Bronisław PIWOWAR, mgr Zbigniew SUBSTYK, prof. dr hab. Tadeusz WALCZAK

Materiałów nie zamówionych Redakcja nie zwraca.

MACZELNA ORGANIZACJA TECHNICZNA
WYDAWNICTWO
CZASOPISMA I KSIĄŻEK TECHNICZNYCH



SIGMA

ul. Świętokrzyska 14a
00-950 Warszawa
skrytka pocztowa 1004

Redakcja: 00-041 Warszawa, ul. Jasna 14/16, pok. 129 i 133, tel. 27-71-40, dyżury redakcji 10.00—12.00

Zakł. Graf. „Tamka”. Zam. 359. Obj. 5,0 ark. druk. Nakład 3700 egz. M-76.

Cena egzemplarza zł 75.—

INDEKS 36124

Prenumerata roczna zł 900.—

	Strona
W NUMERZE:	
TECHNIKA MIKROPROCESORÓWA — każdy sobie <i>Janusz Zalewski</i>	2
Rozwój systemów mikrokomputerowych na przykładzie MERY-60 (SM-1633) <i>Andrzej Grzywak, Zbigniew Suchorończak</i>	3
Wybrane mikrokomputerowe bloki sterujące systemu CAMAC i mikrokomputery o przeznaczeniu uniwersalnym <i>Sławomir Jagiello, Andrzej Koźmiński, Krzysztof Rzymkowski</i>	9
Modułowy system sterowników mikroprocesorowych MIKROSTER <i>Jan Pieńkos, Krystyna Bugajska, Marek Dorywalski, Paweł Dziubiński, Bożena Kosińska, Stanisław Moszyński, Ireneusz Myzik, Małgorzata Nowak-Niedźwie- dzka, Andrzej Piotrowski, Adam Pluta, Jerzy Podsiadły, Zdzisław Pojmański, Piotr Smólski</i>	13
Modułowy System Mikroprocesorowy — MSM <i>Andrzej Woźniak, Marek Pawłowski, Janusz Rzeszut, Andrzej Skorupski, Janusz Sosnowski, Jacek Stochlak</i>	17
Alfaskop ALFA 31M jako urządzenie wejścia-wyjścia systemu mikrokompute- rowego <i>Władysław Binkowski, Marek Furyk, Andrzej Przybylski</i>	22
Jednopakietowy inteligentny sterownik graficzny MSG-3C <i>Marek Tadeusz Jankowski, Jacek Wertel</i>	24
Dyskowy system operacyjny CP/M. Część 1 <i>Marianna Sobczyk (oprac.)</i>	25
ALGORYTMY	
Operacje na liczbach całkowitych wielokrotnej precyzji <i>Andrzej Szalas, Zbigniew Świrski</i>	30
GRY	
Polowanie <i>Z. Świrski</i>	31
Z KRAJU	
Mikroprocesorowe Szkoły Zimowe. Refleksje współorganizatora <i>Jerzy Dańda</i>	32
GIEŁDA INFORMACJI	32—33
ZE ŚWIATA	
Najnowsze tendencje w dziedzinie komputerowych systemów sterowania <i>Janusz Zalewski</i>	34
TERMINOLOGIA	
Słowniczek mikroprocesorowy (cz. 2) <i>Marek Tadeusz Jankowski</i>	37
POGLĄDY	
Małe w informatyce <i>Jerzy Kisielnicki</i>	III okł.

Technika mikroprocesorowa — każdy sobie

Oddajemy do rąk Czytelników drugi numer **INFORMATYKI** poświęcony technice mikroprocesorowej w Polsce, przedstawiając tym razem wybrane konstrukcje systemów o przeznaczeniu uniwersalnym. Choć dobór poszczególnych artykułów był mimo wszystko dość przypadkowy, tematycznie zgrupowano je tak, aby stanowiły aktualny obraz krajowych prac w tej dziedzinie.

Czy Czytelnik otrzyma na tyle miarodajne informacje, aby mógł sobie odpowiedzieć na drugie pytanie postawione w poprzednim numerze — z czego robić? (na pierwsze — czym robić? — odpowiedzią był właśnie nr 1/83). Myślę, że tak — w każdym razie, to było celem zgrupowania tych artykułów w jednym numerze.

Przyszły nabywca i użytkownik systemów mikrokomputerowych otrzymuje wprawdzie ograniczony, ale rzetelny przegląd stanu najbardziej zaawansowanych prac krajowych. Mikrokomputerowy system 16-bitowy MERA-60 jest już sprzedawany od dawna, producent systemu MSM reklamował się nawet w **INFORMATYCE**, a CEMI ma zacząć produkcję systemu **MIKROSTER**. Sądzę, że bardziej wytrwali mogliby liczyć również na sprzedaż mikroprocesorowych sterowników camacowskich. Całość wygląda więc optymistycznie, bo każdy z wymienionych systemów zawiera dość szeroki zestaw modułów (pakietów) i ważniejsze składniki oprogramowania podstawowego.

Praktyka — wydaje się — będzie jednak inna, nawet jeśli tylko ograniczyć się do szczegółów technicznych. Przypuszczam, choć mogę się mylić, że w przyszłych, tzw. profesjonalnych zastosowaniach techniki mikroprocesorowej najważniejsza będzie uniwersalność tego sprzętu, tj. zarówno możliwość rozszerzania konfiguracji podstawowej, przez zakup lub konstruowanie nowych modułów w konsekwencji zwiększenia wymagań czy rozszerzenia funkcji, jak i możliwość łączenia tego sprzętu w większe systemy, dołączania do istniejących już instalacji.

Oczywiście, twórcy opisanych konstrukcji dobrze o tym wiedzą. W jednym z artykułów napisano, że „potrzeba opracowania zunifikowanej bazy konstrukcyjnej do budowy sterowników urządzeń i linii technologicznych wynika z konieczności skrócenia cyklu projektowania i wykonania urządzeń, zwiększenia ich niezawodności, a także zapewnienia odpowiednich warunków konserwacji”. Choć każdy system rozpatrywany oddzielnie ma budowę modułową i jest w pełni rozszerzalny — są one wzajemnie niezgodne. Przedstawionych systemów nie ze sobą nie łączy ani pod względem mechanicznym, ani elektrycznym, ani logicznym.

Począwszy od wymiarów płytek, przez standard magistrali, aż do organizacji logicznej — żaden z tych systemów nie odpowiada drugiemu. Wprawdzie systemy **MIKROSTER** i **MSM** mają ten sam wymiar płytki (nawiasem mówiąc — nie wiem z czego to wynika), niemniej, trudno przecież przypuszczać, aby w praktyce utrzymał się inny standard mechaniczny niż tzw. podwójna Eurokarta (160 × 233,4 mm). Jeżeli chodzi o magistralę, to poza dyskusją jest **MERA-60** — z założenia zgodna ze standardem firmy **DEC**. Wybór magistrali pozostałych systemów był

arbitralny i choć zawsze coś za nim przemawiało, a stosunkowo łatwo można go uzasadnić — jestem przekonany, że ta niezgodność utrzymywana dłużej przyniesie nam spore straty.

Co pocznie użytkownik, który chce w miarę sprawnie i szybko złożyć zestaw do określonego zastosowania (może nawet bez użycia lutownicy), gdy żaden pojedynczy producent nie zapewnia mu wszystkich modułów funkcjonalnych, za niektóre z nich żąda zbyt wygórowanych cen, czy wreszcie — nie dostarcza ich w wymaganym terminie.

Można kupować u różnych producentów (nawet za granicą), ale przy wzajemnej niezgodności różnych systemów jest to w zasadzie niemożliwe. Nie na darmo tak prężna organizacja jak **IEEE** (Institute of Electrical and Electronics Engineers, USA) obok trzech istniejących komitetów zajmujących się normalizacją magistrali **P696** (S-100), **P-796** (**MULTIBUS**) i **P-896** (**Future-Bus**) powołała czwarty — **P-961** do prac nad normalizacją magistrali **STD-Bus** (**STD — Simple To Design**).

Organizacja logiczna każdego z systemów prezentowanych w tym numerze, jest z innej planety. Można zrozumieć całkowitą odrębność systemu **MERA-60**, bo przecież **ELEKTRONIKA-60** jest mikroprocesorem 16-bitowym, ale wszystkie pozostałe zestawy są oparte na mikroprocesorze **INTEL 8080**. Możliwość łączenia opisywanego sprzętu w większe systemy jest również ograniczona. Zestawy są co prawda wyposażone w moduły służące do komunikacji przez standardowy sprzęg **V24** — jest to jeden z warunków podstawowych aby konstrukcja utrzymała się przy życiu, ale może to wystarczyć do komunikacji z jednym komputerem. Wątpię jednak aby wystarczyło do tworzenia bardziej złożonych zestawów, choćby wieloprocessorowych czy sieci mikrokomputerowych — a perspektywa ta nie jest wcale tak odległa. Obecnie jedynie system **CAMAC** stwarza możliwości standardowej realizacji takich zestawów ale jak zwracają uwagę autorzy, ma on znaczne ograniczenie zarówno od strony magistrali (zbyt skomplikowany), jak i ze względu na wymagania struktur sieciowych (jest niedostosowany). Warto w każdym razie zwrócić uwagę że zagadnienia tzw. systemów o rozłożonej inteligencji (ang. distributed intelligence systems), a mówiąc prościej — sieci mikrokomputerowych będą stanowiły niedługo podstawowe zadanie konstrukcyjne. Niestety, w tym numerze nie ma o nich mowy.

Reprezentowane jest natomiast inne ważne zagadnienie konstrukcyjne, tj. opracowanie systemów graficznych. Choć z konieczności często stosuje się tu rozwiązania doraźne, są już jaskółki bardziej systematycznego podejścia do tego problemu — niezwykle ważkiego w masie różnych zastosowań.

Generalnie biorąc — choć powstaje wiele ciekawych konstrukcji, trudno oprzeć się wrażeniu, że naczelną zasadą konstruktorów jest „każdy sobie”. Taką byłaby ocena zawartości bieżącego numeru. Niewątpliwie patrząc na to zagadnienie z innej strony, można z przekonaniem powiedzieć, że są to bardzo dobre konstrukcje. Myślę jednak, że wielu pochwał nie warto wygłaszać, robią to lepiej inne redakcje. A i tak dobry system obroni się sam.

JANUSZ ZALEWSKI

Rozwój systemów mikrokomputerowych na przykładzie MERY-60 (SM-1633)

Centrum Naukowo-Produkcyjne Systemów Sterowania MERA-STER w Katowicach rozpoczęło w 1979 r. produkcję systemu mikrokomputerowego MERA-60, przeznaczonego do sterowania procesami technologicznymi, obliczeń inżynierskich oraz badań naukowych [1, 3]. System opracowano we współpracy ze Związkiem Radzieckim, w oparciu o elementy o dużym stopniu scalenia produkcji ZSRR i — elementy krajowe. Jest on wyposażony w urządzenia peryferyjne produkcji polskiej, jak również krajowe urządzenia sprzężenia z obiektem. System MERA-60 ulega ciągłej rozbudowie i modernizacji, zarówno pod względem sprzętowym, jak i programowym.

W ubiegłym roku mikrokomputer MERA-60 został poddany badaniom międzynarodowym. Przeszedł je z wynikiem pozytywnym jako pierwsza polska maszyna cyfrowa w rodzinie SM (nadano mu kryptonim SM-1633).

Zakłada się stały rozwój systemu MERA-60, przede wszystkim — jego cech funkcjonalnych. Jednocześnie podejmuje się prace nad przekształceniem systemu mikrokomputerowego MERA-60 w system następnej generacji — MERA-60/256, którego wprowadzenie do produkcji seryjnej przewiduje się na lata 1986—1987.

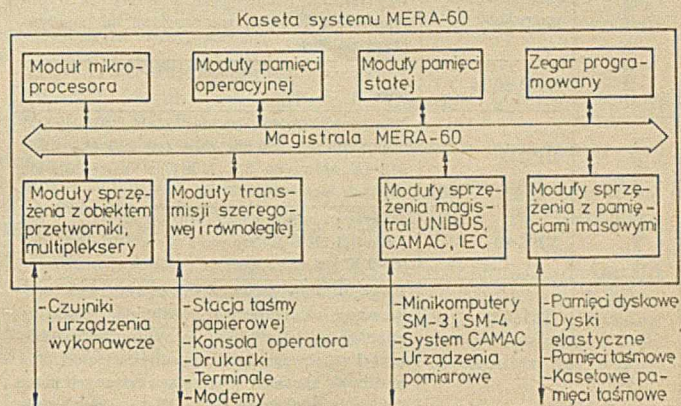
W artykule przedstawiono opis systemu MERA-60, wyniki badań międzynarodowych oraz podstawy koncepcji przekształcenia tego systemu w rodzinę MERA-60/256.

SYSTEM MIKROKOMPUTEROWY MERA-60

MERA-60 jest modułarnym systemem mikrokomputerowym opartym na współczesnych elementach półprzewodnikowych MSI oraz LSI (mikroprocesor serii K-590 produkcji radzieckiej). Podstawowym składnikiem systemu jest magistrala złożona z 16 linii danych i adresów oraz 23 linii sterujących. Do magistrali jest dołączony procesor, moduły pamięci, wybrane moduły standardowe oraz moduły specjalne; zaprojektowane zgodnie z wymaganiami użytkowymi. Architektura systemu przedstawiono na rysunku 1, a pełny wykaz modułów w tabeli 1.



Prof. dr hab. inż. ANDRZEJ GRZYWAK ukończył w 1954 r. Wydział Elektryczny Politechniki Śląskiej w Gliwicach, specjalność automatyka. Doktorat i habilitację uzyskał na tej samej uczelni. Od 1975 r. profesor nadzwyczajny nauk technicznych. Obecnie zatrudniony jako zastępca dyrektora ds. naukowo-badawczych w Instytucie Systemów Sterowania w Katowicach i profesor w Instytucie Informatyki Czasu Rzeczywistego Politechniki Śląskiej. Specjalizuje się w zagadnieniach architektury i budowy maszyn cyfrowych. Laureat nagrody państwowej I stopnia.



Rys. 1. Architektura systemu MERA-60

Obsługa urządzeń w systemie może odbywać się dwoma sposobami:

- programowo, na zasadzie badania rejestru stanu wybranego modułu
- sprzętowo, przez badanie stanu logicznego określonych linii sterujących magistrali, po wykonaniu kolejnego rozkazu.

Przy obsłudze sprzętowej stosuje się tzw. wektorowy system przerwania, umożliwiający praktycznie natychmiastowe zidentyfikowanie źródła przerwania (ze zwłoką 2—35 μ s) i wykonanie programu obsługi. Moduł elektronicznie najbliższy procesora ma najwyższy priorytet obsługi, a moduł najdalszy — priorytet najniższy. Oprócz przerwania sprzętowych mogą występować tzw. przerwania programowe realizowane za pomocą specjalnych rozkazów.

Wszystkie moduły dołączone do magistrali mają adresy z zakresu 16000₈—177776₈, pamięć operacyjna jest adresowana w zakresie 0₈—157776₈. Adresy 0₈—376₈ zarezerwowano dla tzw. wektorów przerwania.

Stos systemowy, obsługiwany sprzętowo, umożliwia automatyczne zapisywanie lub odczytywanie adresów przy skokach do podprogramów i powrotach z podprogramów.



Mgr inż. ZBIGNIEW SUCHORONCZAK ukończył w 1979 r. Wydział Automatyki i Informatyki Politechniki Śląskiej w Gliwicach. Pracuje w Instytucie Systemów Sterowania w Katowicach, w Zakładzie Konstrukcji Mikroprocesorowych. Zajmuje się architekturą systemów mikrokomputerowych. Jako kierownik tematu prowadził ze strony polskiej badania międzynarodowe MERY-60. Fierze udział w pracach nad konstrukcją MERY-60/256.

Tabela 1. Wykaz modułów elektronicznych systemu MERA-60

Nazwa	Typ	Opis
1	2	3
Processory	M1	Procesor 16-bitowy zrealizowany na pojedynczej płycie z blokiem 4 K słów pamięci półprzewodnikowej (odpowiednik procesora LSI-11 firmy DEC)
	M2	Procesor M1 z listą rozkazów rozszerzoną o rozkazy arytmetyczne i zmiennooprzecinkowe
Moduły pamięci operacyjnej	P1	Dynamiczna pamięć półprzewodnikowa o pojemności 4 K słów, zrealizowana na pojedynczym pakiecie, współpracująca z procesorem M1 (wymaga dodatkowego sygnału sterującego)
	P2	Pamięć o cechach pamięci P1 nie wymagająca żadnego sygnału dodatkowego
	MPD-60/8	Dynamiczna pamięć półprzewodnikowa o pojemności 8 K słów, zrealizowana na pojedynczym pakiecie
	MPD-60/16	Pamięć o cechach pamięci MPD-60/8 i o pojemności 16 K słów
Moduły pamięci stałych	MPR-60	Półprzewodnikowa pamięć stała typu ROM o pojemności 256-2K słów (co 256 słów) lub 512-4K słów (co 512 słów), w zależności od użytych elementów
	MPE-60	Półprzewodnikowa pamięć typu EPROM o pojemności 1K-8 K słów (co 1 K) lub 2K-16K słów (co 2 K) w zależności od użytych elementów
Zegar		Programowany zegar czasu rzeczywistego zrealizowany na podwójnym pakiecie, zliczający impulsy jednej z pięciu częstotliwości generatora kwarcowego lub częstotliwości sieci. Ma możliwość zliczania zdarzeń zewnętrznych przez 2 wejścia analogowe (± 12 V) i — sterowania przez 3 wyjścia (standard TTL)
Moduły sprzężenia z obiektem	MWW-60	Moduł wejść-wyjść cyfrowych dla 16 niebuforowanych linii danych z dwoma sygnałami sterującymi i dwoma sygnałami zewnętrznymi, mogącymi wywołać przerwanie (poziom sygnał TTL). Może być wykorzystany jako moduł transmisji równoległej
	MUX-60	Moduł multiplexera 64 wejść dwustanowych zorganizowanych bitowo (64×1) lub bajtowo (8×8 bitów, poziom sygnałów TTL)
	MDX-60	Moduł wejść-wyjść cyfrowych — niebuforowanych (2×16 wejść) i buforowanych linii danych (2×16 wyjść) z 16 sygnałami sterującymi i 4 sygnałami zewnętrznymi mogącymi wywołać przerwanie (poziom sygnał TTL)
Terminatory	K500 K502 K504	Moduły terminatorów magistrali MERA-60; 00 — kabel, 20 — dopasowanie 120 Ω , 04 — dopasowanie 250 Ω
	MTB-60	Moduł terminatora 250 Ω z wbudowanym programem samoladującym (ang. bootstrap)
Moduły transmisji	MDK-60/1	Moduł podstawowy transmisji szeregowej (według standardu V-24), z możliwością wyboru szybkości transmisji i liczby bitów stopu
	MDL-60	Moduł transmisji szeregowej (według standardu V-24), z możliwością przełączenia na transmisję w pętli prądowej. Ma możliwość wyboru szybkości transmisji i liczby bitów stopu
	MMT-60	Moduł czterokanałowy transmisji szeregowej (według standardu V-24), z rozszerzoną możliwością wyboru szybkości transmisji i liczby bitów stopu i sposobu kontroli transmisji dla obu kierunków (zrealizowany na podwójnym pakiecie)
Moduły transmisji	MTT-60	Moduł transmisji szeregowej (według standardu V-24) z rozszerzoną możliwością wyboru szybkości transmisji, liczby bitów stopu i długości słowa. Zapewnia kontrolę danych odbieranych i możliwość wyboru kontroli transmisji niezależnie dla obu kierunków. Zmiana szybkości transmisji może być dokonana programowo. Zrealizowany jest na podwójnym pakiecie
	MLP-60	Moduł transmisji równoległej słowa 8 bitowego, do komunikacji z urządzeniami znakowymi pracującymi w trybie start-stopowym; wyposażony w dodatkowe sygnały do współpracy z drukarką DZM-180

	1	2	3
Moduły transmisji		MCD-60	Moduł transmisji równoległej słowa 8 bitowego do komunikacji z urządzeniami znakowymi pracującymi w trybie start-stopowym
		MTS-60	Moduł transmisji synchronicznej zgodnej z protokołem BSC (zrealizowany na podwójnym pakiecie)
Urządzenia sprzężenia magistrali		MCM-60	Moduł sprzężenia magistrali MERA-60 ze sterownikiem kasyety CAMAC, typu 106 (zrealizowany na podwójnym pakiecie)
		MQU-60	Moduł sprzężenia magistrali MERA-60 z magistralą UNIBUS (zrealizowany na podwójnym pakiecie)
Moduły sprzężenia z pamięciami masowymi		MIE-60	Moduł sprzężenia magistrali MERA-60 z magistralą IEC (zrealizowany na podwójnym pakiecie)
		MDE-60	Moduł sprzężenia z pamięcią na dyskach elastycznych, typu SP-60 M
		MPK-60	Moduł sprzężenia z kasetową pamięcią taśmową typu PK-1
		MDT-60	Sterownik 4 pamięci dyskowych typu SM-5400 (zrealizowany na podwójnym pakiecie)
		MPT-60	Moduł sprzężenia z pamięcią taśmową typu PT-305 (zrealizowany na podwójnym pakiecie)

Podczas przerwania następuje automatyczne zapisywanie na stosie aktualnej zawartości licznika rozkazów i rejestru stanu procesora, a następnie — wpisanie wektora przerwania do licznika rozkazów i rejestru stanu.

Lista rozkazów mikrokomputera MERA-60 jest zgodna z listą rozkazów mikrokomputera PDP-11/03 firmy DEC [2, 5] i obejmuje rozkazy: bezadresowe, jednoadresowe i dwuadresowe. Kod rozkazów bezadresowych zawiera tylko kod operacji. W kodach rozkazów jednoadresowych i dwuadresowych zwykle zawarta jest informacja określająca wykonywaną operację, rejestry uniwersalne wykorzystywane przy pobieraniu argumentów, tryb adresowania.

Procesor systemu MERA-60 ma 8 rejestrów uniwersalnych (RO-R7). Są one wykorzystywane jako akumulatory, rejestry adresowe, rejestry adresowe o zawartości zmieniającej się automatycznie oraz rejestry indeksowe, których zawartość jest dodawana do tzw. przemieszczenia w celu otrzymania adresu argumentu (umożliwia to łatwe operowanie elementami tablicy).

Automatyczne zwiększanie zawartości rejestru adresowego podczas manipulowania zawartością kolejnych komórek pamięci nazywa się autoinkrementacją, natomiast automatyczne zmniejszanie zawartości rejestru adresowego — nazywa się autodekrementacją. Wykorzystanie autoinkrementacji i autodekrementacji umożliwia organizowanie stosu. Jako wskaźnik stosu może służyć dowolny rejestr uniwersalny. Niektóre rozkazy (np. używane przy obsłudze przerwań, skokach do podprogramów i powrocie) wykorzystują automatycznie rejestr R6 jako sprzętowy wskaźnik stosu.

Rejestr R7 pełni funkcję licznika rozkazów. Z tego powodu jego użycie w charakterze rejestru uniwersalnego ogranicza się do adresowania z autoinkrementacją, adresowania pośredniego z autoinkrementacją, względnego oraz względnego pośredniego. Ponadto można stosować adresowanie proste (ang. *immediate*) i bezpośrednie. We wszystkich trybach adresowania można używać dowolnego z ośmiu rejestrów uniwersalnych.

Znaczną zaletą systemu MERA-60 jest bogate oprogramowanie podstawowe, diagnostyczne, komunikacyjne i użytkowe (tab. 2). Oprogramowanie systemu MERA-60 jest zgodne z oprogramowaniem minikomputerów SM-3, SM-4 i PDP-11 [6]. W tabeli 2 podano także zamierzenia dotyczące rozszerzenia oprogramowania dla systemu MERA-60/256. Powstanie ono przede wszystkim w oparciu o wieloprogramowy i wielozadaniowy system operacyjny RSX-60M oraz języki programowania COBOL i ADA.

Tabela 2. Oprogramowanie mikrokomputerów MERA-60 i MERA-60/256

Nazwa i nośnik	Charakterystyka
System operacyjny RT-60 (dysk elastyczny lub twardy)	System operacyjny czasu rzeczywistego, przeznaczony w zasadzie dla jednego użytkownika. Umożliwia wykonywanie zadania czasu rzeczywistego, przy jednoczesnej pracy w trybie interakcyjnym lub wsadowym. Zapewnia korzystanie z następujących języków programowania: MACRO-ASSEMBLER, FORTRAN IV, FOCAL, APL, BASIC
System operacyjny RSX-60S (dysk elastyczny lub twardy)	Wieloprogramowy i wielozadaniowy system operacyjny czasu rzeczywistego, pełniący funkcje wykonawcze. Wykonuje programy przygotowane w językach MACRO i FORTRAN IV, natomiast nie zapewnia możliwości przygotowywania i uruchamiania programów
System operacyjny RSX-60M ¹⁾ (dysk elastyczny lub twardy)	Wieloprogramowy i wielozadaniowy system operacyjny czasu rzeczywistego stanowiący rozszerzenie systemu RSX-60S. Może być generowany dla różnych zastosowań, w postaci wielu odmian, od bardzo małych i prostych do dużych i rozbudowanych. Umożliwia programowanie w następujących językach: MACRO-ASSEMBLER, FORTRAN IV-PLUS, BASIC-PLUS-2, BASIC, COBOL, FOR, ADA
Oprogramowanie diagnostyczne (dysk elastyczny lub twardy, taśma papierowa)	Zbiór programów testowych dla wszystkich modułów dołączonych do magistrali (przeznaczony głównie do konserwacji i uruchamiania) oraz zestaw zadań kontrolnych umożliwiających testowanie pod kontrolą systemu operacyjnego
System komunikacyjny — wersja RT (dysk elastyczny lub twardy)	Wraz z systemem RT-60 stanowi oprogramowanie węzła sieci komputerowej zapewniające komunikację z pozostałymi węzłami sieci, tzn.: komunikację między zadaniami dwóch węzłów, transmisję zbiorów między węzłami, dostęp do zasobów drugiego węzła. Dostęp do systemu komunikacyjnego jest możliwy dla programów użytkowych napisanych w językach MACRO i FORTRAN
System komunikacyjny — wersja RSX ¹⁾ (dysk elastyczny lub twardy)	Wymienione funkcje komunikacyjne dla systemu RSX-60M
Inne oprogramowanie jak: edytor tekstowy, biblioteki programów itp.	

¹⁾ przewidziany jedynie dla mikrokomputera MERA-60/256

BADANIA MIĘDZYNARODOWE SYSTEMU MERA-60

Na przełomie maja i czerwca ub.r. w Instytucie Systemów Sterowania w Katowicach odbyły się 21-sze Międzynarodowe Badania Środków Technicznych systemu SM elektronicznych maszyn cyfrowych. Wzięli w nich udział specjaliści z ZSRR, CSRS, NRD i Bułgarii. Celem badań, do których przedstawiono mikrokomputer MERA-60, było włączenie go do środków technicznych jednolitego systemu SM. W myśl klasyfikacji środków technicznych systemu SM mikrokomputer MERA-60 należy do drugiej generacji maszyn cyfrowych w klasie SM 50/50-3. Główne cechy charakterystyczne tej klasy to oparcie procesora na elementach o dużym stopniu scalenia (LSI), 16 bitowe słowo maszynowe, możliwość adresowania 32 K słów pamięci, programowa zgodność z komputerami SM-3 i SM-4, możliwość współpracy z innymi urządzeniami systemu SM. Średni czas międzyawaryjny mikrokomputera powinien być rzędu 1000 h.

Mikrokomputer MERA-60 został poddany badaniom w następującej konfiguracji:

- moduły elektroniczne — P2 (6 szt.), MPR-60 (1 szt.), MDE-60 (1 szt.), MDK-60 (2 szt.), MQU-60 (1 szt.)
- kasetę o wysokości 8U (U = 14,45 mm) z podwójnym zestawem procesora (M2) i zasilaczem
- konstrukcja nośna o wysokości 14U.

Do tak zestawionego mikrokomputera dołączono: monitor ekranowy MERA-7952, drukarkę znakowo-mozaikową DZM-180 KSR, stację pamięci na dyskach elastycznych SP-60 MU, pamięć dyskową SM-5400 (przez sterownik SM-5402).

W celu wykazania powtarzalności parametrów mikrokomputera badano dwa egzemplarze jednocześnie.

Zasadnicze punkty harmonogramu badań mikrokomputera MERA-60 były następujące:

- praca ciągła przez 48 godzin przy wykorzystaniu programów testowych poszczególnych modułów i systemu operacyjnego
- badanie zgodności programowej z minikomputerami SM-3 i SM-4 przez pracę pod systemem operacyjnym RT-60
- badanie czasu wykonania podstawowych operacji z wykorzystaniem specjalnego programu
- badanie zgodności programowej z mikrokomputerem LSI-11 za pomocą oryginalnych programów testowych firmy DEC
- badanie działania mikrokomputera przy zmianach napięcia zasilania
- sprawdzenie zamienności modułów elektronicznych określonego typu przez ich wymianę między testowanymi egzemplarzami mikrokomputera
- pomiar poboru mocy.

Ponadto, niejako automatycznie, dla podanej konfiguracji systemu MERA-60 zbadano:

- obciążalność magistrali dzięki pełnemu obsadzeniu podwójnego zestawu
- adresowalność pamięci w zakresie 28 K słów
- możliwość dołączenia urządzeń systemu SM (drukarka DZM-180 KSR, sterownik SM-5402).

Sprawdzono również zgodność wykonania mikrokomputera z dokumentacją i jej poprawność. Mikrokomputer MERA-60 przeszedł wszystkie badania pomyślnie i został włączony do systemu SM elektronicznych maszyn cyfrowych, kryptonim SM 1633, jako pierwszy polski komputer tej klasy.

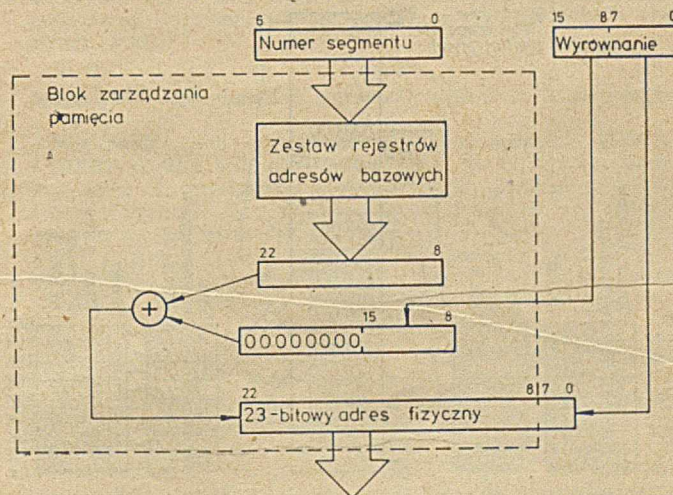
PODSTAWOWE ZAŁOŻENIA SYSTEMU MERA-60/256

Zgodnie z ustalonym planem Rady Głównych Konstruktorów systemu małych maszyn cyfrowych SM, w latach 1985—1995 będą produkowane komputery tzw. trzeciej kolejności [4]. Zakłada się, że będą one miały następujące cechy:

- zwiększoną, praktycznie o rząd wielkości, niezawodność
- konstrukcję przystosowaną do produkcji kilkanaście razy większej od obecnej produkcji komputerów
- architekturę przystosowaną do pracy wieloprocesorowej i pracy w sieci komputerowej, z mechanizmami umożliwiającymi programowanie współbieżne
- lepsze o rząd wielkości parametry eksploatacyjne
- nowe rodzaje urządzeń zewnętrznych i pamięci
- oprogramowanie zgodne z oprogramowaniem swoich poprzedników (tzw. maszyn pierwszej i drugiej kolejności).

W Instytucie Systemów Sterowania podjęto prace nad rozszerzeniem systemu MERA-60 do systemu MERA-60/256. Założono wprowadzenie następujących zmian w stosunku do MERA-60 produkowanej obecnie:

- rozszerzenie magistrali przez wprowadzenie dodatkowych linii danych i adresów oraz linii przerwań (rys. 2, tab. 3),

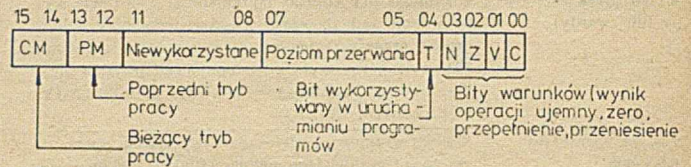


Rys. 2. Organizacja magistral w systemie MERA-60 (liczby w nawiasach dotyczą rozszerzenia magistrali MERA-60/256)

Tabela 3. Zestawienie sygnałów magistrali MERA-60 i MERA-60/256

Funkcje grupy linii	Nazwy sygnałów
Linie danych i adresów	DA00 — DA15 (dla obu magistrali) DA16 — DA17 (tylko dla magistrali MERA-60/256)
Linie sterujące wymianą danych	SIA — sygnał synchronizacji urządzenia aktywnego WYW — sterowanie operacją wyjścia SIP — sygnał odpowiedzi z urządzenia biernego WWO — sterowanie operacją wejścia BAJT — sygnał informacyjny o wprowadzeniu bajtu WU — wybór urządzenia zewnętrznego
Systemowe linie kontrolne	OST — żądanie zatrzymania pracy procesora RGN — regeneracja pamięci PST — napięcie stałe w normie PTN — napięcie zmienne w normie SBR — zerowanie wstępne magistrali PRT — przerwanie od zdarzenia zewnętrznego
Linie obsługi przerwania	Linie przerwania; TPD — żądanie bezpośredniego dostępu do pamięci KPW — potwierdzenie wyboru bezpośredniego dostępu TPR — żądanie przerwania TPR1 — „ „ „ tylko dla magistrali MERA-60/256 TPR2 — „ „ „ TPR3 — „ „ „ Linie zezwoleń; PPD — zezwolenie na bezpośredni dostęp do pamięci PPR — zezwolenie na przerwanie

- rozbudowę pamięci operacyjnej systemu jednoprocusorowego do 256 K bajtów (rys. 3)
- rozbudowanie słowa stanu procesora (rys. 4)
- rozszerzenie listy rozkazów.



Rys. 4. Słowo stanu procesora MERA-6/256 i MERA-6 (do przerwania jednopoziomowych w mikrokomputerze MERA-6) wykorzystuje się jedynie bit 7, niewykorzystane pozostają też pola CM i PM)

Układ zarządzania pamięcią

Przewiduje się, że mikrokomputer będzie stosowany przede wszystkim w systemach wieloprogramowych i wielozadaniowych. W tym celu konstruowany jest układ zarządzania pamięcią MMU (*Memory Management Unit*). Oprócz rozszerzenia fizycznej pamięci do 256 K bajtów będzie on umożliwiał pracę procesora w dwóch trybach KERNEL i USER oraz ochronę wybranych obszarów pamięci. Pełny zestaw funkcji realizowanych przez układ MMU podano w zestawieniu podstawowych parametrów (tab. 4).

W trybie KERNEL program ma pełną kontrolę nad systemem i może realizować wszystkie instrukcje. W trybie tym wykonywany jest monitor i jądro systemowe. W zastosowanych wieloprogramowych i wielozadaniowych, gdy w pamięci operacyjnej może rezydować wiele programów jednocześnie, oprogramowanie w trybie KERNEL spełnia następujące funkcje:

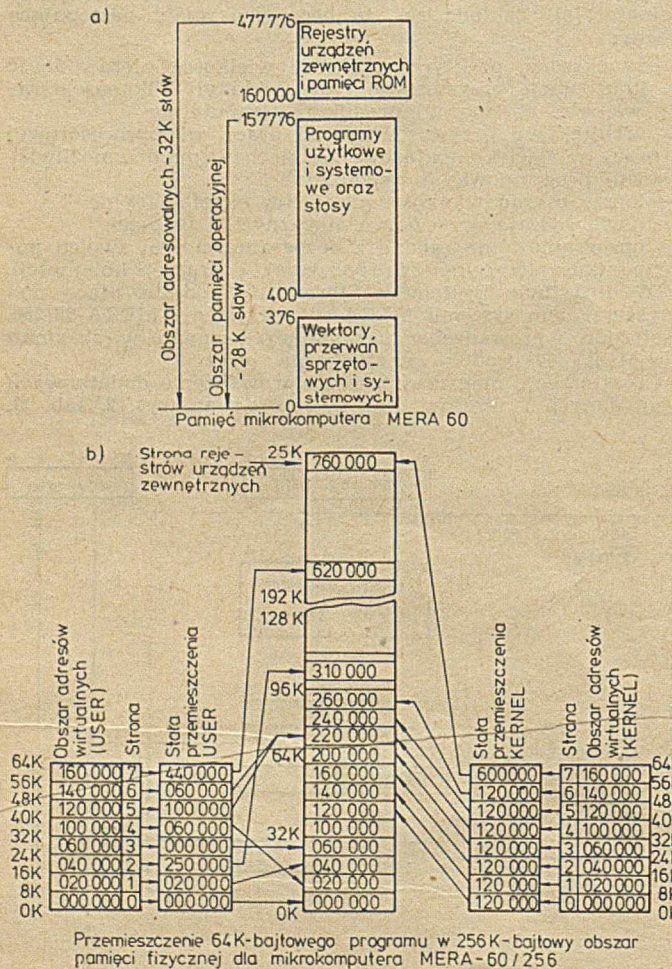
- kontrola wykonania różnych programów użytkowych
- przydział zasobów pamięciowych i sprzętowych
- ochrona integralności systemu jako całości.

W trybie USER programy są wykonywane w ograniczonym zakresie i zabezpieczone przed użyciem niektórych instrukcji mogących zniszczyć oprogramowanie systemowe lub użytkowe. Zadaniem oprogramowania w trybie KERNEL w systemach wielozadaniowych przy wykorzystaniu układu MMU, jest przydzielanie pamięci (tzw. stron) programom użytkowym oraz zabezpieczenie przed niekontrolowanym i niedozwolonym dostępem do innych obszarów pamięci.

Do realizacji opisanych funkcji układ MMU jest wyposażony w dwa zestawy ośmiu 32-bitowych (dostępnych programowo) rejestrów strony aktywnej (APR). Rejestr APR dzieli się na dwa rejestry 16-bitowe: rejestr adresu strony (PAR) i rejestr opisu strony (PDR). Przechowują one pełną informację niezbędną do opisu i przemieszczenia aktualnie aktywnych stron pamięci. Jeden zestaw APR wykorzystuje się w trybie KERNEL, a drugi w trybie USER, co jest określone przez słowo stanu procesora — PSW (rys. 4). Rejestr opisu strony (PDR) zawiera informacje niezbędne dla tworzenia mechanizmów ochrony obszarów pamięci. Układ MMU jest ponadto wyposażony w matrycę rejestrów niezbędnych do realizacji operacji zmiennoprzecinkowych, wykonywanych przez specjalny sprzęt.

Długość słowa mikrokomputera MERA-60/256 wynosi 16 bitów co nie wystarcza do zaadresowania całej pamięci. Dlatego adres nie może być interpretowany jako adres fizyczny (PA), lecz — wirtualny (VA), zawierający informację wykorzystywaną do utworzenia 18-bitowego adresu fizycznego. W celu otrzymania adresu fizycznego, do adresu VA dodaje się informację zawartą w rejestrze PAR. Adres 16-bitowy jest przekształcony na 18-bitowy w sposób automatyczny, niewidoczny dla użytkownika.

Obszar pamięci wirtualnej jest podzielony na 8 stron po 8 K bajtów każda (łącznie 64 K bajtów). Każdej stronie odpowiadają dwa rejestry APR — jeden dla trybu KERNEL, drugi dla trybu USER. Trzy najstarsze bity 16-bitowego adresu wirtualnego określają odpowiadający mu re-



Rys. 3. Organizacja pamięci operacyjnych w systemach MERA-60 i MERA-6/256

Tabela 4. Podstawowe parametry mikrokomputera MERA-60 i zakładane — mikrokomputera MERA-60/256

Parametr	MERA-60	MERA-60/256
	3	2
Długość słowa maszynowego (w bitach)	16	16
Lista rozkazów	Ponad 60 rozkazów podstawowych oraz 8 arytmetycznych i zmienoprzecinkowych (lista rozkazów mikrokomputera LSI-11)	Ponad 60 rozkazów podstawowych i ok. 50 zmienoprzecinkowych, realizowanych przez specjalny układ scalony przy wykorzystaniu układu zarządzania pamięcią (lista rozkazów komputera PDP-11/34r)
Liczba rejestrów uniwersalnych	8	8
Szybkość wykonywania operacji dodawania przy rejestrowanym trybie adresowania (w tys. op./s)	220	ok. 500
Liczba trybów adresowania	8	8 (dodatkowe tryby adresowania dla operacji zmienoprzecinkowych)
Możliwość wykonywania operacji na bajtach	+	+
Możliwość wykonywania operacji z wykorzystaniem stosu	+	+
Bezpośredni dostęp do pamięci	+	+
Liczba poziomów wektorowego systemu przerwań	1	4 (dodatkowe poziomy przerwań zrealizowano dzięki trzem dodatkowym linom przerwań (rys. 2) i dodatkowym bitom w słowie stanu procesora (rys. 4))
Asynchroniczna realizacja operacji na magistrali w trybie MASTER-SLAVE	+	+
Tryb pracy mikrokomputera	Programy lub komunikacja z konsolą — ODT	Programy — KERNEL lub USER, komunikacja z konsolą — ODT (tryby pracy programowej zapewniają układ zarządzania pamięcią)
Mikrokomputer komunikacji z konsolą (ODT)	+	+
Program samoladujący dla taśmy papierowej (ang. bootstrap)	+	— (tzn. nie rezyduje w module procesora)
Maksymalna pojemność pamięci operacyjnej (w K bajtach), tj. wielkość obszaru adresowego zmniejszonego o 8 K bajtów zarezerwowane na rejestry urządzeń zewnętrznych	64-8 = 56	256-8 = 248 (adresowanie zrealizowane za pomocą układu zarządzania pamięcią)
Stronicowa i blokowa organizacja pamięci operacyjnej	—	Obszar pamięci wirtualnej o pojemności 64 K bajtów dzieli się na 16 stron o zmiennej długości od 64 do 8 K bajtów (realizacja przy użyciu układu zarządzania pamięcią)
Ochrona dostępu do zastrzeżonych obszarów pamięci operacyjnej	—	Realizacja przy użyciu układu zarządzania pamięcią, rodzaje ochrony; obszar niedostępny, tylko odczyt, odczyt-zapis

	1	2	3
Detekcja błędu parzystości przy odczycie z pamięci do procesora	—	—	Możliwość tę zapewniają dwie dodatkowe linie magistrali DA16 i DA17 (rys. 2); realizacja przy użyciu układu zarządzania pamięcią
Detekcja zaniku zasilania	—	+	+
Maksymalna liczba modułów elektronicznych	—	6 (zestaw pojedynczy) 12 (zestaw podwójny) 20 (zestaw potrójny)	7 (przy założeniu nie-12 zmienionych stan-21 dardów mechanicznych)
Wymiary mechaniczne	—	Moduł; 135 × 240 mm (pojedynczy) 275 × 240 mm (podwójny) Kaseta; 19 cali × 790 mm × kU n = 14 (24 lub 36) U = 14,45 mm k = 4 (6 lub 8) Konstrukcja nośna; 19 cali × 840 mm × nU	Zakłada się możliwość wprowadzenia innego standardu mechanicznego, np. tzw. eurokarty
Poziomy sygnałów na magistrali (oprócz sygnałów zaniku zasilania)	—	Logiczne 0; +2 — +3,4 V Logiczne 1; 0 — +0,8 V	Logiczne 0; +2 — +3,4 V Logiczne 1; 0 — +0,8 V
Napięcie zasilania	—	220 V +10%, —15% 50 ± 1 Hz	220 V +10%, —15% 50 ± 1 Hz

jest AP. Taka organizacja umożliwia umieszczenie dużych programów w nieciągłych obszarach pamięci fizycznej oraz jednoczesne wykonywanie w tej pamięci kilku programów o tych samych adresach wirtualnych.

Na rysunku 3 przedstawiono 64 K-bajtowy obszar pamięci wirtualnej zajęty przez program użytkowy i jego przykładowe rozmieszczenie w pamięci fizycznej. Poszczególne adresy VA obszaru wirtualnego mogą być przekształcone na większe, mniejsze lub te same adresy w pamięci fizycznej. Dla ułatwienia — początki wszystkich stron dla trybu USER (z wyjątkiem drugiej) zostały przekształcone na adresy będące wielokrotnością 8 K bajtów, choć mogą być przekształcone na adresy stanowiące wielokrotności 64 bajtów (co wynika z organizacji strony, p. tab. 4). Różne strony pamięci wirtualnej mogą być umieszczone w tym samym obszarze pamięci fizycznej, co umożliwia dostęp wielu programów do wspólnych danych komunikacyjnych.

Obszar adresowy dla trybu KERNEL na rysunku 4 przesunięto prawie w całości — tak, aby zachować jego ciągłość. Górne 8 K bajtów obszaru fizycznego jest przeznaczony na stronę urządzeń zewnętrznych, dostępną jedynie w trybie KERNEL. Stałe przemieszczenia określa się na podstawie informacji zawartych w odpowiednich rejestrach PAR.

Magistrala systemowa i pozostałe założenia konstrukcyjne

Większość cech magistrali mikrokomputerów MERA-60 i MERA-60/256 jest identyczna (rys. 2). Wszystkie moduły dołączone do magistrali operują tymi samymi sygnałami. Linie magistrali są dwukierunkowe z wyjątkiem sygnałów zezwolenia na przerwanie i bezpośredni dostęp do pamięci. Stan aktywny każdej z linii (logiczne 1), jest określony przez niski poziom napięcia. Dokładną specyfikę sygnałów magistrali MERA-60 i MERA-60/256 przedstawiono w tabeli 3.

Magistrala określa priorytety urządzeń wejścia-wyjścia. Przerwanie i bezpośredni dostęp do pamięci są realizowane przez sygnały zezwoleń i żądań. Priorytet danego poziomu przerwania jest określony przez sygnały zezwoleń. Najwyższy priorytet ma urządzenie elektrycznie najbliższe procesora (rys. 2). Moduł przekazuje sygnał do urządzenia o niższym priorytecie tylko wówczas, gdy sam nie wymaga obsługi. Magistrala jest drugim, obok procesora, elementem realizacji wektorowego systemu przerwań dla urządzeń wejścia-wyjścia.

Operacje na magistrali MERA-60 i MERA-60/256 są wykonywane w sposób asynchroniczny, a komunikacja między dwoma, dołączonymi do niej urządzeniami, odbywa się według zasady MASTER-SLAVE¹⁾. W danej chwili tylko jedno urządzenie zwane MASTER może mieć kontrolę nad magistralą. Przykładem takiego urządzenia jest procesor pobierający instrukcje z pamięci (z urządzenia typu SLAVE). Oprócz procesora urządzeniem typu MASTER może być każdy moduł realizujący funkcję bezpośredniego dostępu do pamięci.

Magistrala mikrokomputera MERA-60/256, w porównaniu z magistralą mikrokomputera MERA-60, jest rozszerzona o dwie linie adresowe (DA 16, DA 17) i trzy linie żądania przerwania (TPR1, TPR2, TPR3). Wymienione różnice umożliwiają zwiększenie obszaru adresowego i kontrolę parzystości przy przesyłaniu informacji przez magistralę oraz zwiększenie liczby poziomów przerwań.

Z przedstawionych założeń dotyczących budowy mikrokomputera MERA-60/256 wynika, że będzie on zgodny z mikrokomputerem PDP-11/23 firmy DEC (opartym na procesorze LSI-11 [5]). Dotychczas nie ustalono wymiarów pakietu i typu złącza.

W celu zapewnienia możliwości pracy wieloprocesorowej przewiduje się dodatkowe zastosowanie w mikrokomputerze MERA-60/256 szerokiej magistrali 32-bitowej, umożliwiającej arbitraż. Przewiduje się ponadto zminimalizowanie liczby typów modułów przez zwiększenie ich inteligencji i wprowadzenie możliwości zmiany funkcji przez zmianę programu działania. Oddzielnym problemem jest zastosowanie nowych typów urządzeń zewnętrznych.

1) Po polsku: nadrzędny-podległy (przyp. red.)

Program prac nad mikrokomputerem MERA-60/256 zakłada, że zrealizowanie prototypowej konfiguracji nastąpi we współpracy ze Związkiem Radzieckim w drugiej połowie bieżącego roku.

* * *

System mikrokomputerowy MERA-60 jest bez wątpienia nowym elementem w rozwoju sprzętu komputerowego produkowanego w kraju. Rozszerza on znacznie możliwości realizacji systemów sterowania w przemyśle, transporcie, medycynie i innych gałęziach gospodarki. Modułarna budowa umożliwi ciągły rozwój systemu, zarówno sprzętowy, jak i programowy.

Realizacja planów produkcji mikrokomputera MERA-60/256 wymaga — oprócz prac konstrukcyjnych i programistycznych — podjęcia zasadniczych działań w kierunku unowocześnienia technologii produkcji sprzętu mikrokomputerowego. Jedynie przyjęcie nowoczesnej technologii wytwarzania może zapewnić temu sprzętowi wymaganą niezawodność oraz obniżyć koszty.

LITERATURA

- [1] Dokumentacja techniczno-ruchowa systemu MERA-60. Instytut Systemów Sterowania, Katowice, 1981
- [2] Eckhouse R.H.: Systemy minikomputerowe — organizacja i programowanie. WNT, Warszawa, 1979
- [3] Grzywak A., Pregiel R.: System mikrokomputerowy MERA-60 — zastosowanie i architektura. Biuletyn Techniczny MERA, nr 1 (215), str. 5 (1980)
- [4] Materiały Rady Głównych Konstruktorów SM EMC dotyczące maszyn cyfrowych trzeciej kolejności, 1982
- [5] Microcomputer Processors. Digital Equipment Corporation, Maynard, MA, 1981
- [6] PDP-11 Software Handbook. Digital Equipment Corporation, Maynard, MA, 1978/1979.

KONFERENCJE

INFOGRYF'83

W dniach 10—13 maja 1983 r. odbędą się kolejne, VI Kołobrzeskie Dni Informatyki INFOGRYF'83. Tradycyjnymi organizatorami tej konferencji są: TNOiK (oddział w Szczecinie), Instytut Informatyki i Zarządzania Politechniki Szczecińskiej oraz ZETO Szczecin. Obrady konferencji odbywać się będą w kołobrzeskich hotelach „Solny” i „Skanpol”. Podobnie jak w latach poprzednich, celem konferencji jest ocena oraz wymiana doświadczeń w zakresie zastosowań informatyki do zarządzania przedsiębiorstwem. Tematyka konferencji obejmuje sześć grup tematycznych:

1. Systemy informatyczne zarządzania przedsiębiorstwem
2. Metody informatyki w zarządzaniu przedsiębiorstwem
3. Oprogramowanie systemów informatycznych przedsiębiorstw
4. Efektywność zastosowań informatyki w przedsiębiorstwie
5. Stan i perspektywy rozwoju sprzętu i oprogramowania dla potrzeb zastosowań informatyki w przedsiębiorstwie
6. Informatyka a reforma gospodarcza.

Do wymienionych sesji tematycznych można zgłaszać komunikaty o objętości do 5 stron maszynopisu w terminie do 15 marca 1983 r. Organizatorzy zastrzegają sobie prawo selekcji komunikatów (do publikacji i wygłoszenia).

W ramach imprez towarzyszących przewiduje się zorganizowanie GIELDY NOWOŚCI INFORMATYKI, mającej na celu popularyzację nowych rozwiązań sprzętowych i programowych, a zwłaszcza rozpowszechnienie informacji o możliwościach ich nabycia. Do udziału w Gieldzie organizatorzy zapraszają zarówno przedsiębiorstwa, jak i osoby fizyczne, zajmu-

jące się oprogramowaniem, a także producentów sprzętu komputerowego. Przewiduje się wydanie katalogu Gieldy, w którym informacje będą prezentowane w następującym układzie:

- nazwa produktu
- właściciel
- charakterystyka produktu
- stosowany sprzęt (dotyczy oprogramowania)
- ograniczenia
- dotychczasowe wdrożenia
- warunki zakupu
- autorzy.

Przy zgłoszeniach powyższe informacje należy przedstawić na dwóch stronach maszynopisu (max.). Przewiduje się możliwość prezentacji zgłoszonych rozwiązań w stoiskach lub w ramach specjalnych prelekcji. Odpłatność za umieszczenie w katalogu — ok. 2000 zł.

Zgodnie z tradycją INFOGRYFU przewiduje się zorganizowanie imprez towarzyszących (wieczory autorskie, spotkania, dyskusje panelowe).

Warunki uczestnictwa

Zgłoszenia udziału w konferencji przyjmowane są do 15 marca 1983 r. W zgłoszeniu należy zadeklarować numery grup tematycznych. Koszt uczestnictwa wynosi 8250 zł. od osoby i obejmuje: zakwaterowanie i wyżywienie (rozliczane w ramach obowiązującej diety — 4 doby po 640 zł) oraz materiały konferencyjne. Po otrzymaniu zgłoszenia i dokonaniu wpłaty na konto NBP I O/M Szczecin nr 81012-420-132 — zostanie przekazany szczegółowy program obrad, natomiast materiały konferencyjne uczestnicy otrzymają na miejscu w Kołobrzegu.

Szczegółowych informacji udziela Biuro Oddziału TNOiK w Szczecinie, ul. Pocztowa 30/12, tel. 447-25.

Wybrane mikrokomputerowe bloki sterujące systemu CAMAC i mikrokomputery o przeznaczeniu uniwersalnym

CAMAC jest uniwersalnym, modułowym systemem elektronicznym, stosowanym do automatyzacji zarówno procesów pomiarowych w eksperymentach naukowych, jak i sterowania procesami technologicznymi [2]. Dzięki możliwości tworzenia różnorodnych zestawów funkcjonalnych aparatura CAMAC znajduje szerokie zastosowanie w wielu dziedzinach nauki i techniki (p. INFORMATYKA nr 4—5/1982, System CAMAC — numer monograficzny).

Parametry i właściwości urządzeń przeznaczonych do pracy w zestawach CAMAC powinny odpowiadać wymaganiom określonym w dokumentach międzynarodowego komitetu ESONE (European Standards on Nuclear Electronics). W kraju, odpowiednie zalecenia normalizacyjne są sprecyzowane w Polskich Normach.

System CAMAC zapewnia dwustronne połączenia i przekazywanie informacji pomiędzy układem pomiarowym a komputerem lub innymi urządzeniami sterującymi. Blokowa budowa systemu umożliwia łatwe modyfikowanie zestawów pomiarowych CAMAC. Aparatura pomiarowa jest tworzona z poszczególnych bloków funkcjonalnych, zestawionych w kasetach specjalnie dostosowanych do tego celu.

Kaseta CAMAC obejmuje 25 stanowisk. Zazwyczaj dwa prawie skrajne stanowiska są przeznaczone dla bloku sterującego kasetą. Na pozostałych — umieszcza się bloki wykonawcze, takie jak: przetworniki analogowo-cyfrowe i cyfrowo-analogowe, bloki sterowania i współpracy z urządzeniami zewnętrznymi, bloki przetwarzające informację (np. konwertery kodów), bloki pamięci, bloki sprzęgające itp. Wymiana informacji (w postaci sygnałów elektrycznych) w obrębie jednej kasety jest dokonywana za pośrednictwem znormalizowanej magistrali.

System CAMAC umożliwia tworzenie zestawów wielokasetowych przez łączenie poszczególnych kaset za pomocą magistrali zewnętrznych oraz — użycie kilku komputerów. Możliwe jest również wykorzystanie kilku źródeł sterowania wewnątrz jednej kasety. Ta całkowicie odmienna koncepcja organizacji aparatury CAMAC, określana mianem systemu o zdecentralizowanej inteligencji, stanowi podstawę do tworzenia zestawów o znacznie szerszych możliwościach użytkowych.

Postanowienia dotyczące organizacji logicznej oraz wymagań mechanicznych systemu CAMAC są zawarte w następujących normach:

- dla konfiguracji jednokasetowej — PN 72/T 06530
- dla konfiguracji wielokasetowej w systemie szeregowym — PN 75/T 06532, w systemie równoległym — PN 80/T 06535
- dla zestawów zawierających wiele źródeł sterowania w obrębie jednej kasety — BN 80/5620-05.

Bloki sterujące kasety CAMAC, w tym również — przeznaczone do pracy w systemie o zdecentralizowanej inteligencji, są z powodzeniem realizowane w postaci mikrokomputerów, zbudowanych z elementów o dużym stopniu scalenia [3, 4].

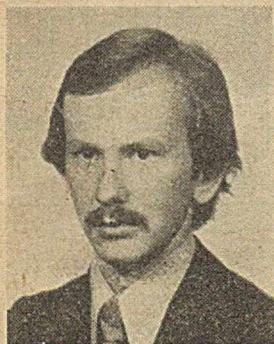
Poniżej zostaną omówione niektóre mikrokomputerowe sterowniki kaset CAMAC, opracowane i wykonane w Instytucie Badań Jądrowych w Świerku. Pierwszy z nich — CMC 8080 jest sterownikiem przeznaczonym do zestawów jednokasetowych. Drugi, oznaczony symbolem ABS 80, nadaje się do wykorzystania w systemach o zdecentralizowanej inteligencji.

Sterowniki CMC 8080 i ABS 80 oraz trzeci — ABSP 8080 opracowano w Pracowni Mikrokomputerów Zakładu Elektroniki Jądrowej, w latach 1977—1981. Sterownik CMC 8080 należał do pierwszej generacji sterowników z mikroprocesorami i został wykonany w ramach współpracy z Instytutem Badań Jądrowych w Seibersdorf (Austria).

BLOKI STERUJĄCE KASETY CAMAC Z MIKROPROCESOREM (MIKROKOMPUTERY)

Bloki sterujące kasety CAMAC często konstruuje się w oparciu o układy mikroprocesorowe. Wymagania, jakie stawia się takim rozwiązaniom, są następujące:

- zgodność z dokumentami normalizacyjnymi
- pełne wykorzystanie możliwości systemu mikroprocesorowego
- dopasowanie konfiguracji mikrokomputera do przeznaczenia bloku
- możliwość modyfikowania konfiguracji mikrokomputera
- możliwość rozbudowy mikrokomputera, np. przez dołączenie dodatkowej pamięci, w celu zapewnienia bezpośrednio współpracy z urządzeniami zewnętrznymi lub zastosowania w systemie wieloprocesorowym.



Mgr inż. SLAWOMIR JAGIELLO ukończył w 1974 r. studia na Wydziale Elektroniki Politechniki Warszawskiej. Od 1973 r. pracował w Biurze Konstrukcyjno-Technologicznym NPCP—CEMI, gdzie zajmował się problemami automatycznego testowania układów scalonych. Od 1978 r. pracuje w Instytucie Badań Jądrowych, specjalizując się w problematyce techniki mikrokomputerowej.



Mgr inż. ANDRZEJ KOZMIŃSKI ukończył w 1982 r. studia na Wydziale Elektroniki Politechniki Warszawskiej, specjalność — aparatura elektroniczna. Pracuje w Instytucie Badań Jądrowych w Świerku w Pracowni Mikrokomputerów na stanowisku konstruktora.

Spośród wielu systemów mikroprocesorowych, dotychczas najczęściej stosowano 8-bitowe mikroprocesory ze stałą listą rozkazów, produkcji firm MOTOROLA (system 6800), INTEL (system 8080 i 8085), ZILOG (system Z80) lub ich odpowiedniki produkcji innych firm. Rodzaj wybranego systemu mikroprocesorowego decyduje o właściwościach sterownika.

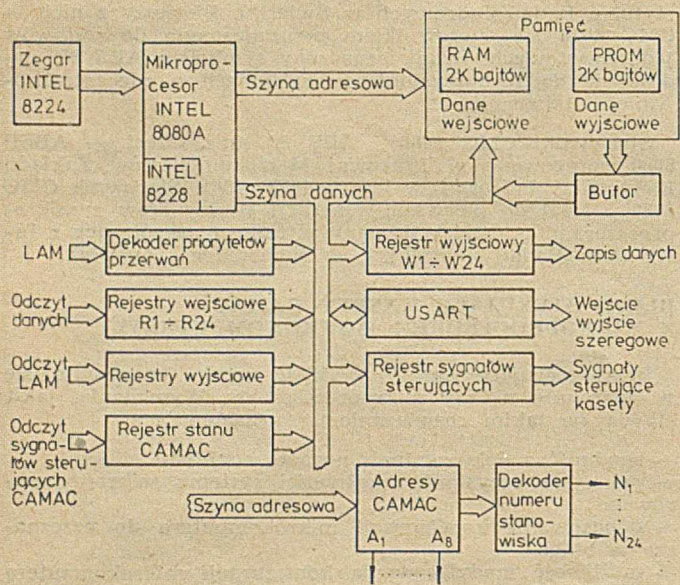
Podstawowa konfiguracja mikrokomputera w systemie CAMAC (sterownika kasyety CAMAC) zawiera dwa podstawowe zespoły układów: jednostki centralnej i logiki CAMAC. W skład jednostki centralnej wchodzi typowe układy mikrokomputera, a więc: mikroprocesor, pamięć programu (PROM), pamięć danych (RAM), układy wejścia-wyjścia.

Autonomiczny sterownik kasyety CAMAC — CMC 8080 [5]

Blok sterownika CMC 8080 jest przeznaczony do sterowania pojedynczego zestawu pomiarowego lub zestawu przeznaczonych do automatycznej regulacji procesów technologicznych. W skład sterownika (rys. 1) wchodzi następujące układy:

- mikroprocesor INTEL 8080A
- pamięć stała PROM, o pojemności 2 K słów 8-bitowych
- pamięć o dostępie swobodnym RAM, o pojemności 2 K słów 8-bitowych
- uniwersalny nadajnik-odbiorca asynchroniczny, umożliwiający współpracę z dalekopisem
- programowy zegar czasu rzeczywistego
- logika CAMAC.

Mikroprocesor INTEL 8080A współpracuje bezpośrednio ze standardowymi układami sterownika systemowego (INTEL 8228) i generatora impulsów zegarowych (INTEL 8224). Ze względu na małą obciążalność szyny mikroprocesora są buforowane.



Rys. 1. Autonomiczny sterownik kasyety CAMAC — CMC 8080



Mgr inż. KRZYSZTOF RZYMKOWSKI studia ukończył na Wydziale Elektroniki na Politechnice Warszawskiej w 1968 r. Pracę zawodową rozpoczął w 1967 r. w Zakładzie Doświadczalnym w Biurze Urzędów Techniki Jądrowej w Warszawie. Obecnie pracuje w Instytucie Badań Jądrowych (od 1973 r. na stanowisku kierownika Pracowni ds. systemu CAMAC, a od 1977 r. na stanowisku kierownika Pracowni Mikrokomputerów w Zakładzie Elektroniki Jądrowej).

Blok pamięci mikrokomputera składa się z pamięci stałej, przechowującej program sterujący oraz pamięci o dostępie swobodnym. Pamięć stałą zbudowano z układów programowalnej pamięci INTEL 1702, a w pamięci o dostępie swobodnym zastosowano układy INTEL 2102-4.

Dzięki użyciu uniwersalnego, asynchronicznego nadajnika-odbiorcy, zamieniającego informację równoległą na standardowy przebieg szeregowy (układ scalony TMS 6011), możliwe jest sprzężenie mikrokomputera z dalekopisem. Złącze dalekopisowe może również służyć do komunikacji z innymi urządzeniami zewnętrznymi, np. z innym komputerem.

Obsługę zgłoszeń bloków CAMAC, dalekopisu oraz zegara zorganizowano przez 8-poziomowy układ przerwań, umożliwiający maskowanie poszczególnych sygnałów przerywających. Jako zegar programowy wykorzystano układ scalony MK 5009P.

Druga zasadnicza część sterownika CMC 8080, logika CAMAC — spełnia następujące funkcje:

- pośredniczy w wymianie informacji pomiędzy szynami danych kasyety CAMAC (24 bity) a magistralą mikroprocesora (8 bitów)
- rejestry W i R
- generuje adresy CAMAC oraz sygnały wyboru bloku N_1-N_{24}
- generuje sygnały cyklu CAMAC
- steruje rezerwacją magistrali.

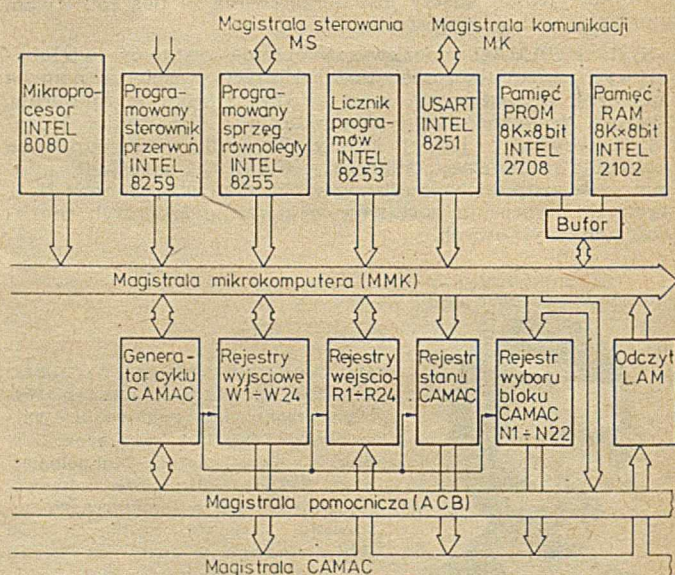
Autonomiczny blok sterowania kasyetą CAMAC — ABS 80 [1]

Sterownik ABS 80 jest mikrokomputerem przeznaczonym do sterowania kasyetą CAMAC, zbudowanym w oparciu o mikroprocesor INTEL 8080. W stosunku do sterownika CMC 8080 — ABS ma rozszerzone możliwości i zapewnia realizację następujących funkcji:

- sterownika kasyety
- sterownika pomocniczego
- uniwersalnego szeregowego kanału łączności, umożliwiającego sprzężenie z urządzeniem zewnętrznym, innym komputerem lub dowolnym urządzeniem mającym dopasowany kanał transmisji przez magistralę komunikacyjną MMK oraz programowalny nadajnik-odbiorca (USART) INTEL 8251.

Ponadto możliwe jest:

- tworzenie systemów wieloprocessorowych ze wspólną pamięcią oraz systemów o zdecentralizowanej inteligencji, w oparciu o magistralę mikrokomputera MMK
- wykorzystanie sterownika poza kasyetą, po odłączeniu płytki wewnątrz bloku
- bezpośrednie sterowanie obiektem przez magistralę sterowania MS i programowalny układ INTEL 8255.



Rys. 2. Autonomiczny blok sterowania kasyetą CAMAC — ABS 80

Schemat blokowy sterownika ABS 80 przedstawiono na rysunku 2. Mikrokomputer zrealizowano w konfiguracji podobnej do omówionego poprzednio, tj. w oparciu o mikroprocesor INTEL 8080A, sterownik systemowy INTEL 8228 i generator zegarowy INTEL 8224.

W stosunku do mikrokomputera CMC 8080 istotnie powiększono pojemność pamięci. Sterownik ABS 80 zawiera 8 K bajtów pamięci PROM i 8 K bajtów pamięci RAM. Układy scalone pamięci stałej zamontowano w podstawkach, aby umożliwić łatwą wymianę programu bloku sterującego.

W układzie przerwań mikrokomputera ABS 80 wykorzystano programowy sterownik przerwań INTEL 8259. Uzyskano w ten sposób możliwość programowego ustalania priorytetów z przydziałem zmienianym w sposób dynamiczny.

W celu umożliwienia bezpośredniej współpracy z urządzeniami zewnętrznymi, sterownik ABS 80 wyposażono w programowalny nadajnik-odbiornik asynchroniczny (USART) INTEL 8251. Jako generator przebiegu zegarowego zastosowano scalony zespół programowalny liczników 16-bitowych — INTEL 8253. Jedna sekcja licznika, pracująca jako dzielnik częstotliwości, generuje wymagany przebieg prostokątny, a dwie pozostałe — mogą być wykorzystywane dowolnie.

Mikrokomputer ABS 80 zawiera również układ sprzężenia równoległego INTEL 8255 z trzema 8-bitowymi rejestrami wejścia-wyjścia, w konfiguracji określonej programowo.

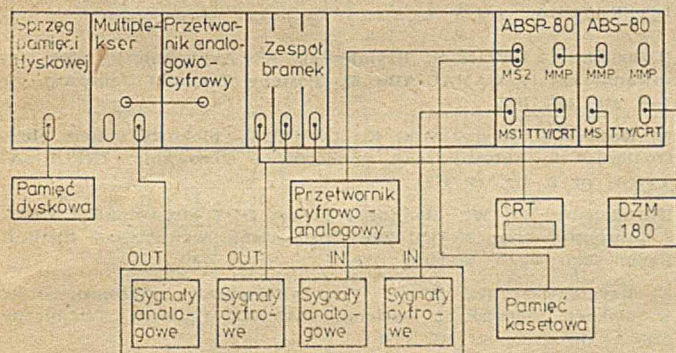
Część sprzęgająca zapewnia połączenie mikrokomputera z magistralą kasyety CAMAC. W jej skład wchodzi trzy 8-bitowe rejestry linii R, trzy 8-bitowe rejestry linii W, rejestr adresów i funkcji CAMAC, rejestr stanu CAMAC, układ generacji cyklu CAMAC, dekodery linii N i układ rezerwacji magistrali.

Rola mikrokomputera pomocniczego — ABSP 80

Blok ABS 80 może współpracować według zasady nadrzędny — podległy (ang. *master-slave*) z innym mikrokomputerem opracowanym w IBJ — ABSP 80, ABS 80 odgrywa w takim połączeniu rolę bloku nadrzędnego, inicjując wykonanie programów zawartych w pamięci PROM bloku ABSP 80 lub przepisując do tego bloku program z własnej pamięci.

Mikrokomputer pomocniczy ABSP 80 ma architekturę podobną do sterowników omówionych powyżej, lecz nie zawiera układów pośredniczących we współpracy z systemem CAMAC — komunikacja z magistralą kasyety może odbywać się tylko przez blok sterownika głównego. Oprogramowanie mikrokomputera ABSP 80 jest dostosowywane każdorazowo do konkretnego zastosowania.

Komunikacja mikrokomputera ABSP 80 z mikrokomputerem nadrzędnym (ABS 80) odbywa się za pośrednictwem wejściowego i wyjściowego rejestru sygnałów sterujących. Trzy bity słowa sterującego służą do bezpośredniego przekazywania sygnałów RESET, INT i HOLD do jednostki centralnej bloku pomocniczego. Pozostałe pięć bitów przekazuje zakodowane polecenia sterujące pracą bloku pomocniczego. Przesyłanie informacji w przeciwnym kierunku odbywa się w podobny sposób, przy użyciu rejestru wyjściowego sygnałów sterujących ABSP 80.



Rys. 3. Wykorzystanie mikrokomputerów ABS 80 i ABSP 80 w zestawie kontrolno-pomiarowym

Przykładowy zestaw aparatury kontrolnej, wykorzystującej mikrokomputery ABS 80 i ABSP 80 (pracujące w systemie nadrzędny-podległy) przedstawiono na rysunku 3. Wyjście cyfrowe sterowanego obiektu jest połączone bezpośrednio z mikrokomputerem ABS 80 i zespołem bramek. Sygnały analogowe są wstępnie selekcjonowane programowo w bloku multiplexera i przetwarzane na postać cyfrową w bloku przetwornika analogowo-cyfrowego.

Informacja pomiarowa może być gromadzona w pamięciach obu mikrokomputerów, w pamięci dyskowej, lub — przedstawiona w postaci wydruku. Możliwe jest również wykorzystanie układu wyświetlającego i pamięci kasetowej.

Sygnały sterujące obiektem, cyfrowe i analogowe (odpowiednio przetworzone w bloku przetwornika cyfrowo-analogowego) są wysyłane bezpośrednio przez magistralę sterującą bloku ABSP 80.

W przedstawionym zestawie aparatury kontrolnej możliwe jest zorganizowanie współpracy źródeł sterujących w konfiguracji wieloprocesorowej, ze wspólnym obszarem pamięci wykorzystywanym do wzajemnej komunikacji.

Oprogramowanie

Obydwa przedstawione bloki sterujące kaset CAMAC wyposażono w programy, tzw. monitory, sterujące wykonaniem podstawowych funkcji użytkowych. Są one napisane w języku symbolicznym mikroprocesora INTEL 8080 i przechowywane w górnej części pamięci stałej (PROM). W pozostałych obszarach pamięci można umieszczać procedury użytkowe.

Monitor mikrokomputera CMC 8080 umożliwia wpisywanie programu z dalekopisu ASR 33 do pamięci RAM, wykonywanie zapisanego programu, wypisywanie zawartości oraz programowanie pamięci PROM (przy użyciu programatora zbudowanego jako dodatkowy blok CAMAC). Program zajmuje 1200 komórek pamięci. Wywołanie procedur realizujących poszczególne funkcje odbywa się za pomocą pięciu jednoliterowych poleceń.

Blok sterujący ABS 80 wyposażono w program monitora o znacznie szerszych możliwościach. Oprócz funkcji wymienionych powyżej możliwe jest:

- wykonanie programu użytkowego od zadanego adresu
- drukowanie zawartości wybranego obszaru pamięci
- badanie i zmiana zawartości komórek pamięci, rejestrów, akumulatora, wskaźnika stosu, licznika rozkazów
- ustawianie punktów wstrzymania, parametru omijania i usuwania poszczególnych lub wszystkich punktów wstrzymania
- kontynuowanie programu po dojściu do punktu wstrzymania
- wczytywanie i perforowanie taśmy papierowej w formacie szesnastkowym mikroprocesora INTEL
- wprowadzanie danych w postaci szesnastkowej do wybranego obszaru pamięci
- przeszukiwanie wybranego obszaru pamięci
- zachowanie stanu przerywacza przerwań.

Program monitora jest wykonywany bezpośrednio po włączeniu zasilania lub po sygnale zerowania systemu. Pierwsza faza wykonania programu obejmuje: zablokowanie przerywacza przerwań, utworzenie tablicy punktów wstrzymania, ustawienie stosu, wydruk komunikatu zgłoszenia, wydruk komunikatu gotowości do pobrania polecenia i wykonania go.

Program akceptuje ponad dwadzieścia jednoliterowych poleceń, wywołujących procedury realizujące poszczególne funkcje. Pojawienie się nieznanego polecenia lub błędnego znaku powoduje przerwanie wykonywania programu, powrót do monitora i wydruk komunikatu gotowości do dalszej pracy.

Ponadto monitor zawiera programy pomocnicze, używane przy obsłudze przerwań. Umożliwiają one zapamiętanie, a następnie odtworzenie zawartości rejestrów i akumulatora po zgłoszeniu przerwania, umożliwiają też blokowanie i odblokowywanie przerywacza przerwań.

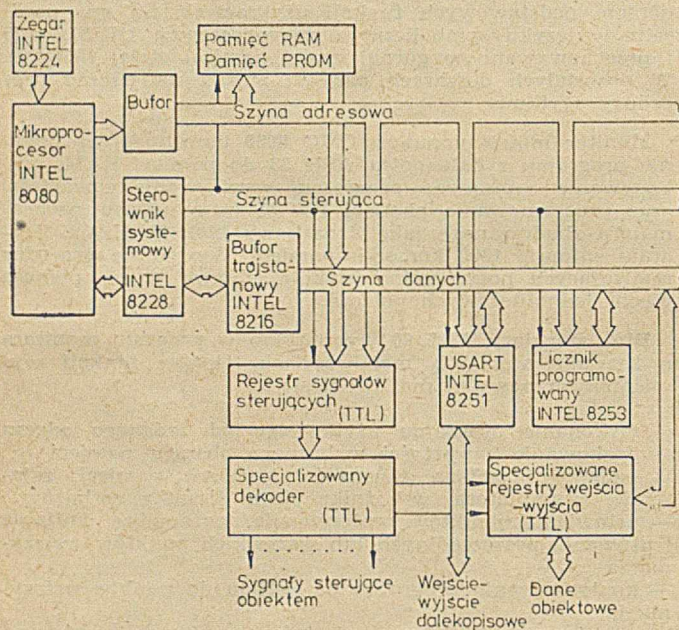
UNIwersalny sterownik mikrokomputerowy ABM 80

Stosowanie aparatury elektronicznej wykonanej w standardzie CAMAC nie zawsze jest wygodne i opłacalne. Dotyczy to przede wszystkim zadań wymagających użycia

pojedynczych urządzeń elektronicznych do sterowania unikalnymi procesami pomiarowymi lub technologicznymi, wykonywania prostych obliczeń arytmetycznych i operacji przetwarzania danych. W tych zastosowaniach mały stopień komplikacji zadania nie uzasadnia wysokich kosztów związanych z wykorzystaniem komputera. W takich przypadkach najwygodniejsze są uniwersalne sterowniki mikrokomputerowe, wyposażone w program sterujący odpowiedni do wykonywanego zadania. Podstawowym celem przy projektowaniu bloków sterujących tego rodzaju jest tworzenie układów uniwersalnych, które można łatwo adaptować. Stanowi to o ich przewadze w wymienionych zastosowaniach nad specjalizowanymi sterownikami przeznaczonymi do pracy w kasetach CAMAC.

Schemat uniwersalnego sterownika ABM 80, wykonanego w Instytucie Badań Jądrowych, przedstawiono na rysunku 4. Sterownik zawiera dwie zasadnicze części:

- uniwersalny, a jednocześnie prosty układ mikrokomputera
- specjalizowane układy wejścia-wyjścia przeznaczone do konkretnego zadania.



Rys. 4. Uniwersalny sterownik mikrokomputerowy ABM 80

Mikrokomputer sterownika ABM 80 zbudowano w oparciu o mikroprocesor INTEL 8080A, w konfiguracji omówionej uprzednio. Wykorzystanie układu licznika programowego i układu szeregowej transmisji danych (USART) umożliwia dołączenie mikrokomputera ABM 80 do urządzeń zewnętrznych np. do dużego systemu komputerowego. Stanowi to ogromne ułatwienie w fazie uruchamiania oprogramowania urządzeń i stwarza możliwości posługiwania się takimi narzędziami, jak: asembler, edytor i inne programy wchodzące w skład oprogramowania podstawowego. W tym celu niezbędne jest umieszczenie prostego programu monitora w pamięci PROM mikrokomputera. Steruje on transmisją danych przez sprzęg szeregowy (USART) oraz umożliwia inicjowanie i sprawdzanie wykonania programu użytkowego, wstępnie zapisanego w pamięci RAM.

Obecnie, w Pracowni Mikrokomputerów Zakładu Elektroniki Jądrowej IBJ opracowuje się projekt zastosowania sterownika ABM 80 do współpracy z miernikiem stężenia pierwiastków radioaktywnych w próbkach materiałów stosowanych w budownictwie. Zadaniem bloku sterującego jest zbieranie danych pomiarowych, dokonywanie niezbędnych obliczeń i prezentacja wyników w formie wydruku (przewiduje się współpracę z drukarką RD-102).

W systemie uruchomieniowym wymienionego urządzenia wykorzystano prosty — lecz w pełni wystarczający — program monitora, zajmujący ok. 300 bajtów pamięci. Po zakończeniu testowania urządzenia, na miejsce monitora wpi-

suje się właściwy program użytkowy, a układy licznika programowanego i nadajnika-odbiornika można usunąć, o ile nie są wykorzystywane w urządzeniu (w tym celu montuje się je na podstawkach).

Sterownik ABM 80 zastosowano także w konstrukcji automatycznego, filtracyjnego miernika zapylenia powietrza i gazów, pracującego w oparciu o metodę radiometryczną. Mikrokomputer steruje tu pracą układów wykonawczych miernika, umieszczonych w kominie fabrycznym. Wymiana informacji pomiędzy urządzeniami wykonawczymi a sterownikiem jest dokonywana za pośrednictwem linii o długości ok. 200 m. Komunikaty o aktualnym stanie układu mogą być wyświetlane lub drukowane przez rejestrator cyfrowy. W algorytmie sterowania przewidziano również możliwość pracy krokowej, w celu dokonywania kontroli bloków wykonawczych.

* * *

Zastosowanie mikroprocesorów ze stałą listą rozkazów (np. INTEL 8080) do budowy zestawów CAMAC jest ograniczone małą szybkością pracy takich zestawów. Przygotowanie operacji zapisu 24-bitowego słowa CAMAC zajmuje około 50 μ s, co jest czasem bardzo długim w porównaniu z cyklem CAMAC, trwającym około 1 μ s. Powstawanie tak znacznych opóźnień jest spowodowane głównie różnicą długości słów mikroprocesora (8 bitów) i systemu CAMAC (24 bity) oraz stosunkowo małą szybkością pracy mikroprocesora o stałej liście rozkazów.

W przypadku, gdy szybkość działania systemu jest warunkiem krytycznym, można z powodzeniem stosować sterowniki z procesorami specjalnie dostosowanymi do standardu CAMAC (np. procesor autonomiczny 130 i 131).

Duża atrakcyjność układów o wielu źródłach sterowania sprawia, że obecnie Komitet ESONE prowadzi intensywne prace nad nowym systemem aparatury elektronicznej, w pełni odpowiadającym nowym wymaganiom. System ten, znany jako E3S (ESONE Small System Standard) może odegrać istotną rolę we wprowadzaniu mikrokomputerów do sterowania nowoczesnymi zestawami pomiarowymi. Przewidziano w nim zarówno współpracę kilku źródeł sterujących w ramach jednej kasy, jak i użycie układów mikrokomputerowych do sterowania pracą poszczególnych bloków wykonawczych.

Dostosowanie urządzenia do pracy w standardowym systemie elektronicznym, jakim jest CAMAC, wymaga rygorystycznego przestrzegania norm dotyczących konstrukcji mechanicznej i organizacji logicznej systemu. W wielu przypadkach praktycznych, wynikająca stąd komplikacja układu i mechaniczna nie jest uzasadniona. Znacznie prostsze i tańsze jest w takiej sytuacji wykorzystanie układów uniwersalnych, łatwo dostosowalnych do aktualnych potrzeb. Doskonale nadają się do tego celu uniwersalne układy mikrokomputerowe, mające bardzo szerokie możliwości, a jednocześnie umożliwiające dokonywanie łatwych zmian w algorytmie pracy — przez wymianę programu sterującego. Przykładem takiego sterownika jest omówiony mikrokomputer ABM 80.

LITERATURA

- [1] Jagiello S., Koślacz S., Rzymkowski K.: Autonomiczny blok sterowania kasy CAMAC-ABS 80. Postępy Techniki Jądrowej, nr 7-8, str. 431, 1981
- [2] Koślacz S., Rzymkowski K.: CAMAC — blokowy system elektroniczny do automatyzacji pomiarów i sterowania, INFORMACYKA, nr 2, str. 14, 1980
- [3] Michalski M. W.: Monitor i pakiet DDT sterownika CAMAC z mikroprocesorem INTEL 8080. Opracowanie wewnętrzne, Instytut Badań Jądrowych, Świerk, 1980
- [4] Rzymkowski K., Witort P.: Zastosowanie mikroprocesorów w blokach sterujących systemy CAMAC. Biuletyn Techniczny MERA, nr 8(198), str. 11, 1978
- [5] Rzymkowski K., Witort P.: Autonomiczny blok sterowania CMC 8080. Opracowanie wewnętrzne, Instytut Badań Jądrowych, Świerk, 1979.

Modułowy system sterowników mikroprocesorowych MIKROSTER

Koncepcja budowy modułowych sterowników mikroprocesorowych jest realizowana przez wiele firm, zajmujących się stosowaniem techniki mikroprocesorowej w przemyśle. Potrzeba opracowania zunifikowanej bazy konstrukcyjnej do budowy sterowników urządzeń i linii technologicznych wynika z konieczności skrócenia cyklu projektowania i wykonania urządzeń, zwiększenia ich niezawodności, a także zapewnienia odpowiednich warunków konserwacji.

Modułowy system sterowników mikroprocesorowych — MIKROSTER, opracowany w Przemysłowym Instytucie Elektroniki, umożliwia konstruowanie sterowników programowanych oraz budowę systemów wspomagania projektowania, niezbędnych do uruchamiania i testowania oprogramowania użytkowego. Jest przeznaczony do:

- sterowania urządzeniami i liniami technologicznymi
- sterowania urządzeniami pomiarowo-kontrolnymi
- przygotowywania i uruchamiania programów użytkowych
- przetwarzania danych.

OPIS OGÓLNY SYSTEMU MIKROSTER

Mikrokomputerowy system MIKROSTER zrealizowano w oparciu o elementy rodziny INTEL 8080. Przy jego projektowaniu starano się wykorzystać przede wszystkim elementy, które mają być produkowane w kraju.

System MIKROSTER ma strukturę modułową. Składa się z szeregu bloków funkcjonalnych — pakietów, które można podzielić na następujące grupy (rys. 1):

- pakiety podstawowe — jednostki centralnej, wielofunkcyjnych mikrokomputerów, pamięci RAM i EPROM, re-

lowanej pamięci EPROM, współpracy z urządzeniami zewnętrznymi, pracy krokowej

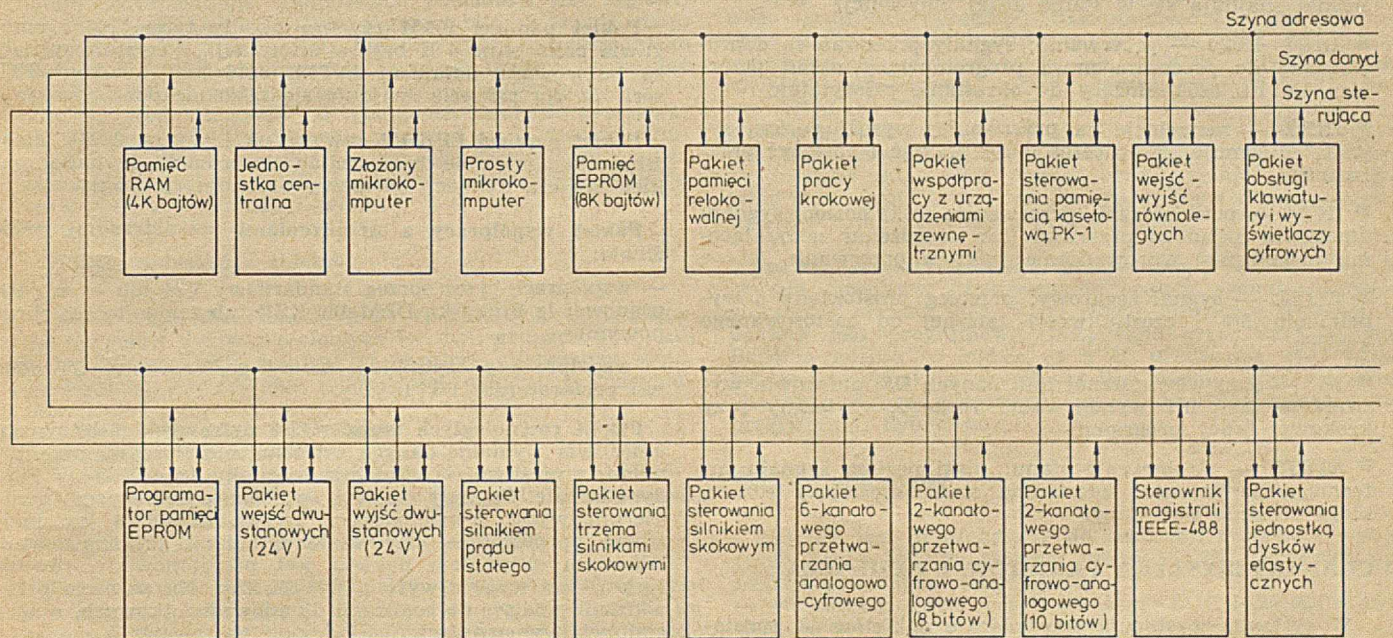
- pakiety specjalizowane — programatora pamięci EPROM, obsługi klawiatury i wyświetlaczy, sterowania pamięcią kasetową PK-1, sterowania dyskami elastycznymi
- pakiety sprzężeń z obiektem — wejść dwustanowych 24 V, wyjść dwustanowych 24 V, sterowania silnikami skokowymi, sterowania silnikiem prądu stałego, sprzężeń z dwoma przetwornikami cyfrowo-analogowymi typu DAC 44 lub DAC 45 (produkcji Przemysłowego Instytutu Telekomunikacji), jednokanałowego lub sześciokanałowego przetwarzania analogowo-cyfrowego.

Podstawowe dane techniczne systemu są następujące:

- wymiary pakietów — 140 × 150 mm (druk dwustronny z metalizacją otworów)
- napięcia zasilania ± 5 V, ± 12 V, ± 15 V
- równoległa magistrala systemowa do transmisji informacji między pakietami przy użyciu łącz ELTRA 83/084
- zewnętrzne złącza do komunikacji z otoczeniem, umieszczone w przedniej części pakietów
- wstępne dekodowanie adresów na magistrali
- pojedyncze (wyjątkowo podwójne) obciążenie magistrali przez poszczególne pakiety.

Magistrala systemowa

Wewnętrzna magistrala systemowa składa się z szyny adresowej (linie AB15—AB Φ), szyny danych (linie DB7—DB Φ) i szyny sterującej, obejmującej kilkanaście linii o różnym przeznaczeniu.



Rys. 1. Zestawienie bloków funkcjonalnych systemu MIKROSTER

Szyna adresowa umożliwia adresowanie 64 K bajtów pamięci lub 256 urządzeń wejściowych i 256 urządzeń wyjściowych. Dzięki wstępnemu dokodowaniu bitów A13, A14 i A15 (na linii SB7—SB0) możliwe jest zmniejszenie liczby użytych dekodów adresów pamięci i portów wejścia-wyjścia. W przypadku wykonywania rozkazów wejścia-wyjścia (IN, OUT) linie A13, A14, A15 powtarzają stan linii A5, A6, A7. Najmniej znaczącym bitem szyny adresowej jest bit AB0.

Szyna danych umożliwia przesyłanie 8 bitów danych między jednostką centralną, pamięcią i urządzeniami wejścia-wyjścia. Najmniej znaczącym bitem szyny danych jest bit DB0.

Poniżej omówiono najważniejsze sygnały szyny sterującej:

- **MEMR*** — sygnał strobujący odczyt z pamięci, niski poziom sygnału warunkuje odczyt informacji z pamięci RAM lub ROM
- **MEMW** — sygnał strobujący zapis do pamięci; niski poziom sygnału warunkuje zapis informacji do pamięci
- **I/OR** — sygnał strobujący odczyt urządzeń wejściowych; niski poziom sygnału warunkuje odczyt informacji z portów wejściowych na szynę danych systemu
- **I/OW** — sygnał strobujący zapis do urządzeń wyjściowych; niski poziom sygnału warunkuje zapis informacji do portów wyjściowych
- **MID** — blokada pamięci i portów wejścia-wyjścia; niski poziom sygnału blokuje dostęp do pamięci RAM i ROM (z wyjątkiem relokowanego obszaru pamięci EPROM) oraz urządzeń wejścia-wyjścia
- **READY** — gotowość pamięci i urządzeń wejścia-wyjścia; służy do spowolnienia pracy procesora przy współpracy z wolno działającymi pamięciami i urządzeniami wejścia-wyjścia, może być wykorzystany w pracy krokowej
- **RESET** — sprowadzanie układów do stanu początkowego; wysoki poziom sygnału (logiczne 1) powoduje zainicjowanie odpowiednich rejestrów jednostki centralnej i innych urządzeń; w przypadku relokowanej pamięci EPROM uaktywnia się układ relokowania całej pamięci i włącza pamięć zawierającą MONITOR
- **RESET EX** — zewnętrzny sygnał RESET
- **HOLD** — żądanie odłączenia jednostki centralnej od magistrali; niski poziom sygnału sygnalizuje jednostce centralnej, że inne urządzenie (np. sterownik DMA) żąda dostępu do magistrali
- **HOLDA** — potwierdzenie odłączenia od magistrali; wysoki poziom sygnału informuje, że układy jednostki centralnej znajdują się w stanie dużej impedancji
- **IRQ7—IRQ0** — przerwania; sygnały przerwania są doprowadzane do procesora przez programowany układ przerwania INTEL 8259, służący do określania priorytetów
- **INTE** — zezwolenie na przerwanie; wysoki poziom sygnału informuje, że procesor może przyjąć i obsłużyć przerwania
- **INTA** — potwierdzenie przerwania; niski poziom sygnału umożliwia układowi przerwania (lub urządzeniu zgłaszającemu przerwanie) wprowadzenie wektora przerwania
- **02TTL** — sygnał zegarowy; przebieg prostokątny o wypełnieniu 5/9 i częstotliwości zależnej od zastosowanego kwarcu
- **ID5** — stan wewnętrznej linii danych D5; umożliwia wyszukanie fazy M1 wykonywania rozkazu, użyteczny przy krokowej pracy programu
- **STSTB** — strobowanie stanu; niski poziom sygnału informuje, że procesor udostępnia stan określający rozkaz, który zostanie wykonany.

CHARAKTERYSTYKA PAKIETÓW I ZASILACZA

W systemie występują trzy rodzaje pakietów — podstawowe, specjalizowane, sprzężeń z obiektem, a także zasilacz. Poniżej zostaną scharakteryzowane ich funkcje oraz parametry techniczne.

Pakiety podstawowe

Pakiet jednostki centralnej spełnia następujące funkcje:

- generuje sygnały adresowe i sterujące
- steruje przepływem danych
- dekodując adresy dzieli wstępnie obszar pamięci na 8 podobszarów
- wprowadza magistralę systemową w stan dużej impedancji (na żądanie układu bezpośredniego dostępu do pamięci)
- przyjmuje i zapewnia obsługę 8 sygnałów przerwania.

Pakiet składa się z procesora, sterownika systemowego, sterownika przerwania, buforów magistrali (wszystkie linie magistrali systemowej są buforowane) i dekodera typu „1z8” (trzech najbardziej znaczących bitów szyny adresowej).

Pakiet złożonego mikrokomputera zawiera:

- mikroprocesor, sterownik systemowy i zegar
- relokowalną pamięć EPROM o pojemności 2 K bajtów
- pamięć RAM o pojemności 2 K bajtów
- 8-poziomowy sterownik przerwania INTEL 8214/8212
- programowany czasomierz INTEL 8253.

Szyny adresowe, szyny danych i szyny sterujące transmisyją, a także pozostałe szyny magistrali są buforowane, co umożliwia współpracę pakietu z dużym systemem. Pakiet może współpracować także z urządzeniami bezpośredniego dostępu do pamięci (DMA). Programowany licznik, umieszczony na pakiecie, umożliwia odmierzenie czasu, realizację opóźnień oraz zliczanie impulsów z różnych źródeł sygnałów, angażując w minimalnym stopniu mikroprocesor i pamięć systemu.

Pakiet prostego mikrokomputera zawiera:

- mikroprocesor, sterownik systemowy i zegar
- 8-poziomowy sterownik przerwania
- pamięć EPROM o pojemności 2 K bajtów
- pamięć RAM o pojemności 1 K bajtów
- programowany licznik (6 X UCY 74193)

Ze względu na ograniczoną wydajność prądową układów sterujących szynami i układów pamięciowych (2 mA w stanie logicznego 0), pakiet ten może współpracować tylko z niewielką liczbą innych pakietów i jest przeznaczony do realizacji małych systemów.

Zadaniem zegara jest wysyłanie przerwania do mikroprocesora w regularnych odstępach czasu (np. co 1 s). Przy każdym przerwaniu procesor zawiesza swą dotychczasową działalność i przechodzi do programu obsługi przerwania, uaktualniając zawartość określonych komórek pamięci i podejmując dalsze przewidziane akcje, a następnie powraca do przerwanego programu. Okres wysyłania przerwania może wynosić od kilku μ s do kilku sekund.

Pakiet pamięci RAM zawiera układy typu 2102 i umożliwia pamiętanie 4 K bajtów informacji. Wszystkie sygnały są buforowane układami INTEL 8216 lub układami TTL serii 74. Na pakiecie znajduje się także dekodery adresowy.

Pakiet pamięci EPROM zawiera układy typu 2708 i umożliwia zapamiętanie 8 K bajtów informacji. Ponadto, pakiet zawiera dekodery adresowy oraz bufor magistrali.

Pakiet współpracy z urządzeniami zewnętrznymi umożliwia:

- współpracę przez sprzęg standardowy V24 lub — w pętli prądowej (z drukarką DZM-180 KSR lub z monitorem ekranowym)
- współpracę z czytnikiem taśmy perforowanej CT 2200 lub perforatorem DT 105S.

Pakiet równoległych wejść-wyjść cyfrowych (8-bitowych) umożliwia wymianę danych z trzema niezależnymi urządzeniami, z możliwością ustalenia kierunku transmisji w każdym kanale. Sygnały wejścia-wyjścia są buforowane dwukierunkowymi wzmacniaczami INTEL 8216/8226, co umożliwia bezpośrednio sterowanie układami wykonawczymi. Głównym elementem pakietu jest programowany, równoległy układ wejścia-wyjścia INTEL 8255. Na pakiecie przewidziano miejsce do realizacji układów dodatkowych, o specjalnych wymaganiach.

Pakiet relokowalnej pamięci stałej zrealizowany z układów typu 2708 umożliwia pamiętanie 8 K bajtów informa-

cji dostępnej pod adresami od F000 do FFFF. Najistotniejszą cechą pakietu jest dekodowanie adresów od F000 do F3FF na dwa obszary adresowe. Po sygnale RESET jest to obszar 000-3FF, natomiast po dokonaniu relokacji — obszar F000-F3FF. Relokacja umożliwia umieszczenie MONITORA na końcu obszaru adresowego i dowolne dysponowanie pozostałym obszarem. Inicjowanie systemu po sygnale RESET odbywa się zgodnie z programem MONITORA, który do chwili powrotu do naturalnego obszaru adresowego występuje pod adresami 000-3FF. Pamięć RAM jest w tym czasie zablokowana.

Pakiet pracy krokowej umożliwia krokowe wykonywanie programów użytkowych oraz układową realizację punktu wstrzymania. Układ pracy krokowej i układ wstrzymania działają na zasadzie przerwania zgłoszonego przez linię IRQ1 układu INTEL 8259. Sygnał wstrzymania zostaje wytworzony w komparatorze cyfrowym, a układ pracy krokowej sygnalizuje przejście przez zero z licznika, zliczającego odczyty pamięci.

Pakiety specjalizowane

Pakiet programatora pamięci EPROM służy do programowania układów typu 2708, 2704 i 2716. Został zaprojektowany w oparciu o programowany układ INTEL 8255 oraz układy dopasowania pamięci EPROM. Z pakietem związane jest oprogramowanie, umożliwiające programowanie pamięci oraz weryfikację zaprogramowanych obszarów. Pakiet zajmuje cztery miejsca w obszarze adresowym wejścia-wyjścia.

Pakiet obsługi klawiatury i wyświetlaczy umożliwia obsługę 66 przycisków klawiatury (w tym — przycisków SHIFT i CONTROL) oraz 16 wyświetlaczy siedmiosegmentowych ze wspólną anodą. Zmiany stanu mogą być sygnalizowane przerwami lub odczytywane przez mikroprocesor przeglądający stan układu INTEL 8279.

Pakiet sterowania pamięcią kasetową PK-1 umożliwia obsługę dwóch jednostek tej pamięci. Zapewnia zamianę równoległej informacji z procesora na informację szeregową (w kodzie dwufazowym) dla pamięci kasetowej, a także zamianę informacji szeregową na równoległą. Obsługa pamięci PK-1 może następować wskutek przerwania z układu transmisji i licznika lub — programowo — po badaniu stanu układów.

Pakiet sterowania jednostką dysków elastycznych PL45xD umożliwia dołączenie do systemu jednej lub dwóch takich jednostek. Transmisję można realizować przy użyciu kanału DMA lub programowo — przez wykorzystanie procesora. Przy współpracy z kanałem DMA procesor inicjuje transmisję, wysyłając do sterownika współrzędne pliku na dysku elastycznym oraz adres obszaru pamięci, do którego należy przesłać dane. Zakończenie transmisji jest sygnalizowane przerwaniem lub ustawieniem bitu w słowie stanu sterownika. Przy pracy bez wykorzystania bezpośredniego dostępu do pamięci procesor musi dodatkowo przesyłać w odpowiednio krótkim czasie kolejne bajty danych. Możliwa jest realizacja następujących funkcji: przesuwanie głowicy zapisu i odczytu, odczyt i zapis danych, anulowanie danych, weryfikacja danych, formowanie zapisu w standardzie IBM 3740, z możliwością zapisu sektorów na ścieżce w dowolnej kolejności.

Pakiety sprzężeń z obiektem

Pakiet wejść dwustanowych 24 V umożliwia wprowadzanie sygnałów dwuwartościowych (w dwu grupach po osiem linii). Zajmuje dwa miejsca w obszarze adresowym wejścia-wyjścia. Podstawowe właściwości pakietu są następujące:

- galwaniczne rozdzielanie wejść
- sygnały wejściowe 20 mA, 24 V
- częstotliwość graniczna sygnałów wejściowych 1 kHz
- zabezpieczenie przeciw przepięciowe (filtr dolnoprzepustowy)
- sygnalizacja stanu wejścia diodą świecącą.

Pakiet wyjść dwustanowych 24 V umożliwia przesyłanie sygnałów sterujących do urządzeń wykonawczych, po ośmiu równoległych liniach (jedno miejsce w obszarze adresowym wejścia-wyjścia). Podstawowe właściwości pakietu są następujące:

- galwaniczne rozdzielanie wejść
- napięcie zasilania wyjść 24 V, prąd obciążenia do 0,5 A
- częstotliwość graniczna 30 kHz
- wysoki poziom sygnału, sygnalizowany przy użyciu diody świecącej.

Pakiet sterowania silnikiem prądu stałego jest przeznaczony do sterowania i regulacji prędkości obrotowej silnika prądu stałego małej mocy. Regulacja polega na impulsowymysterowaniu obwodu twornika z modulacją wypełnienia (przy stałej szerokości impulsu). Pętla regulacji jest zamknięta przez system mikroprocesorowy. Podstawowe właściwości pakietu są następujące:

- moc sterowanych silników do 120 W
- zakres regulacji prędkości obrotowej 1:40
- dokładność regulacji 1% (powyżej 10% obrotów maksymalnych), 2% (poniżej 10% obrotów maksymalnych)
- rozdzielanie galwaniczneysterowania obwodu silnika
- zmiana kierunku obrotów za pomocą przekaźnikaysterowanego z rozdzielaniem galwanicznym
- pomiar prędkości obrotowej przy użyciu przetwornika obrotowo-impulsowego (300 imp./obrót)
- możliwość stosowania hamowania dynamicznego przez zwieranie obwodu twornika.

Pakiet sterowania silnikiem skokowym umożliwia programowo-sprzętowo sterowanie silnikiem skokowym czterofazowym z kształtowaniem charakterystyki rozruchu i hamowania przy niewielkim zaangażowaniu samego procesora. Pakiet zawiera następujące układy: generator taktujący, licznik skoków dla rozruchu i hamowania oraz pracy synchronicznej, programowany dzielnik częstotliwości, komutator zrealizowany w oparciu o pamięć UCY 74186 (cztery algorytmy), układy rozdzielania galwanicznego, układy sterowania i dekodera.

Pakiet sterowania trzema silnikami skokowymi umożliwia programowy rozruch, hamowanie i kontrolę przesunięć silników. Zawiera następujące układy: rejestr umożliwiający pamiętanieysterowania poszczególnych faz silników, układy rozdzielania galwanicznego, dekodera adresowy. Pakiet ma możliwość współpracy z różnymi wzmacniaczami mocy: dwunapięciowym, wzmacniaczem z impulsowym lub oporowym ograniczeniem prądu.

Pakiety sprzężenia z przetwornikami cyfrowo-analogowymi są przeznaczone do realizacji sprzężenia jednostki centralnej z dwoma przetwornikami 8-bitowymi typu DAC44 (dziesiętnymi — DAC44D) lub 10-bitowymi typu DAC45 (12-bitowymi dziesiętnymi — DAC46D). Czas konwersji jest krótszy od 10 μ s, a zakres napięcia wyjściowego wynosi 0-5 V. Pakiety zajmują dwa miejsca w obszarze adresowym wejścia-wyjścia (cztery miejsca w przypadku przetworników DAC45 i DAC46).

Pakiet jednokanałowego przetwornika analogowo-cyfrowego umożliwia sprzężenie mikroprocesora z przetwornikiem 8-, 10- lub 12-bitowym. Możliwe jest dołączenie przetworników hybrydowych (np. ADC8S, ADC10S), jak również przetworników monolitycznych typu AD751KN firmy ANALOG DEVICES. Zakres wejściowego sygnału napięciowego jest równy 0-5 V. Koniec przetwarzania może być sygnalizowany przez zgłoszenie przerwania, programowo badanie stanu linii „koniec przetwarzania” lub odczytanie stanu wyjść przetwornika, po upływie czasu gwarantującego ustalenie wyniku przetwarzania.

Pakiet sześciokanałowego przetwarzania analogowo-cyfrowego umożliwia przetwarzanie sześciu sygnałów analogowych z zakresu 0-5 V na słowo 10-bitowe. Sterowanie pakietu jest programowe, inicjowane przez użytkownika. Podanie adresu w kodzie „1 z 6” uruchamia cykl przetwarzania sygnału z określonej linii. Koniec przetwarzania jest sygnalizowany przerwaniem. W pakiecie zastosowano przetworniki hybrydowe.

Zasilacz systemowy

Zasilacz jest przeznaczony do realizacji zestawów wykonywanych pakietów systemu MIKROSTER. Dostarcza następujących napięć zasilających:

- nie buforowanych + 5 V (10 A), - 15 V (0,25 A), + 15 V (0,25 A)
- buforowanych baterią akumulatorów + 5 V (2,5 A), - 5V (0,6 A), + 12 V (1 A).

Przykładowe parametry techniczne zasilacza dla wyjścia +5V/10A są następujące:

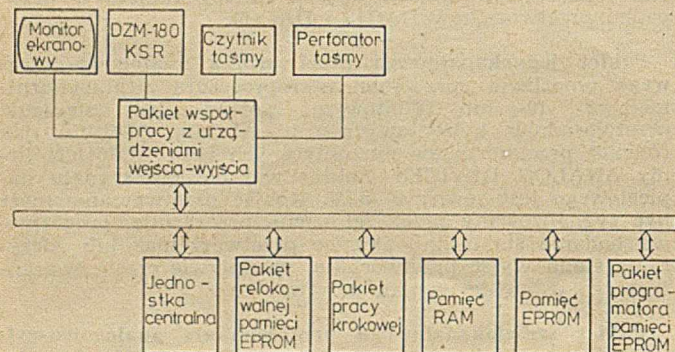
- współczynnik stabilizacji napięcia wyjściowego w zależności od zmian napięcia sieci w zakresie 187—242 V, 0,1%
- współczynnik stabilizacji napięcia wyjściowego w zależności od zmian obciążenia w zakresie 0—100%, 0,1%
- składowa zmienna napięcia wyjściowego (tętnienia i szumy), 25 mV — wartość szczytowa, 5mV — wartość skuteczna
- impedancja wyjściowa przy częstotliwości 100 kHz, 0,1 Ω
- amplituda stanu przejściowego dla spadku 50%, 4%
- czas powrotu do stanu ustalonego dla spadku 50%, 50 μ s
- zabezpieczenie przed przeciążeniem
- zabezpieczenie przepięciowe
- tłumienie zakłóceń w zakresie do 150 MHz 50 dB
- zakres temperatury pracy 0—70°C.

ZESTAW PROJEKTOWO-URUCHOMIENIOWY OPARTY NA MODULACH SYSTEMU MIKROSTER

W skład mikroprocesorowego systemu projektowo-uruchomieniowego wchodzi następujące pakiety podstawowe (rys. 2):

- pakiet procesora INTEL 8080 wyposażony w zegar typu 8228; sterownik systemowy typu 8228, układ przerwań typu 8259 oraz bufor szyny adresowej i szyny danych
- pakiet pamięci RAM o pojemności od 1 K do 4 K bajtów, zbudowany z układów typu 2102
- pakiet pamięci EPROM o maksymalnej pojemności 8 K bajtów, zawierający układy typu 2708
- pakiet relokowalnej pamięci stałej, zawierający osiem układów typu 2708 w obszarze adresowym F000—FFFF, przy czym adresy od F000 do F3FF są dekodowane na dwa obszary
- pakiet pracy krokowej i przerwań programowych, umożliwiający krokowe wykonywanie programów użytkowych oraz wprowadzanie punktów wstrzymania
- pakiet transmisji szeregowo-równoległej, umożliwiający współpracę z urządzeniami wejścia-wyjścia przez sprzęg szeregowy V 24 lub w pętli prądowej 20 mA oraz z czytnikiem i perforatorem taśmy papierowej
- pakiet programatora pamięci typu 2704, 2708, 2716.

System ma w pełni równoległą magistralę i jest zorganizowany według zasady minimalnego obciążenia magistrali przez wejścia pakietów. Dzięki temu, jak również dzięki możliwości dowolnego rozmieszczenia pakietów w obszarze adresowym, część pakietów może być zwielokrotniona aż do pełnego wykorzystania tego obszaru.



Rys. 2. Konfiguracja systemu wspomagającego

OPROGRAMOWANIE

Oprogramowanie systemu MIKROSTER składa się z następujących programów: MONITOR, ASSEMBLER, TEXT EDITOR, system operacyjny MICROS.

Program MONITOR

Program ten zajmuje 4 K bajtów pamięci ROM w obszarze F000—FFFF oraz wykorzystuje 256 bajtów pamięci RAM. Poszczególne dyrektywy programu spełniają następujące funkcje:

- wprowadzanie bloków danych z konsoli operatorskiej do pamięci RAM
- wyprowadzanie bloków danych z pamięci na konsolę lub wyprowadzanie zawartości poszczególnych komórek z możliwością ich modyfikacji
- przemieszczanie bloków danych w pamięci
- wyprowadzenie zawartości rejestrów wewnętrznych procesora na konsolę z możliwością modyfikacji
- wyprowadzanie bloków danych z pamięci na taśmę perforowaną
- wprowadzanie bloków danych z taśmy perforowanej w dowolne miejsce pamięci RAM
- umieszczanie przerwań programowego w programie użytkowym
- wykonanie programu użytkowego
- krokowe wykonywanie programu
- przepisywanie bloków danych z pamięci RAM do pamięci EPROM i odwrotnie
- obsługa pamięci kasetowej PK-1
- wywoływanie programów TEXT EDITOR i ASSEMBLER.

MONITOR ściśle odpowiada opracowanym pakietom. Do cech wyróżniających go spośród innych programów tego rodzaju należy zaliczyć układową realizację następujących funkcji:

- inicjacji programu umieszczonego w górnym obszarze adresowym
 - wykonywania przerwań programowych
 - krokowego wykonywania programu.
- Umieszczenie programu MONITOR w górnym obszarze pamięci ROM umożliwia udostępnienie użytkownikowi zew-
rowej strony pamięci.

Programy ASSEMBLER i TEXT EDITOR

Dwuprzebiegowy ASSEMBLER zajmuje 8 K bajtów pamięci ROM i wykorzystuje ponad 2 K bajtów pamięci RAM, w której przechowuje pola robocze i tablicę symboli. Program źródłowy dla ASSEMBLERA jest przechowywany na taśmie dziurkowanej. Wynikiem tłumaczenia jest również taśma dziurkowana zawierająca program wynikowy oraz kod wydruku, na który składa się program źródłowy, program wynikowy i opis wykrytych błędów. ASSEMBLER akceptuje program źródłowy napisany zgodnie z zasadami przedstawionymi w podręczniku 8080 ASSEMBLY LANGUAGE PROGRAMMING MANUAL (firmy INTEL). Program źródłowy może zawierać do pięciuset symboli.

Program TEXT EDITOR zajmuje 4 K bajtów pamięci ROM. W pamięci RAM znajdują się zmienne robocze edytora oraz bufor tekstu. Tekst może być wprowadzany do bufora z konsoli operatorskiej lub z taśmy dziurkowanej. Po zakończeniu redagowania tworzona jest taśma wyjściowa.

TEXT EDITOR pracuje w trybie konserwacyjnym, umożliwiając realizację następujących funkcji:

- wprowadzanie tekstu z konsoli operatorskiej i czytnika taśmy
- wyprowadzanie tekstu na taśmę dziurkowaną i konsolę
- manipulowanie wskaźnikiem bufora
- usuwanie i modyfikowanie tekstu
- wyszukiwanie zadanego ciągu znaków i zastępowanie go innym.

ASSEMBLER i TEXT EDITOR wymagają podczas pracy obecności programu MONITOR.

System operacyjny MICROS

Dyskowy system wspomaganie projektanta MICROS ułatwia i przyspiesza proces projektowania oprogramowania urządzeń, opartych o układy mikroprocesorowe. Jego podstawową funkcją jest zarządzanie plikami użytkownika na poziomie logicznym (tworzenie, usuwanie, zmiana nazw i atrybutów, przepisywanie na dowolne urządzenie zewnętrzne itp.), co uwalnia projektanta (użytkownika) od wykonywania bardzo kłopotliwych i czasochłonnnych operacji na poziomie sprzętowym.

Istotną zaletą systemu jest łatwość i efektywność współpracy z urządzeniami zewnętrznymi oraz możliwość korzystania z kompilatorów (interpreterów) języków wysokiego poziomu (BASIC, FORTH, PL/M-80), MACROASSEMBLERA 80/85 i bogatych bibliotek systemowych (w tym biblioteki arytmetyki zmiennie-przecinkowej). Stosunkowo duże

możliwości redagowania programów, konstruowania modułowego oprogramowania (przemieszczalnego i bezwzględnie), łączenia poszczególnych modułów (także napisanych w różnych językach) i dynamicznego umieszczania ich w pamięci, znacznie przyspieszają proces pisania i uruchamianie oprogramowania urządzeń mikroprocesorowych.

Realizacja funkcji systemu odbywa się przez wydawanie poleceń wywołujących odpowiednie programy systemowe. Zasada taka umożliwi rozszerzenie lub zmianę zbioru poleceń przez wymianę lub dołączenie nowych programów. System MICROS umożliwia również przenoszenie oprogramowania z systemów ISIS I i ISIS II firmy INTEL.

W następnej wersji systemu planuje się dołączenie pamięci kasetowej jako pamięci masowej (archiwum programów, biblioteka itp.) lub jako urządzenia systemowego zastępującego stanowisko dyskowe (do zastosowań nie wymagających dużej szybkości). Ponadto planowane jest znaczne rozszerzenie możliwości współpracy z urządzeniami zewnętrznymi na zasadzie przydziału dynamicznego, a także — rozszerzenie samego systemu operacyjnego w celu włączenia wielu istotnych funkcji wspomagających proces uruchamiania (obecnie są one włączone do monitora współpracującego z systemem operacyjnym). Planuje się również dołączenie kompilatorów innych języków (FORTRAN, PASCAL).

Typowa konfiguracja sprzętu dla systemu operacyjnego MICROS obejmuje:

- pamięć RAM, o pojemności od 32 K do 62 K bajtów, w tym 12 K bajtów zajętych na stałe przez program rezydujący

ANDRZEJ WOŹNIAK, MAREK PAWŁOWSKI,
JANUSZ RZESZUT, ANDRZEJ SKORUPSKI, JANUSZ SOSNOWSKI,
JACEK STOCHŁAK
Instytut Informatyki
Politechnika Warszawska

- pamięć ROM o pojemności od 2 K do 32 K bajtów, w tym 2 K bajtów zajęte przez program MONITOR

- następujące urządzenia zewnętrzne

— pamięci na dyskach elastycznych SP45DE, SP60M lub PLx45D

— konsola operatorska DZM-180 KSR lub MERA-7952

— czytnik taśmy dziurkowanej CT-2100 lub CT-2200

— dziurkarka taśmy papierowej DT-105

— drukarka mozaikowa DZM-180

— pamięć kasetowa PK1.

* * *

Obecnie opracowywane są m.in. pakiety transmisji szeregowej według normy RS232, nowe pakiety wielokanałowego przetwarzania analogowo-cyfrowego i cyfrowo-analogowego, pakiet wejść-wyjść dwustanowych, pakiet sterowania wyjść prądowych z układami tyrystorowymi. Przewiduje się podjęcie w bieżącym roku produkcji systemu w Naukowo-Produkcyjnym Centrum Półprzewodników, w oparciu o elementy opracowane i produkowane w NPCP oraz elementy produkcji radzieckiej. Dotychczas w Przemysłowym Instytucie Elektroniki wykonano kilka systemów wspomaganiania oraz opracowano szereg sterowników urządzeń technologicznych, m.in. próbnik PSC-80 do pomiaru rezystywności warstw epitaksjalnych, sterownik urządzeń do epitaksji i in.

Modułowy System Mikroprocesorowy — MSM

Technika cyfrowa w kraju przekracza obecnie kolejny próg rozwoju, następuje przejście od układów MSI do układów LSI, w szczególności — do układów mikroprocesorowych. Jest możliwe prawie natychmiastowe zastosowanie techniki mikroprocesorowej w wielu dziedzinach nauki i techniki, jeśli zostaną rozpowszechnione doświadczenia i wyniki prac różnych ośrodków.

W celu ułatwienia projektowania i uruchamiania układów i systemów mikroprocesorowych, w Instytucie Infor-

matyki PW opracowano Modułowy System Mikroprocesorowy — MSM. System ten znalazł zastosowanie przede wszystkim w procesie dydaktycznym, gdyż umożliwia zaprojektowanie i uruchomienie złożonych układów mikroprocesorowych w bardzo krótkim czasie. W oparciu o system MSM, wprowadzony do produkcji na Wydziale Doświadczalnym Instytutu Informatyki PW, zaprojektowano już wiele urządzeń i systemów specjalizowanych. Powstają również pierwsze opracowania i projekty wykonywane poza Instytutem.

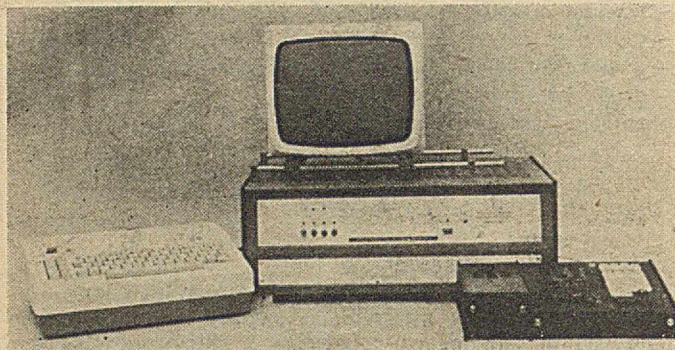


Mgr inż. ANDRZEJ WOŹNIAK ukończył w 1977 r. studia na Wydziale Elektroniki Politechniki Warszawskiej. Pracuje w Instytucie Informatyki PW. Zajmuje się projektowaniem i budową systemów i urządzeń cyfrowych, głównie mikroprogramowanych i mikroprocesorowych.

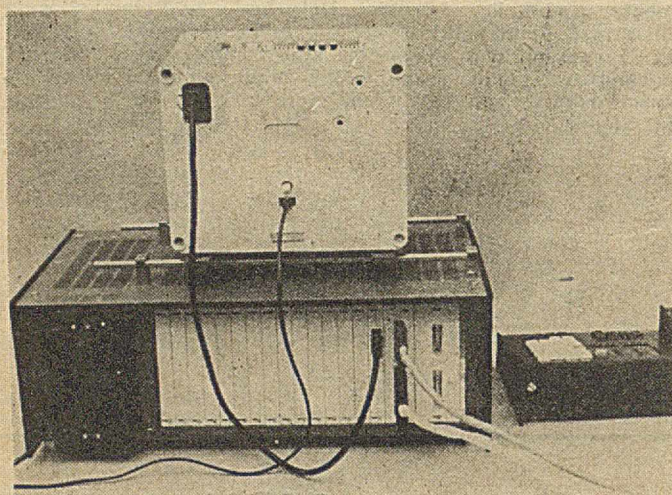


Mgr inż. MAREK PAWŁOWSKI ukończył w 1977 r. studia na Wydziale Elektroniki Politechniki Warszawskiej (specjalizacja — informatyka). Od 1977 r. pracuje w Instytucie Informatyki PW. Zajmuje się zagadnieniami związanymi z mikroprogramowaniem i układami mikroprocesorowymi.

Modułowy System Mikroprocesorowy — MSM [5, 6] opracowano w formie umożliwiającej zestawienie szerokiej klasy systemów mikroprocesorowych w różnych konfiguracjach (fot. 1, 2).



Fot. 1. Modułowy System Mikroprocesorowy — widok z przodu



Fot. 2. Modułowy System Mikroprocesorowy — widok z tyłu

OGÓLNA KONCEPCJA SYSTEMU

Podstawą systemu jest kasetka wyposażona w standardową magistralę MSM-BUS, która łączy elektrycznie pakiety (moduły) między sobą. Istnieje możliwość wyprowadzenia magistrali poza kasetkę w celu połączenia kilku kaset razem lub dołączenia dodatkowej aparatury. Kasetki są wyposażone w autonomiczne (oddzielne) zasilacze o mocy 250 VA lub 500 VA.

Moduł systemu stanowi najczęściej pojedynczy pakiet (o wymiarach 140 × 150 mm), a w przypadkach bardziej złożonych — kilka pakietów dołączonych do magistrali. Wiele modułów umożliwia dokonywanie wyboru realizowa-

nych funkcji — przez zaprogramowanie fizyczne lub logiczne. Programowanie fizyczne polega na wykonaniu lub usunięciu pewnych połączeń wewnątrz modułu, a programowanie logiczne — na przesłaniu odpowiednich informacji do modułu. W sposób fizyczny programowany jest np. obszar adresowy modułów pamięci i wejścia-wyjścia, a logicznie programowany może być np. sposób transmisji w module sprzęgającym.

W czasie opracowywania systemu MSM przeanalizowano szereg znanych rozwiązań systemów mikroprocesorowych [4] i zdecydowano się na przyjęcie własnego standardu magistrali. Wynikło to z zakresu zastosowań oraz możliwości realizacji systemu w Instytucie Informatyki. System MSM łączy w sobie cechy standardu poziomu układowego (ang. *component level*) i systemowego (ang. *system level*) [1]. Cechy poziomu układowego umożliwiają użytkownikowi konstruowanie własnych modułów dołączanych do magistrali systemowej. Cechy poziomu systemowego ułatwiają szybkie zestawienie systemu mikroprocesorowego o dużych możliwościach przetwarzania, w różnych konfiguracjach, przy zastosowaniu standardowych modułów.

W porównaniu ze znanymi rozwiązaniami, magistrala MSM jest najbliższa magistrali STD-BUS opracowanej w firmie PROLOG [2, 3]. Podstawowe różnice wynikają z przyjętego standardu mechanicznego oraz rozbudowania szyny sterującej, co zapewnia łatwą i bezpośrednią realizację systemu w oparciu o różne rodziny mikroprocesorowe, jak np. INTEL 8080, INTEL 8085, INTEL 8048, Z80, MOTOROLA 6800. Ze względu na rozpowszechnienie — zwłaszcza w krajach RWPG — mikroprocesorów firmy INTEL, system MSM skierowano na tę właśnie rodzinę mikroprocesorów.

Zestawy są uzupełniane dodatkowym sprzętem uruchomieniowym, jak: klawiatura, monitor magistrali, analizator stanów logicznych, emulator systemowy itp. System jest wyposażony w różnorodne oprogramowanie, od prostych monitorów systemowych, do standardowych systemów operacyjnych, jak CP/M i ISIS.

MODUŁY SYSTEMU MSM

W skład systemu MSM wchodzi szereg modułów standardowych umożliwiających tworzenie właściwych zestawów dla konkretnych zastosowań. Większość modułów zrealizowano w postaci pojedynczego pakietu. Dopuszcza się również stosowanie modułów wielopakietowych.

Biorąc pod uwagę właściwości funkcjonalne można wyróżnić następujące grupy modułów:

- moduły procesorów
- moduły pamięci
- moduły wejścia-wyjścia
- moduły sterowników urządzeń zewnętrznych.

Poniżej scharakteryzowano podstawowe moduły systemu MSM.

Moduły procesorów

Opracowano kilka modułów uniwersalnych procesorów przeznaczonych do wykorzystania jako jednostki centralne systemu MSM. Są one przystosowane do współpracy z urządzeniami ułatwiającymi uruchamianie i testowanie systemu (np. pulpitu operatorskiego, monitora magistrali).

Moduł CPU#1 jest jednopakietowym modułem procesora INTEL 8080A ze sterownikiem systemowym typu 8228/8238 i układem zegarowym — 8224 (rys. 1). Zawiera układ przezwania typu 8214 umożliwiający obsługę ośmiu linii prze-

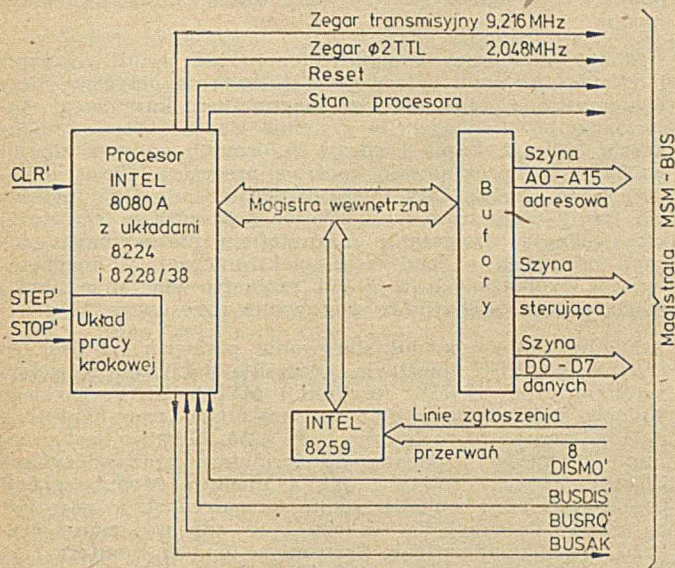


Mgr inż. JANUSZ RZESZUT ukończył w 1978 r. studia na Wydziale Elektroniki Politechniki Warszawskiej. Pracuje w Instytucie Informatyki PW. Zajmuje się projektowaniem systemów mikroprocesorowych i urządzeniami techniki graficznej.



Dr inż. ANDRZEJ SKORUPSKI ukończył w 1966 r. studia na Wydziale Elektroniki Politechniki Warszawskiej, a w 1975 r. uzyskał tytuł doktora nauk technicznych (praca doktorska z dziedziny teorii układów logicznych). Zajmuje się projektowaniem urządzeń i systemów cyfrowych i cyfrowo-analogowych (z uwzględnieniem techniki mikroprocesorowej). Od 1978 r. pełni funkcję zastępcy dyrektora Instytutu Informatyki PW.

rwai, a ponadto — układ automatycznego zerowania po włączeniu zasilania i układ pracy krokowej. Moduł generuje sygnał zegarowy o częstotliwości 9,216 MHz. Buforowane szyny danych, adresów i sterowania umożliwiają wystawianie ok. 10 typowych modułów MSM.



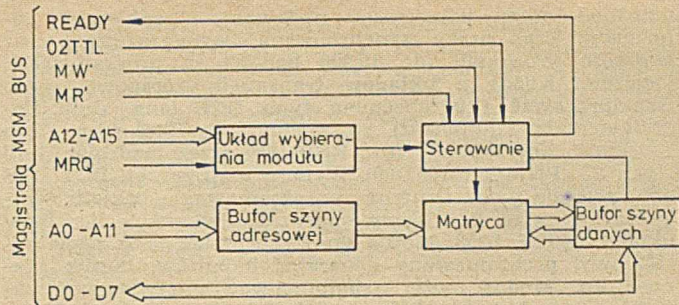
Rys. 1. Schemat blokowy modułu procesora CPU 2

CPU#2 jest jednopakietowym modułem procesora INTEL 8080A ze sterownikiem systemowym typu 8228/8238 i układem zegarowym — 8224. Zawiera także programowany układ przerwań typu 8259 umożliwiający obsługę ośmiu linii przerwań. Możliwe jest dołączenie zewnętrznego układu przerwań, np. modułu kaskadowo połączonych układów typu 8259 do obsługi 64 linii przerwań. Podobnie jak moduł CPU#1, CPU#2 zawiera układ automatycznego zerowania po włączeniu zasilania, układ pracy krokowej oraz generuje sygnał zegarowy o częstotliwości 9,216 MHz. Buforowane szyny danych, adresów i sterowania umożliwiają dołączenie ok. 20 typowych modułów MSM.

Moduł CPU#3 jest jednopakietowym modułem procesora ZILOG Z-80. Ma on analogiczne możliwości jak CPU#2, rozszerzone o dodatkowe rozkazy (dzięki zastosowaniu procesora Z-80), powielony zestaw rejestrów procesora oraz niemaskowalne przerwanie NMI. Dodatkowo procesor Z-80 generuje na magistralę MSM cykle odświeżania pamięci dynamicznej RAM.

Moduły pamięci

W systemie MSM dostępne są moduły pamięciowe czterech typów: statyczna pamięć RAM, dynamiczna pamięć RAM, pamięć ROM i pamięć przechowująca program samoadujący (ang. *bootstrap*). Z wyjątkiem tego ostatniego modułu pozostałe są wyposażone w programowany układ dekodera adresu modułu. Programowanie fizyczne dekodera (przez wykonanie odpowiednich połączeń lutowanych) umożliwia określenie położenia reprezentowanego przez moduł bloku pamięci w obszarze pamięci systemu. Schemat przykładowego modułu pamięci przedstawiono na rys. 2.



Rys. 2. Schemat blokowy modułu pamięci statycznej RAM

Jednopakietowy moduł pamięci statycznej RAM o pojemności 4096 słów 8-bitowych składa się z 32 układów scalonych równoważnych układowi INTEL 2102. Moduł jest wybierany przez zdekodowanie adresu, zawiera również układ generacji sygnału READY umożliwiający wprowadzenie dodatkowych cykli oczekiwania (Tw) dla wolno działających odmian układu scalonego typu 2102.

Jednopakietowy moduł pamięci stałej zawiera pamięć EPROM o pojemności od 1024 do 65 536 słów 8-bitowych w zależności od typu użytej pamięci. Jest on wyposażony w osiem podstawek 28-krotnych przeznaczonych dla układów scalonych INTEL 2708, 2716, 2732, 2764 lub ich odpowiedników innych firm. Moduł jest wybierany przez zdekodowanie adresu pod warunkiem, że wskazana adresem podstawa zawiera układ scalony pamięci. Umożliwia współpracę z wolno działającymi pamięciami przez generację odpowiedniego sygnału READY.

Moduł pamięci stałej o pojemności 2048 lub 4096 słów 8-bitowych (w zależności od typu pamięci — 2x2708 lub 2x2716), oznaczony literą B zajmuje obszar od adresu 0000H i jest przeznaczony do przechowywania programu samoadującego. W stanie aktywnym następuje odczytanie pamięci RAM o adresach pokrywających się z adresami modułu B. Uaktywnienie modułu następuje pod wpływem sygnału RESET (np. po włączeniu zasilania) lub programowo.

Moduły wejścia-wyjścia

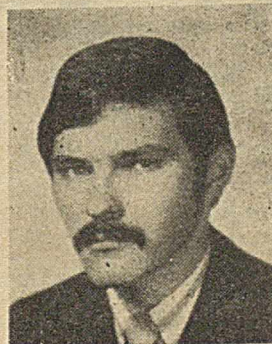
W celu ułatwienia dołączenia urządzeń wejścia-wyjścia do systemu opracowano uniwersalne moduły z równoległym i szeregowym łączem zewnętrznym oraz moduł bezpośredniego dostępu do pamięci.

Jednopakietowy moduł PPI umożliwia dołączenie do systemu MSM dwóch równoległych urządzeń zewnętrznych — nadawczego i odbiorczego. Istnieje możliwość programowania protokołów transmisji (oddzielnie — nadawczego i odbiorczego) w celu bezpośredniej współpracy modułu PPI z urządzeniami krajowymi (jak: czytniki i perforatory, taśmy papierowej — CT-2000 i DT-15, drukarki i klawiatury alfanumeryczne — DZM-180 i KSR-185. Możliwe jest dołączenie drukarki DZM-180 zamiast perforatora DT-105S bez jakichkolwiek zmian w pakiecie. Sterowanie transmisją zrealizowano przy użyciu układu INTEL 8255. Współpraca z urządzeniami zewnętrznymi jest możliwa pod kontrolą programu (bez przerwań i z przerwaniami) lub pod kontrolą układu DMA.

Jednopakietowy moduł V24/CTC zawiera dwa uniwersalne układy transmisji szeregowej z autonomicznymi zegarami oraz uniwersalny układ licznikowy (rys. 3). Transmisję szeregową zrealizowano przy użyciu układu INTEL 8251,

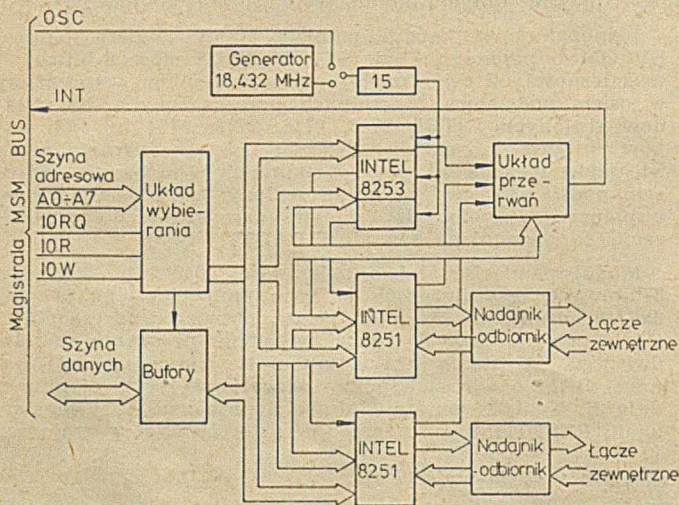


Dr inż. JANUSZ SOSNOWSKI ukończył w 1968 r. studia na Wydziale Elektroniki Politechniki Warszawskiej, a w 1976 r. uzyskał tytuł doktora nauk technicznych (praca doktorska z dziedziny teorii układów logicznych). Od 1968 r. pracuje w Instytucie Informatyki PW. Zajmuje się projektowaniem systemów, urządzeń cyfrowych i cyfrowo-analogowych (z uwzględnieniem techniki mikroprocesorowej).



Dr inż. JACEK STOCHLAK ukończył w 1974 r. Wydział Elektroniki Politechniki Warszawskiej (specjalizacja — informatyka), a w roku 1982 uzyskał tytuł doktora nauk technicznych. Od roku 1974 pracuje w Instytucie Informatyki PW obecnie na stanowisku adiunkta. Zajmuje się modelowaniem systemów cyfrowych oraz opracowywaniem i realizacją oprogramowania podstawowego dla systemów mini- i mikrokomputerowych.

umożliwiającego jednocześnie nadawanie i odbieranie informacji. Tryb pracy (synchroniczna, asynchroniczna) oraz szybkość transmisji (do 19200 bodów) są programowane logicznie. Każdy z układów transmisji szeregowej może współpracować z urządzeniem typu DCE (ang. *Data Communication Equipment*) np. modem lub DTE (ang. *Data Terminal Equipment*), np. KSR-180 przez podzbiór linii styku S2 (PN75/T-05-052-00). Wyboru konfiguracji DCE/DTE dokonuje się przez fizyczne zaprogramowanie pakietu. Uniwersalny układ licznikowy oparto na jednym z liczników układu INTEL 8253. Można go wykorzystać jako zegar systemowy, programowany generator impulsów, licznik zdarzeń itp. Moduł może współpracować z procesorem zarówno w trybie przeglądania, jak i z przerwami. Przerwaniami sygnalizowane jest odebranie lub wysłanie znaku przez układ transmisji szeregowej, a także — zliczenie zadanej liczby impulsów przez układ licznika. Rejestry modułu zajmują 16 kolejnych miejsc w obszarze adresowym. Ustalenie adresu modułu odbywa się przez fizyczne zaprogramowanie dekodera adresu.



Rys. 3. Schemat blokowy modułu transmisji szeregowej i zegara czasu rzeczywistego

W jednopakietowym module czterokanałowej jednostki bezpośredniego dostępu do pamięci DMA-4 wykorzystano scalony układ DMA typu INTEL 8257. Moduł zgłasza przerwanie po zakończeniu transmisji — w którymkolwiek z kanałów. Buforowane szyny: adresowa, danych i sterowania — umożliwiają wystereowanie ok. 20 standardowych modułów MSM.

Moduły sterowników urządzeń zewnętrznych

Opracowano kilka specjalizowanych modułów umożliwiających bezpośrednie dołączenie wybranych krajowych urządzeń zewnętrznych do magistrali systemu MSM.

Moduł MCC umożliwia współpracę z czterema jednostkami pamięci kasetowej typu PK1/1, PK1/2 lub PK1/3. Odmiana jednopakietowa zapewnia możliwość współpracy z pamięciami kasetowymi PK1 w trybie bezpośredniego, ciągłego nadzoru procesora nad pracą jednostki. Transmisja informacji może odbywać się pod kontrolą procesora lub też może być prowadzona przez układ bezpośredniego dostępu do pamięci — DMA. Moduł MCC zapewnia niezależną rezerwację dołączonych jednostek PK, sterowanie ruchem taśmy, odczytywanie stanu dołączonej logicznie pamięci i zliczanie bloków podczas szybkiego ruchu taśmy. Szeregowy zapis i odczyt informacji odbywa się z wykorzystaniem układu INTEL 8251 wyposażonego dodatkowo w układy kodera i dekodera kodu PE (ang. *Phase Encoding*, kodowanie fazowe).

W dwupakietowej odmianie modułu przewidziano użycie indywidualnie maskowanych przerw znacznie ułatwiających obsługę jednostek PK, z zachowaniem wszystkich właściwości odmiany jednopakietowej. Moduł dwupakie-

towy umożliwi ponadto autonomiczne wykonanie przewijania z sygnalizacją końca operacji — przy współpracy z PK1/3. Moduł zajmuje osiem miejsc w obszarze adresowym, a ustalenie adresu odbywa się przez fizyczne zaprogramowanie dekodera. Fizycznie programowany jest również numer kanału układu bezpośredniego dostępu do pamięci oraz numer (poziom) przerwania zgłoszonego przez moduł.

Moduł współpracy z magnetofonem MT umożliwia zapis i odczyt informacji cyfrowej na taśmie magnetycznej przy wykorzystaniu standardowego magnetofonu kasetowego. Informacja jest kodowana w systemie Kansas City¹⁾ z szybkością 300 b/s. Zapis i odczyt informacji odbywa się bit po bicie, pod programową kontrolą procesora. Układ dekodera kodu Kansas City toleruje nierównomierność przesuwu taśmy dochodzące do $\pm 25\%$, co przesądza o jego dużej niezawodności. Dołączanie magnetofonu nie wymaga żadnych przeróbek w jego części elektronicznej. W magnetofonach wyposażonych w układ zdalnego włączania silnika możliwe jest programowe sterowanie przesuwem taśmy.

Jednopakietowy moduł sterowania pamięcią na dyskach elastycznych FDC umożliwia dołączenie stacji pamięci z formatem typu SP45DE, produkcji MERA-KFAP. Do jego budowy wykorzystano dwa wieloskalone, programowalne układy sprzęgu równoległego typu 8255. Możliwe jest przesłanie rozkazów i danych do formatera oraz odczytanie zarówno stanu formatera, jak i modułu. Moduł zgłasza przerwanie przy zmianie stanu formatera, tzn. gdy jest on zdolny do przyjęcia rozkazu bądź gdy wykonanie rozkazu zostanie zakończone. Transmisja danych z bufora i do bufora formatera może odbywać się pod kontrolą programu, metodą przerwań, lub pod kontrolą układu DMA.

Moduł współpracy z monitorem ekranowym CRT umożliwia wyświetlanie na ekranie monitora (w szczególności na ekranie standardowego odbiornika telewizyjnego) szesnastu wierszy informacji alfanumerycznej lub pseudograficznej przy użyciu sygnałów synchronizacji i jasności. Przez odpowiednie zaprogramowanie modułu można uzyskać 48, 64 lub 80 znaków w wyświetlanym wierszu. Wyświetlana informacja (znaki w kodzie ISO-7) jest pobierana w trybie DMA z programowo określonego, zwartego obszaru RAM. Istnieje możliwość wyświetlania znaków w postaci zwykłej (biały znak na tle ekranu) lub — odwróconej (czarny znak na białym tle). Możliwe jest zaprogramowanie jednej z kilku możliwych postaci kursora. Moduł CRT zaprojektowano w dwóch odmianach: CRT1 — z wykorzystaniem układów MSI TTL (3 pakiety), CRT2 — z wykorzystaniem układu INTEL 8275.

PULPIT TECHNICZNY I MONITOR MAGISTRALI W SYSTEMIE MSM

Magistrala MSM zawiera zestaw linii ułatwiających uruchamianie, konserwację i diagnostykę systemu. Linie te mogą być dołączone do pulpitu technicznego lub do monitora magistrali. Ze względu na użycie nadajników linii z otwartym kolektorem możliwe jest jednocześnie dołączenie pulpitu technicznego i monitora magistrali.

Pulpit techniczny

Pulpit techniczny systemu MSM zawiera cztery przełączniki oraz sześć diod elektroluminescencyjnych. Funkcje przełączników są następujące:

RESET — zerowanie systemu

STOP — zatrzymanie procesora w stanie WAIT

STEP — wykonanie jednego cyklu maszynowego (przy aktywnym przełączniku STOP)

INTR — zgłoszenie przerwania o najwyższym priorytecie.

Każdemu z wymienionych przełączników odpowiada dioda elektroluminescencyjna o tej samej nazwie. Dioda sygnalizuje stan odpowiednich linii magistrali, a nie stan przełącznika, dzięki czemu można obserwować wystereowanie linii także z innego źródła, np. z monitora magistrali. Dwie dodatkowe diody elektroluminescencyjne sygnalizują stan linii WAIT i INTE procesora (dla układu INTEL 8080A).

¹⁾ Logicznej jedynce odpowiadają impulsy o częstotliwości 2400 Hz, a zeru — 1200 Hz

Monitor magistrali

Monitor magistrali MSM jest autonomicznym urządzeniem dołączanym do systemu mikroprocesorowego, zbudowanego z modułów MSM. Spełnia on następujące funkcje diagnostyczne i kontrolne, niezbędne przy uruchamianiu i testowaniu systemu mikroprocesorowego:

- wyświetlanie stanu szyny danych, adresu i szyny sterującej
- wyświetlanie stanu procesora
- zatrzymanie procesora i krokowe wykonanie programu (cykl po cyklu lub rozkaz po rozkazie) z wyświetlaniem aktualnego numeru cyklu maszynowego
- odebranie sterowania magistralą procesorowi i umożliwienie zapisu lub odczytu dowolnej komórki pamięci lub rejestru wejścia-wyjścia (w tym stanie możliwa jest również praca krokowa procesora przy danych dostarczanych z monitora magistrali, można więc odczytywać i zmieniać stan rejestrów wewnętrznych procesora)
- zatrzymywanie procesora na ustalonym adresie pamięci lub rejestru wejścia-wyjścia.

OPROGRAMOWANIE PODSTAWOWE

Oprogramowanie podstawowe systemu MSM składa się z oprogramowania operacyjnego oraz oprogramowania systemowego. Oprogramowanie operacyjne tworzą programy zarządzające pracą systemu i sterujące poszczególnymi urządzeniami zewnętrznymi. Jest ono dostosowywane do konfiguracji systemu zadanej przez użytkownika. Oprogramowanie systemowe obejmuje środki umożliwiające tworzenie, przechowywanie i uruchamianie programów użytkowych.

Oprogramowanie podstawowe modułowego systemu mikroprocesorowego MSM jest ściśle uzależnione od jego konfiguracji sprzętowej. Decydującym czynnikiem wpływającym na wybór oprogramowania podstawowego jest pojemność pamięci RAM oraz dostępność jednostki dysków elastycznych. Dla systemów nie zawierających pamięci na dyskach elastycznych opracowano system monitorujący DEMON/MNEMON i monitor symboliczny SYMON.

Oprogramowanie podstawowe systemu DEMON/MNEMON obejmuje monitor/debugger DEMON i mikroassembler/relokator MNEMON. Wymienione programy tworzą wspólny pakiet i realizują szereg funkcji przydatnych przy tworzeniu, przechowywaniu i uruchamianiu programów użytkowych.

Program DEMON umożliwia:

- wprowadzenie, przeglądanie i modyfikację kolejnych bajtów pamięci
- wypełnianie wskazanego obszaru określoną informacją
- przesuwanie zawartości wskazanego obszaru pamięci
- składowanie zawartości wskazanych obszarów pamięci na taśmie papierowej
- wprowadzanie informacji z taśmy papierowej do pamięci
- krokowe wykonywanie programu
- wykonywanie programu etapami z wykorzystaniem pułapek programowych²⁾
- przeglądanie i modyfikację rejestrów i wskaźników procesora podczas wykonywania programu.

Program MNEMON umożliwia:

- wprowadzanie, modyfikację i wyprowadzanie zawartości obszarów pamięci w formacie rozkazowym, szesnastkowo i znakowo
- relokację wskazanych obszarów pamięci, zawierających programy.

Oprócz systemu monitorującego DEMON/MNEMON opracowano monitor symboliczny SYMON. Tworzą go trzy moduły: monitor/debugger, assembler symboliczny oraz edytor wierszowy. Wymienione programy realizują następujące funkcje:

- wprowadzanie, przeglądanie i modyfikację wskazanych obszarów pamięci
- tworzenie i poprawianie plików z programami źródłowymi w pamięci systemu
- tłumaczenie programów źródłowych
- uruchamianie programów z wykorzystaniem pułapek programowych
- składowanie i odtwarzanie (ang. restore) plików z programami źródłowymi i wynikowymi.

²⁾ Pułapka programowa jest to rozkaz wstrzymania wykonywania programu użytkowego i przejścia do programu nadzorczego

System MSM wyposażony w pamięć RAM o pojemności większej od 16 K bajtów oraz sterownik pamięci na dyskach elastycznych może pracować pod kontrolą systemów operacyjnych CP/M i ISIS-II. Oba wymienione systemy operacyjne są uniwersalne, mają bardzo bogaty zestaw funkcji przeznaczonych do tworzenia oprogramowania użytkowego. System CP/M jest przeznaczony przede wszystkim do opracowywania programów użytkowych, natomiast ISIS-II — ze względu na rozbudowany system programowania — może być wykorzystany do prac rozwojowych przy tworzeniu własnego oprogramowania operacyjnego i systemowego.

Oprogramowanie operacyjne systemów CP/M i ISIS-II obsługuje system plików na dyskach elastycznych, organizuje współpracę z podstawowymi urządzeniami zewnętrznymi oraz tworzy środowisko dla programów systemowych i użytkowych. W skład oprogramowania systemowego wchodzi:

- edytory
- assembly i makroassembly
- kompilatory języków programowania (PL/M, BASIC, FORTRAN, PASCAL)
- programy konsolidujące
- programy adresujące
- programy obsługi bibliotek podprogramów
- programy typu debugger
- interpretery języków programowania (BASIC, FORTH)
- programy konwersji nośników.

ROZWOJ ORAZ ZASTOSOWANIA SYSTEMU MSM

Modułarna struktura systemu MSM umożliwia tworzenie różnych konfiguracji w zależności od potrzeb. Dzięki zastosowaniu odpowiednich buforów magistrali, moduły standardowe są przystosowane do pracy w systemach rozbudowanych. W pewnych zastosowaniach wymagane jest użycie modułów opracowywanych przez użytkowników. Projektując je można wykorzystywać opracowane moduły uniwersalne oraz standardowe układy współpracy z magistralą.

Obecnie opracowuje się szereg nowych modułów standardowych, m.in.:

- moduły specjalizowanych procesorów (arytmetyczny, graficzny, komunikacyjny)
- moduły śledzenia i sterowania procesami fizycznymi (przetworniki cyfrowo-analogowe i analogowo-cyfrowe, przetworniki napięcie-częstotliwość i częstotliwość-napięcie, multiplexery analogowe, układy wejść i wyjść cyfrowych z izolacją galwaniczną. Zaawansowane są ponadto prace nad narzędziami ułatwiającymi uruchamianie i testowanie systemów mikroprocesorowych (symulatory, emulatory, itp.).

* * *

Od roku kilka systemów MSM jest intensywnie wykorzystywanych w laboratorium Instytutu Informatyki (w celach dydaktycznych i badawczych). Powstały już pierwsze urządzenia wykorzystujące te systemy, jak np. System Uruchomieniowy Mikroprogramowanych Układów Sterowania — SUMUS. Wymienione urządzenie umożliwia symulowanie pamięci sterującej (o różnych strukturach) oraz fizyczne uruchamianie układów mikroprogramowanych (dzięki układom śledzenia, pułapek itp.). Obecnie systemy MSM wykorzystuje się przy opracowywaniu specjalizowanych urządzeń do analizy sygnałów analogowych dla celów medycznych (miograf), geofizycznych (poszukiwanie złóż ropy naftowej) i przemysłowych (nadzorowanie procesów technologicznych). Ponadto prowadzi się prace nad lokalnymi sieciami mikroprocesorowymi. Kilka systemów MSM jest wykorzystywanych na zamówienie zewnętrzne.

LITERATURA

- [1] Borrill P. L.: Microprocessors Bus Structures and Standards. IEEE Micro, Vol. No. 1, p. 84, 1981
- [2] Titus C. A., Titus J. A., Larsen D. G.: STD Bus Interfacing. Howard Sams, Indianapolis, IN, 1982
- [3] Prolog Corp.: Series 7000 STD-BUS Technical Manual. Monterey, CA, 1979
- [4] Woźniak A.: Analiza współpracy układów rodzin mikroprocesorowych INTEL 8080/85, ZILOG Z-80, MOTOROLA 6800. Raport Badawczy I.I, PW nr 47, Warszawa, 1982
- [5] Woźniak A., Pawłowski M., Rzeszut J., Sosnowski J.: Dokumentacja techniczna systemu MSM. Instytut Informatyki PW, 1982
- [6] Woźniak A., Rzeszut J., Sosnowski J.: Modułarny system mikroprocesorowy. Raport Badawczy I.I, PW nr 34, Warszawa 1981.

Alfaskop ALFA 311M jako urządzenie wejścia-wyjścia systemu mikrokomputerowego

Monitory ekranowe wyposażone w klawiaturę alfanumeryczną cieszą się dużym uznaniem wśród użytkowników systemów komputerowych. Umożliwiają one bezpośredni kontakt z komputerem, także podczas przetwarzania danych. W odróżnieniu od innych urządzeń wejścia-wyjścia (z wyjątkiem grafoskopów) nie wymagają pośrednictwa dodatkowych nośników informacji, jak: taśma papierowa, dyski i taśmy magnetyczne, karty perforowane itp.

Istnieją różne metody zapamiętywania tekstów i wyświetlania na ekranie alfaskopu. Należy do nich zaliczyć:

- zapamiętywanie informacji bezpośrednio na ekranie lampy pamięciowej
- przechowywanie tekstu w pamięci operacyjnej komputera i repetycyjny odczyt
- repetycyjny odczyt tekstu z wewnętrznego bufora monitora.

W systemach mikroprocesorowych zastosowanie pierwszego sposobu jest bardzo kosztowne, ze względu na koszt lampy pamięciowej. Druga metoda natomiast jest mniej skuteczna od trzeciej, ponieważ odczyt z pamięci mikrokomputera jest procesem czasochłonnym.

Najbardziej użyteczne w praktyce okazało się zastosowanie alfaskopu ALFA 311M z wewnętrznym buforem. Alfaskop wykorzystano jako urządzenie wejścia-wyjścia małego systemu mikrokomputerowego, opartego na mikroprocesorze z rodziny INTEL. W przykładowym rozwiązaniu posłużono się mikroprocesorem INTEL 8008 [1]. Zastosowanie nowszych typów mikroprocesorów jest także dopuszczalne i umożliwia zwiększenie efektywności systemu.

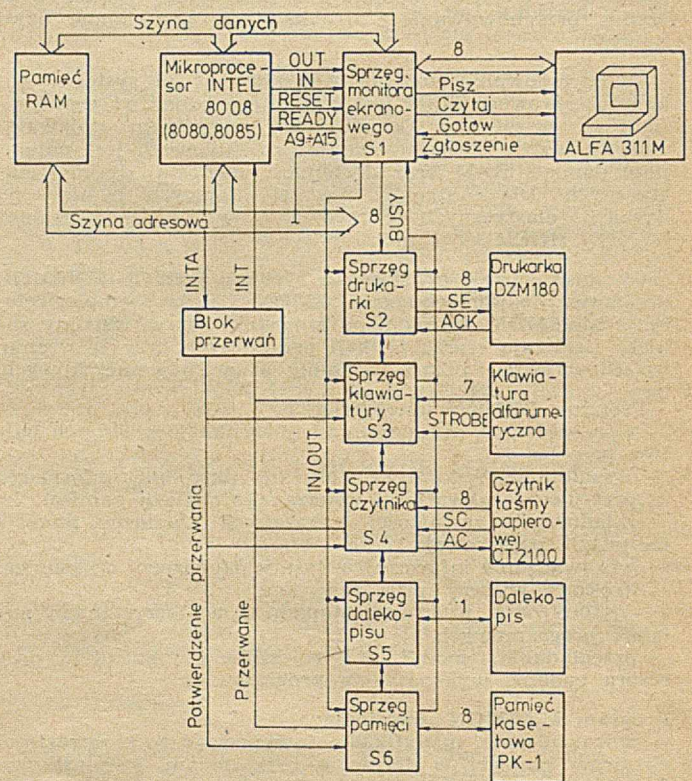
DOŁĄCZENIE ALFASKOPU DO SYSTEMU MIKROKOMPUTEROWEGO

Alfaskopy ALFA 311M sprzężono z systemem mikrokomputerowym za pomocą specjalnie opracowanego sprzęgu (rys.). Sprzęg jest zbudowany z elementów TTL o małym i średnim stopniu scalenia i umożliwia łączenie szeregu innych urządzeń zewnętrznych. Niewielka liczba użytych elementów i pewność działania uzasadniają jego zastosowanie zamiast uniwersalnych lub specjalizowanych sprzęgów LSI w postaci układów scalonych.

Przy projektowaniu sprzęgu starano się uwzględnić następujące założenia:

- rozszerzalność zestawów, tj. możliwość zwiększenia liczby urządzeń zewnętrznych przez dołączanie kolejnych modułów sprzęgu

- łatwość oprogramowania komunikacji z urządzeniami zewnętrznymi
- możliwość wprowadzania danych z klawiatury wyłącznie przez bufor wewnętrzny alfaskopu (takie rozwiązanie umożliwia redagowanie i korygowanie danych przed komunikacją z mikroprocesorem)
- dołączanie urządzeń zewnętrznych bez przeróbek konstrukcyjnych (w alfaskopie wykorzystano tzw. „mały interfejs” [2]).



Schemat blokowy systemu mikrokomputerowego z alfaskopem ALFA 311M



Dr WŁADYSŁAW BINKOWSKI ukończył w 1971 r. studia na kierunku elektrycznym Wydziału Techniki Uniwersytetu Śląskiego. Stopień doktora nauk technicznych uzyskał na Wydziale Elektroniki, Automatyki i Elektroniki w AGH Kraków w roku 1980. Od 1971 r. zatrudniony jest na Wydziale Techniki Uniwersytetu Śląskiego. Zajmuje się dynamiką systemów elektromechanicznych sterowanych przez układy mikroprocesorowe.



Mgr inż. MAREK FURYK jest absolwentem Wydziału Automatyki i Informatyki Politechniki Śląskiej w Gliwicach. Od 1979 r. zatrudniony jest na Wydziale Techniki Uniwersytetu Śląskiego. Zajmuje się zastosowaniami i budową systemów mikroprocesorowych.

Spełnienie tych warunków okazało się łatwe dzięki zastosowaniu alfaskopu ALFA 311M. Egzemplarze alfaskopów, którymi dysponowali autorzy, były pierwotnie przystosowane do współpracy z komputerem ODRA 1305. Po usunięciu tzw. „dużego interfejsu” [2] umieszczono w alfaskopie płytke komunikacji z mikrokomputerem.

Alfaskop jest sterowany sygnałami mikrokomputera, a komunikacja odbywa się wyłącznie w czasie realizacji rozkazów IN i OUT adresowanych do alfaskopu. Mikroprocesor wstrzymuje przesyłanie danych do alfaskopu po odebraniu sygnału „koniec kadru” albo programowo — przez wysłanie bajtu danych z ustawionym ósmym bitem (bit ten nie jest wykorzystywany w kodzie IS07). Kontynuacja transmisji może nastąpić po zgłoszeniu przez operatora gotowości do przyjęcia lub wysłania kolejnej porcji informacji. Zgłoszenie gotowości realizuje się przyciskiem MC klawiatury alfaskopu, tak więc dialog odbywa się przy użyciu bardzo prostych środków programowych.

Po wprowadzeniu informacji z klawiatury kursor może pozostawać w dowolnym miejscu ekranu. Taka organizacja dialogu wymaga znajomości tylko jednej zasady, aby po wprowadzeniu informacji nacisnąć przycisk MC. Jednakże, brak dostępnego z zewnątrz, licznika zawierającego adres kursora zmusza do odliczania znaków od początku kadru.

Dzięki rozbudowanej organizacji wewnętrznej alfaskop, mający 1024-znakowy bufor, nie wymaga zwiększenia pojemności pamięci operacyjnej mikrokomputera ani wydłużenia sekwencji rozkazów. Przeciwnie — okazało się, że bufor alfaskopu można z powodzeniem wykorzystać jako pomocniczą pamięć zewnętrzną o pojemności 1024 bajtów.

PROGRAMY ORGANIZACYJNE

Po zainstalowaniu alfaskopu w systemie mikrokomputerowym opracowano dla niego pakiet programów organizacyjnych. Rodzaj programów jest określony zastosowaniem

mikrokomputera w procesie dydaktycznym. Służą one do wprowadzania danych z alfaskopu, wyświetlania na ekranie zawartości dowolnego obszaru pamięci, wykonywania skoków do dowolnych miejsc w pamięci na polecenie z klawiatury, w celu rozpoczęcia realizacji zadania użytkowego. Z klawiatury monitora inicjuje się wszystkie operacje oraz przerywa realizację zadania przez zerowanie systemu przyciskiem RESET.

Do pakietu dołączono podprogramy czytania dwóch kolejnych znaków z bufora alfaskopu (od miejsca wskazanego przez kursor) i konwersji tak otrzymanej liczby w kodzie szesnastkowym na liczbę dwójkową (i odwrotnie). Ponadto dołączono podprogramy zapisu ciągu znaków z bufora do pamięci oraz odczytu z pamięci do bufora. Po zapisaniu jednego kadru dalsza transmisja jest wstrzymana sygnałem generowanym przez alfaskop w chwili zapełnienia bufora. Dzięki temu informacja zapisana na ekranie zanika, lecz oczekuje na akceptację odbiorcy.

Cały pakiet programów organizacyjnych zajmuje zaledwie 200 bajtów pamięci stałej ROM i dwa bajty pamięci RAM. Wynika stąd, że dołączenie alfaskopu z wewnętrznym buforem i prostą organizacją układową jest praktycznie bez jakichkolwiek nakładów na dodatkowe oprogramowanie i dodatkową pamięć przeznaczoną do przechowywania kadru.

* * *

Dzięki dołączeniu alfaskopu system mikrokomputerowy stał się wszechstronniejszy, a jednocześnie łatwiejszy w obsłudze. Wadą tego rozwiązania jest zbyt mała szybkość transmisji znaków (ograniczona szeregowym dostępem do informacji w buforze). Jednakże w systemach dydaktycznych, jak również przy wykorzystaniu systemu jako „inteligentnej maszyny do pisania” lub małego banku danych, opóźnienie to nie odgrywa większej roli.

Opracowany sprzęg i jego realizacja potwierdzają łatwość dołączenia różnych urządzeń zewnętrznych do mikrokomputera. Konstrukcja sprzęgu nie różni się stopniem złożoności od innych — zbudowanych z układów scalonych LSI, lecz jest znacznie tańsza i oparta na elementach krajowych.

Alfaskopy ALFA 311M, dołączone do mikrokomputera za pomocą opracowanego sprzęgu, okazały się wyjątkowo łatwe w programowaniu i eksploatacji.

LITERATURA

- [1] INTEL Corp., 8008 — 8-bit Parallel Central Processor Unit. Santa Clara, CA, 1977
- [2] Ośrodek Badawczo-Rozwojowy ERA: ALFA 311M — Dokumentacja techniczno-ruchowa. Warszawa, 1981.



Mgr inż. ANDRZEJ PRZYBYŁSKI ukończył studia w 1976 r. na Wydziale Elektroniki Politechniki Wrocławskiej. Pracował przy serwisie maszyn cyfrowych ODRA i automatycznych analizatorach biochemicznych. Od 1981 roku jest pracownikiem Zakładu Elektroniki Wydziału Techniki Uniwersytetu Śląskiego w Katowicach. Zajmuje się projektowaniem i aplikacją systemów mikroprocesorowych.

KONFERENCJE

SIGGRAPH'83

W dniach 25—29 lipca br. w Detroit (Michigan, USA) odbędzie się dziesiąta coroczna konferencja grafiki komputerowej i technik interakcyjnych (the Tenth Annual Conference on Computer Graphic and Interactive Techniques) SIGGRAPH'83. Organizatorem konferencji jest sekcja grafiki komputerowej amerykańskiego stowarzyszenia sprzętu komputerowego ACM/SIGGRAPH (the Association for Computing Machinery's Special Interest Group on Computer Graphics).

Pierwsza konferencja z tej serii odbyła się w 1974 r. w Boulder (Colorado, USA) z udziałem niewielkiej, jak na stosunki amerykańskie, liczby ok. 600 uczestników. Na podstawie zaobserwowanego, szczególnie szybkiego rozwoju dziedziny grafiki komputerowej, a w konsekwencji gwałtownego wzrostu

liczby uczestników kolejnych konferencji, organizatorzy konferencji tegorocznej spodziewają się w bieżącym roku frekwencji rzędu 20 tys. uczestników. Celem zaspokojenia bardzo już zróżnicowanych zainteresowań specjalistów tej dziedziny, w programie konferencji przewidziano m.in. ponad 25 sesji tematycznych, ok. 50 referatów i dyskusji panelowych, kompleksową wystawę sprzętu, projekcje generowanych komputerem filmów i zapisów magnetowidowych, wystawę dzieł sztuki komputerowej, pokazy nowych rozwiązań z zakresu sprzętu i oprogramowania grafiki komputerowej oraz spotkania specjalistyczne grup użytkowników. Bliższe szczegóły na temat programu konferencji otrzymać można w biurze organizacyjnej konferencji: SIGGRAPH'83 Conference Office, 111 East Wacker Drive, Chicago, Illinois 60601, USA.

Jednopakietowy inteligentny sterownik graficzny MSG-3C

Od kilku lat obserwuje się wyraźną tendencję wyposażania systemów mikrokomputerowych w monitory ekranowe, służące nie tylko do alfanumerycznego ale i graficznego obrazowania danych. Tendencja ta jest spowodowana z jednej strony spadkiem cen właściwych monitorów (wskaźników), a z drugiej — stosowaniem systemów mikrokomputerowych w tych dziedzinach, w których graficzne obrazowanie danych jest bardzo pomocne, a nawet niezbędne. To ostatnie dotyczy zwłaszcza:

- wyświetlania wyników obliczeń, obserwacji itp. w postaci wykresów, względnie histogramów
- wyświetlania stanów (sytuacji) sterowanych czy też kontrolowanych obiektów — tzw. tablice synoptyczne itp.
- wyświetlania obrazów tworzonych bądź podlegających obróbce — w systemach projektowania wspomaganego (ang. CAD) bądź przetwarzania obrazów (ang. image processing).

W praktyce rysują się dwie główne drogi tworzenia sprzętowo-programowych środków graficznego obrazowania danych:

- a) tworzenie wydzielonych „wolno stojących” terminali graficznych, sprzężonych z właściwym systemem (mikro) komputerowym przez standardowy sprzęg (np. szeregowy V24 albo RS232), analogicznie do produkowanych od lat terminali alfanumerycznych — przykładem może tu być rozbudowany terminal graficzny VT 105 firmy DEC
- b) tworzenie wyspecjalizowanych pakietów sprzętowych (często „inteligentne”, wyposażone w odpowiednie oprogramowanie graficzne w pamięci ROM), które — stanowiąc opcje — mogą być wkładane do gotowych, bardziej uniwersalnych standardowych konstrukcji, jak np. do kaset mikrokomputerowych systemów sterowania. W tym przypadku monitorem (wskaźnikiem) może być np. standardowy odbiornik telewizyjny albo monitor studyjny, nawet nie przeznaczony pierwotnie do celów komputerowych.

Wariant a) jest może dogodniejszy dla projektanta „docelowego” systemu, ale też jest on droższy: zarówno dla producenta, który musi więcej zainwestować w fazie opracowywania konstrukcji, jak i dla finalnego odbiorcy, który musi zapłacić za terminal bardziej uniwersalny, ale i bardziej rozbudowany. Istotnym problemem dla takiego terminala może być ustalenie odpowiednich standardów parametrów i właściwości funkcjonalnych (wraz z językiem operatorskim) — które w maksymalnym stopniu zaspokoiliby potrzeby wielu różnych użytkowników.

Natomiast wariant b) — zwłaszcza jeżeli mamy do czynienia z pojedynczym pakietem — jest znacznie tańszy, a jego szybkie wdrożenie do produkcji jest w naszych warunkach bardziej realne. Wydaje się, że wariant ten cha-

rakteryzuje się również wyraźnie lepszym stosunkiem kosztów do określonych parametrów technicznych, natomiast wykazuje następujące wady:

- ograniczenia parametrów i właściwości funkcjonalnych, wynikające z ograniczeń powierzchni konstrukcyjnej
- konieczność dopasowywania się do sprzęgu wewnątrz-kasetowego konkretnej konstrukcji
- stawianie do dyspozycji użytkownika modułu, który dla niego jest w pewnym stopniu półproduktem, wymagającym dodatkowego oprogramowania użytkowego.

Sterownik graficzny MSG-3C stanowi jednopakietową realizację wariantu b). Zdaniem autorów jego zalety przeważają nad wspomnianymi wadami, tym bardziej, że:

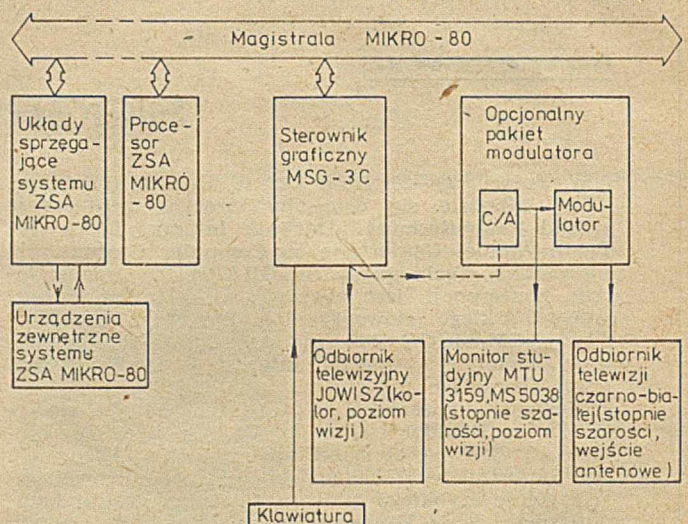
- ograniczenia parametrów i właściwości funkcjonalnych maleją ze wzrostem stopnia scalenia zastosowanych układów, a także powierzchni pakietów
- problemy z dopasowaniem się do konkretnej konstrukcji maleją wraz ze standaryzacją sprzęgów wewnątrz-kasetowych.

Poniżej zostanie pokrótce omówiona budowa sterownika MSG-3C. Ze względu na to, że prace nad wykonaniem serii prototypowej są jeszcze w toku, zastrzega się możliwość wprowadzenia drobnych zmian.

PARAMETRY WYŚWIETLANEGO OBRAZU

- liczba punktów w linii — 240
- liczba linii — 256, w tym różnych linii (kompresja obrazu) — 96
- liczba barw albo stopni szarości (nie licząc czarnego tła) — 4
- sposób definiowania kolorów albo stopni szarości dla pasków — 6 punktów
- możliwość powiększania ćwiartek obrazu — 2 ×

Sterownik MSG-3C jest zamontowany na pojedynczym pakiecie podwójnej Eurokarty o wymiarach 160 × 233,4 mm.



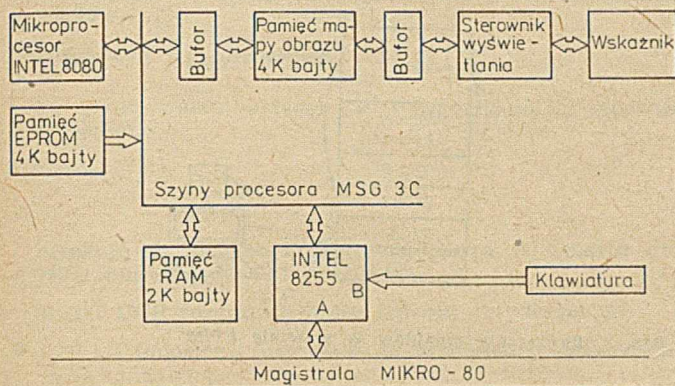
Rys. 1. Schemat blokowy systemu mikrokomputerowego ze sterownikiem MSG-3C



Zyciorys dr. MARKA TADEUSZA JANKOWSKIEGO zamieściliśmy w nr 1/83.

Mgr inż. JACEK WERTEL ukończył w 1977 r. Wydział Elektroniki Politechniki Warszawskiej, specjalność — budowa i oprogramowanie maszyn cyfrowych. Od 1980 r. pracuje w ITE-CEMI, zajmując się projektowaniem sprzętu i oprogramowania inteligentnych terminali graficznych z wykorzystaniem mikroprocesorów. Interesuje się problemami teorii i praktyki systemów wieloprocusorowych.

Wymiary te, a także zastosowane złącza pośrednie 64-stykowe oraz sprzęg wewnętrzny (sygnaly elektryczne i logiczne — wliczając w to również napiecia zasilajace — oraz ich rozmieszczenie na złączach) są zgodne ze standardem zastosowanym w kasie systemu mikrokomputerowego MERA-ZSA MIKRO-80. Dzięki temu, poprzez specjalny, umieszczony na pakiecie, adapter sprzęgu, MSG-3C może pełnić funkcje terminala graficznego, a zarazem korzysta z zasobów programowych i sprzętowych systemu MIKRO-80, zwłaszcza z czytnika i perforatora taśmy oraz docelowo z pamięci dyskietkowej.



Rys. 2. Schemat blokowy sterownika

MARIANNA SOBCZYK (Oprac.)

Warszawa

Dyskowy system operacyjny CP/M. Część 1

System CP/M został opracowany przez firmę DIGITAL RESEARCH początkowo dla mikrokomputera MDS 800, lecz obecnie jest uważany za standardowy. Świadczy o tym liczba firm dostarczających oprogramowanie działające pod nadzorem systemu CP/M, liczba producentów wyposażających swój sprzęt w ten system, jak również — liczba użytkowników zainteresowanych dostępnością systemu CP/M w nabywanym sprzęcie.

System operacyjny CP/M może działać na mikrokomputerach, które są oparte na mikroprocesorach INTEL 8080 i 8085 lub Z-80, mają pamięć typu RAM o minimalnej pojemności 20 K bajtów (począwszy od adresu zerowego) i nie używają niektórych rodzajów przerwań (IM1 — w przypadku mikroprocesora Z-80). Oprogramowanie napisane i działające pod nadzorem systemu CP/M zachowuje wszystkie cechy przenośności na dowolną maszynę wyposażoną w ten system (przy pewnych założeniach dotyczących środowiska sprzętowego).

Firma DIGITAL RESEARCH dostarcza system CP/M na dysku elastycznym i dołącza do niego siedem podręczników opisujących poszczególne elementy oprogramowania. W celu skutecznego korzystania z systemu CP/M należy posługiwać się literaturą firmową, natomiast niniejszy artykuł stanowi jedynie ogólną prezentację budowy i cech systemu. Do pełnego zrozumienia przedstawionych informacji konieczna jest znajomość języka symbolicznego mikroprocesora INTEL 8080 oraz organizacji jego rejestrów.

Schemat blokowy usytuowania sterownika MSG-3C w całym systemie mikrokomputerowym jest pokazany na rysunku 1. Poza oczywistą możliwością korzystania z konsoli MIKRO-80, sterownik — poprzez dodatkowy układ sprzęgu równoległego — może korzystać z własnej klawiatury alfanumerycznej. Sterownik jest ponadto wyposażony w sprzęg kolorowego monitora (OTV Jowisz) oraz — po dołączeniu do opcjonalnego konwertera cyfrowo-analogowego i modulatora — w sprzęg dowolnego czarno-białego odbiornika telewizyjnego lub profesjonalnego monitora studyjnego, co umożliwia wyświetlanie obrazów o kilku stopniach szarości.

Schemat blokowy sterownika jest przedstawiony na rysunku 2. Poza mikroprocesorem oraz układami sprzęgu, buforami itp., sterownik jest wyposażony w pamięć programów i danych EPROM o pojemności 4 K bajtów, pamięć mapy obrazu PMO-RAM, również o pojemności 4 K bajtów, roboczą pamięć RAM o pojemności 2 K bajtów oraz zrealizowany sprzętowo właściwy sterownik wyświetlania. Zawartość PMO stanowi w pewien sposób skomprimowane odwzorowanie wyświetlanego obrazu. Zawartość ta jest modyfikowana programowo przez mikroprocesor, który ma dostęp do wszystkich bloków pamięci na pakiecie. Sam proces wyświetlania jest zorganizowany sprzętowo przez sterownik na podstawie aktualnej zawartości PMO, przy czym pamięć ta w trakcie wyświetlania jest odłączana od szyn mikroprocesora.

Sterownik wykorzystuje najbardziej nowoczesne elementy z listy preferencyjnej CEMI (w nawiasach podano termin rozpoczęcia produkcji): mikroprocesor MCY 7880N (1982), programowany równoległy układ wejścia-wyjścia MCY 7855N (1982), pamięć statyczną RAM 4 K bajtów MCY 7114N (1983) oraz pamięć EPROM 16 K bajtów MCY 7716R (1984).

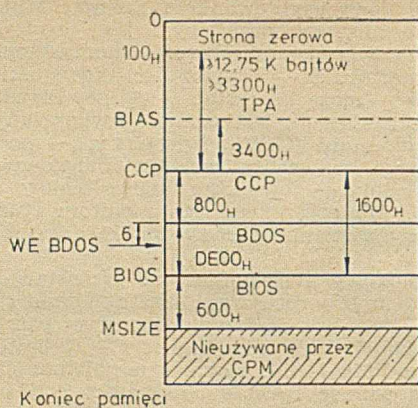
Dla uniknięcia nieporozumień zebrano tu pewne uwagi o stosowanych oznaczeniach i określeniach mogących budzić wątpliwości. Tak więc — indeks H (np. 13_H) oznacza szesnastkowy zapis liczby, a połączenie r z dużymi literami (np. rA , rHL) — wskazuje rejestr mikroprocesora. W systemie CP/M można pomijać niektóre parametry polecenia, nadając im tym samym zadane wartości, zwane wartościami domyślnymi.

Ze względu na to, że polecenia systemu CP/M są w istocie wywołaniami programów, rozróżniono polecenia rezydentne, którym odpowiadają programy stale przechowywane w pamięci operacyjnej i nierezydentne, których programy zapisano na dysku.

BUDOWA SYSTEMU CP/M

Pamięć operacyjna w systemie CP/M dzieli się na pięć obszarów (rys. 1), określonych przez rezydujące w nich programy:

- strona zerowa
- obszar tymczasowego przechowywania programu TPA (ang. *Transient Program Area*)
- obszar przetwarzania dyrektyw operatora CCP (ang. *Console Command Processing*)
- obszar zajęty przez podstawowy system dyskowy BDOS (ang. *Basic Disk Operating System*)
- obszar podstawowego systemu wejścia-wyjścia BIOS (ang. *Basic I/O System*).



Rys. 1. Obraz pamięci w systemie CP/M

Strona zerowa zapewnia łączność między programami CCP, BDOS i programami użytkownika. Jest tworzona dynamicznie podczas inicjowania, tzn. nie ma jej w obrazie systemu na dysku. Opis poszczególnych komórek podano w tabeli 1. Przez komórkę o adresie 0005H następuje wywołanie systemu CP/M. W obszarze TPA oprócz programów użytkowych wykonywane są programy poleceń nie-rezydentnych.

Tabela 1. Opis strony zerowej

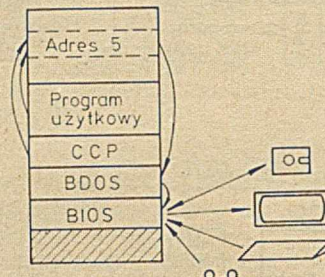
Adresy	Zawartość
0000—0002	Skok do programu ładującego, Skok do adresu 0000H powoduje ponowne załadowanie modułów CCP i BDOS, a następnie — przekazanie sterowania do CCP. Bajty 1 i 2 zawierają adres BIOS+3.
0003	IOBYTE — przydzielenie urządzeń zewnętrznych: jest przeznaczony dla modułu BIOS i programów nierezylujących (np. PIP)
0004	Domyślny numer dysku
0005—0007	Skok do BDOS. Wywołanie modułu BDOS następuje przez wykonanie rozkazu CALL 0005H (po uprzednim umieszczeniu kodu funkcji w rejestrze C). Bajty 6 i 7 zawierają adres modułu
0008—002F	Bajty odpowiadające rozkazom RESTART, od 1 do 5, nie używane przez CP/M (mogą być wykorzystane przez użytkownika)
0030—0037	RESTART 6 — zarezerwowane dla systemu
0038—003A	RESTART 7 — używane przez program DDT
003B—003F	Zarezerwowane dla systemu
0040—004F	Zarezerwowane dla modułu BIOS
0050—005B	Zarezerwowane dla systemu
005C—007C	Obszar tworzenia tablicy FCB (File Control Block) dla programów wywoływanych przez moduł CCP
007D—007F	Rozszerzenie tablicy FCB dla plików o dostępie bezpośrednim
0080—00FF	Bufor o długości 128 bajtów używany do transmisji dyskowych; umieszcza się tu także treść polecenia, gdy moduł CCP ładuje program do TPA

Trzy moduły, stanowiące właściwy system CP/M, są zapisane na dysku i ładowane do pamięci operacyjnej w chwili inicjowania systemu. Moduł przetwarzania poleceń operatora CCP — umożliwia wykonywanie poleceń rezydentnych lub ładowanie i wykonywanie poleceń nierezylujących oraz programów użytkowych. System zarządzania zbiorami i sterowania urządzeniami zewnętrznymi BDOS stanowi połączenie programów użytkowych z modulem BIOS; dostarcza on wszystkich danych potrzebnych do wykonania operacji wejścia-wyjścia (tzn. numer dysku, numer ścieżki, numer sektora itp.). Moduł sterowania fizycznymi urządzeniami wejścia-wyjścia BIOS — zależy od środowiska sprzętowego i dlatego wymaga dopasowania do danego mikrokomputera.

Adres BIAS, zaznaczony w obszarze TPA (rys. 1), służy do obliczania adresów w chwili tworzenia modułu BIOS. Adresy modułów oblicza się według następujących wzorów (MSIZE odpowiada rozmiarowi pamięci, choć nie musi pokrywać się z jej fizycznym końcem):

$$\begin{aligned} \text{BIAS} &= (\text{MSIZE} - 20) \times 1024 \\ \text{CCP} &= \text{BIAS} + 3400\text{H} \\ \text{BDOS} &= \text{CCP} + 806\text{H} \\ \text{BIOS} &= \text{CCP} \times 1600\text{H} \end{aligned}$$

Na rysunku 2 przedstawiono związki zachodzące między różnymi modułami systemu CP/M.



Rys. 2. Powiązania modułów w systemie CP/M

PLIKI

Plik jest ciągiem rekordów, jednostek niepodzielnych dla modułu BDOS oraz transmisji dyskowych. Długość rekordu wynosi 128 bajtów i na ogół odpowiada rozmiarowi sektora na dysku elastycznym. Plik oznaczamy trzema parametrami: numerem dysku, nazwą i typem. Na tym samym dysku nie mogą istnieć dwa pliki o tej samej nazwie i typie, należące do tego samego użytkownika.

Formalna składnia opisu pliku jest następująca:

[d:] <nazwa pliku> [.<typ>]

Nawiasy kwadratowe oznaczają parametr opcjonalny, nawiasy trójkątne — ciąg znaków, d — numer jednostki dyskowej, na której znajduje się plik.

Jednostki dyskowe zawierające pliki przeznaczone dla modułu CCP oznaczają się literami od A do P zamiast numerów od 0 do 15. Brak parametru powoduje użycie wartości domyślnej.

Nazwa pliku jest ciągiem o długości od 1 do 8 znaków identyfikujących, a typ — ciągiem o długości od 1 do 3 znaków. Jeżeli ten ostatni parametr jest pominięty, to przyjmuje się, że typ jest opisany spacją.

Typ pliku może być dowolnie wybrany przez programistę, natomiast programy systemowe wykorzystują lub tworzą pliki następujących typów:

.COM — plik ładowany i wykonywany przez moduł CCP rozkazem LOAD, plik tego typu jest tworzony na podstawie pliku typu .HEX

.HEX — plik wynikowy tworzony przez assembler, może być ładowany i wykonywany tylko pod nadzorem programu uruchamiania DDT

.PRN — plik tekstu programu, tworzony przez assembler .ASM — plik źródłowy dla assemblera, może być tworzony przy użyciu programu redagującego ED

.LIB — plik makrorozkazów przeznaczony dla makroassemblera MAC (nie istniejącego w podstawowej wersji systemu) .BAK — plik tworzony przez program ED; jest kopią pliku pierwotnego i może być użyteczny w przypadku awarii podczas poprawiania programu

.SUB — plik poleceń używany przez program SUBMIT, może być tworzony przy użyciu programu redagującego

.\$\$\$ — tymczasowy plik roboczy, tworzony podczas pracy niektórych programów i niszczone po zakończeniu działania.

Przykładowo — jeżeli plik A:DUMP.ASM jest plikiem wejściowym dla assemblera, to zostaną utworzone pliki:

DUMP.HEX i DUMP.PRN. Po użyciu rozkazu LOAD otrzymamy ostatecznie plik DUMP.COM.

W systemie CP/M dopuszcza się grupowy opis plików. Gwiazdka (*) oznacza wszystkie możliwe nazwy lub typy, a znak zapytania (?) zastępuje wszystkie inne znaki. Przykładowo — skasowanie wszystkich plików na dysku C nastąpi po wydaniu polecenia:

ERA C: * . *

zaś opis w postaci:

?????.COM

odczytują się: wszystkie pliki typu .COM o nazwie od 1 do 4 znaków.

POLECENIA REZYDENTNE

Po zainicjowaniu systemu CP/M sygnalizuje się obecność wydrukiem:

CP/M vers 2.2

A>

i oczekuje na polecenie operatora (litera A oznacza domyślny numer jednostki dyskowej).

Moduł CCP rozpoznaje sześć poleceń rezydentnych:

- **DIR** — umożliwia uzyskanie opisów plików dyskowych; np. po użyciu polecenia:

A>DIR A:ECHO. *

otrzymamy listę plików ECHO wraz z ich typami z dysku A

- **ERA** — umożliwia kasowanie plików dyskowych

- **REN** — służy do zmiany nazwy i typu pliku, np. po wykonaniu polecenia:

A>REN DUMP.SRC=DUMP.ASM

typ .ASM zostanie zastąpiony typem .SRC.

- **TYPE** — umożliwia wyświetlenie lub wydrukowanie zawartości wskazanego pliku w kodzie ASCII na urządzeniu operatora, np.:

A>TYPE DUMP.ASM
A>TYPE B:ECHO.ASM

- **SAVE** — umożliwia ochronę obszaru pamięci; począwszy od adresu 100H, przez przechowanie kopii w zadanym pliku, który może być typu .COM, ponieważ jest dokładnym obrazem pamięci; polecenie jest użyteczne do zabezpieczenia programu zmodyfikowanego — jego składnia ma postać:

A>SAVE xx <nazwa pliku>

przy czym xx oznacza liczbę stron pamięci, które należy przechować (jedna strona odpowiada blokowi 256 bajtów), np. polecenie:

A>SAVE 17 PRES.COM

spowoduje utworzenie zbioru PRES.COM, który jest obrazem obszaru pamięci od adresu 100H do adresu 1200H

- **USER** — powoduje przydzielenie dysku użytkownikom oznaczonym numerami (od 0 do 15), tzn.:

A>USER n

przy czym pliki określonego użytkownika nie są znane innemu; przekazywanie plików jednego użytkownika do obszaru roboczego użytkownika aktualnego jest możliwe przy użyciu polecenia PIP; wartość domyślna numeru użytkownika przyjmowana przez system wynosi 0.

POLECENIA NIEREZYDENTNE

Jeżeli po wydaniu polecenia przez operatora, moduł CCP stwierdzi, że jest to polecenie nierezydentne, to poszukuje na dysku pliku typu .COM o zadanej nazwie. Poszczególne programy odpowiadające poleceniom nierezydentnym są następujące:

- **STAT** — umożliwia określenie stanu i parametrów dysku, określenie i zmianę stanu pliku (stan pliku jest stały i może być zadeklarowany jako R O, R/W, SYS lub DIR, ang. Read-Only, Read-Write, System Directory); wykonując polecenie :A>STAT B:* otrzymamy listę plików zapisanych na dysku B:

RECS	BYTES	EXT	ACC	
66	9K	1	R/W	B:ASM.COM
54	7K	1	R/W	B:ED.COM
60	8K	1	R/W	B:PIP.COM
42	6K	1	R/W	B:STAT.COM

- **ED**, program redagujący — umożliwia tworzenie i modyfikowanie plików źródłowych o dowolnej długości przy użyciu poleceń wewnętrznych (na rysunku 3 przedstawiono obieg informacji w zależności od rodzaju wykonywanego polecenia); otrzymany plik ma tę samą nazwę i ten sam typ co plik pierwotny, który pozostaje na dysku z początkową nazwą i typem .BAK; jeżeli po wywołaniu programu:

A>ED [d:] <nazwa pliku> [.<typ>]

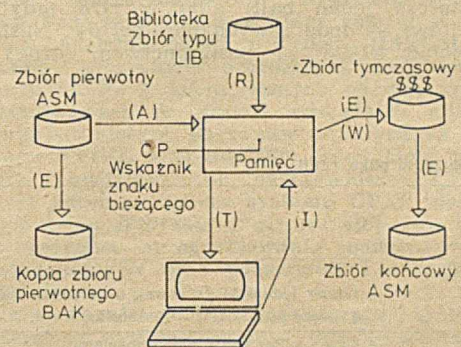
okaże się, że plik nie istnieje, to otrzymujemy odpowiedź:

NEW FILE

a w przeciwnym przypadku;

: *

(tworzenie nowego pliku przy użyciu programu ED należy zakończyć znakiem CTRL-Z)



Rys. 3. Obieg informacji w programie redagującym ED

- **ASM**, assembler — akceptuje wyłącznie rozkazy mikroprocesora INTEL (bez makrorozkazów), w jednym wierszu można umieścić kilka rozkazów oddzielonych wykrzyknikami (!), a małe i duże litery nie są odróżniane; plik źródłowy musi być typu .ASM, a wywołanie assemblera ma postać:

A>ASM<nazwa pliku> [P1P2P3]

gdzie P1 oznacza jednostkę dyskową, na której znajduje się plik źródłowy, P2 — jednostkę dyskową, na której chcemy umieścić plik wynikowy typu .HEX (dla P2=Z assembler nie tworzy pliku wynikowego), a P3 — jednostkę dyskową, na której ma być umieszczony plik tekstu programu (dla P3=X tekst jest wyświetlany bezpośrednio na konsoli operatorskiej, a dla P3=Z assembler nie tworzy tego pliku).

- **LOAD** — umożliwia utworzenie pliku typu COM na podstawie pliku wynikowego typu HEX (moduł CCP ładuje i wykonuje pliki typu COM, natomiast plik typu .HEX jest wykonywany pod nadzorem programu uruchamiania DDT).

- **DDT**, program uruchamiania — zawiera mechanizmy ułatwiające bezpośrednie śledzenie stanu programu w czasie jego wykonywania, np.: T lub Tn — wykonanie jednego lub n rozkazów oraz badanie zawartości rejestrów po każdym kroku; X — opis zawartości rejestrów; G [xxxx] [yyyy] — wykonanie programu od adresu xxxx do adresu yyyy i in.

- **PIP** — służy do przesyłania plików z dysku na dysk, z dysku na inne urządzenie zewnętrzne lub odwrotnie; składnia polecenia jest następująca:

A>PIP <odbiorca> = <nadawca>[opcje] —
 <odbiorca>: : =

CON : konsola operatorska (ekran)
 LST : drukarka
 PRN : drukarka ze stronicowaniem
 PUN : perforator taśmy

[d:]<nazwa> [,<typ>] : plik dyskowy

<nadawca> : : =

CON : konsola operatorska (klawiatura)
 RDR : czytnik taśmy

[d:]<nazwa> [,<typ>] : plik dyskowy

● **SUBMIT (XSUB)** — umożliwia wykonanie ciągu poleceń zebranych w jednym pliku typu .SUB:

A>SUMBIT<nazwa zbioru> [PAR1, PAR2, ...]

przy czym, jeśli w zbiorze poleceń występuje polecenie niezzydentne, to pierwszym poleceniem powinno być XSUB.

● **MOVCPN, SYSGEN** — służą do generacji systemu CP/M.

ZASADY PISANIA PROGRAMÓW UŻYTKOWYCH

Polecenia systemowe powodują realizację tylko podstawowych, najczęściej spotykanych operacji. Dlatego w systemie CP/M jest możliwe, by programy użytkowe komunikowały się bezpośrednio z modułem BDOS, zlecając mu wykonanie elementarnych funkcji. Przyjęte zasady współpracy wymagają umieszczenia kodu funkcji w rC, zaś w rE (gdy wystarcza jeden bajt) albo w rDE (gdy potrzeba dwóch bajtów) — dodatkowych parametrów. Następnie należy wywołać BDOS za pośrednictwem strony zerowej (tab. 1).

Tabela 2. Blok opisu pliku (FCB)

Bajty	Zawartość
0	U — numer jednostki dyskowej, 0 — jednostka domyślna, 1 — jednostka A, 2 — jednostka B itd.
1—8	f — nazwa pliku uzupełniona spacjami
9—11	t — typ pliku i oznaczenie stanu
12	ex — numer wersji rozszerzenia pliku należy zainicjować na 0
13—14	s — zarezerwowane przez system: należy zainicjować na 0
15	rc — liczba rekordów (0—128) rozszerzenia pliku wersji ex
16—31	d — adresy używanych bloków: inicjowane przez BDOS
32	er — bieżący numer rekordu czytanego lub pisanego sekwencyjnie: należy zainicjować na 0
33—35	r — numer rekordu odczytywanego z pliku lub zapisywanego na plik o dostępie bezpośrednim, w celu dotarcia np. do rekordu 258 (102 _H) należy wpisać 02 _H do bajtu 33 i 01 _H do 34

Po wykonaniu operacji system przekazuje informacje do rA lub rHL. Jeśli żądane działania dotyczą plików dyskowych, to rDE musi zawierać adres obszaru FCB (File Control Block) opisującego dany plik (tab. 2). Ogólny schemat wywołania funkcji systemowej w języku symbolicznym wygląda następująco:

[LXI D, <parametr>]
 MVI C, <kod funkcji>
 CALL 0005 ; wywołanie BDOS

W oparciu o podaną metodę, użytkownik może poszerzać zbiór poleceń o własne programy wywoływane bezpośrednio z konsoli. W takich przypadkach warto zadbać o to, aby składnia nowych poleceń była zgodna z umową przyjętą w systemie.

Pełna realizacja programów użytkowych wymaga przekazywania parametrów przez operatora. Umożliwia to moduł CCP odczytujący polecenia i ich parametry. Umieszcza

Tabela 3. Opis funkcji wywoływanych na podstawie kodów

Kod	Działanie
1	2
00	Inicjuje system
01 _H	Czyta znak z klawiatury, interpretuje znaki specjalne, kopluje przeczytany znak na ekran
02 _H	Wypisuje znak na ekranie
03 _H	Czyta znak z czytnika taśmy
04 _H	Wysyła znak do perforatora
05 _H	Wysyła znak na drukarkę
06 _H	Czyta lub pisze znak bez kontroli modułu BDOS: dla (rE) = OFF _H czyta znak z klawiatury, w przeciwnym przypadku zawartość rE jest traktowana jako kod znaku wysyłanego na ekran
07 _H	Czyta IOBYTE ze strony zerowej
08 _H	Ustala IOBYTE
09 _H	Wypisuje na ekranie ciąg znaków zakończony znakiem dolara (\$) Wczytuje ciąg znaków z klawiatury do określonego obszaru; pierwszy bajt obszaru powinien wskazywać maksymalną liczbę oczekiwanych znaków, a drugi — liczbę znaków faktycznie przesłanych (należy go zainicjować na zero); funkcja interpretuje znaki specjalne
0A _H	Określa stan klawiatury; jeśli po wywołaniu funkcji rA=0, to znak z klawiatury jest gotowy do wczytania przez funkcję 01 _H , w przeciwnym przypadku żaden znak nie oczekuje na wczytanie
0B _H	Określa stan klawiatury; jeśli po wywołaniu funkcji rA=0, to znak z klawiatury jest gotowy do wczytania przez funkcję 01 _H , w przeciwnym przypadku żaden znak nie oczekuje na wczytanie
0C _H	W parze rHL przekazuje informację o wersji systemu, (L) = 20 _H oznacza wersję 2.0, (L) = 22 _H — wersję 2.2, rH odróżnia system CP/M (wartość 0) od MP/M (wartość 1)
0D _H	Nadaje dyskom stan R/W oraz inicjuje dysk o numerze domyślnym (np. w przypadku wymiany dysku)
0E _H	Określa domyślny numer jednostki dyskowej, (rE) = 0 oznacza dysk A, (rE) = 1 — dysk B itd.
0F _H	Otwiera plik, którego nazwa musi istnieć w skorowidzu dyskowym; po wykonaniu funkcji wartości (rA) = 0—3 oznacza zbiór znaleziony, (rA) = OFF _H — zbiór nie znaleziony
10 _H	Zamyka plik: zawartość rA jest identyczna jak dla funkcji otwierania
11 _H	Szuka pierwszego opisu w skorowidzu plików (zarezerwowana dla BDOS)
12 _H	Szuka kolejnego opisu w skorowidzu plików (zarezerwowana dla BDOS)
13 _H	Niszczy opis pliku w skorowidzu; po wykonaniu zawartość rA jest identyczna jak dla funkcji 0F _H
14 _H	Czyta sekwencyjnie rekord pliku, każde wywołanie funkcji powoduje zwiększenie o jeden wskaźnik cr (p. tab. 2); przed wywołaniem należy umieścić w pamięci informację o adresie rekordu (funkcja 1A _H); (rA) = 00 oznacza operację wykonaną poprawnie, a (rA) = OFF _H — koniec pliku
15 _H	Zapisuje sekwencyjnie rekord, działa podobnie jak funkcja 14 _H , (rA) = OFF _H oznacza przepelnienie dysku
16 _H	Tworzy nowy plik i otwiera go, (rA) = 00 oznacza poprawne wykonanie operacji, a (rA) = OFF _H — przepelnienie skorowidza plików; przed wywołaniem tej funkcji program powinien sprawdzić, czy plik nie istnieje już na dysku (jeśli tak, to należy go skasować funkcją 13 _H)
17 _H	Zmienia nazwę pliku, pozycja d w bloku FCB musi zawierać nową nazwę i nowy typ pliku, pierwsza część bloku zawiera poprzednią nazwę i poprzedni typ (p. tab. 2.)
18 _H	Określa numery dysków znanych systemowi; wartość 1 kolejnych bitów rH oznacza dołączenie dysków A-H, a wartość 1 kolejnych bitów rL — dołączenie dysków I-P
19 _H	Określa domyślny numer dysku, (rA) = 0 oznacza dysk A, (rA) = 1 — dysk B itd.
1A _H	Dostarcza modułowi BDOS adres obszaru pamięci zawierającego zapisywany lub odczytywany rekord
1B _H	Określa adres tzw. wektora alokacji bloków dyskowych (funkcja zarezerwowana dla systemu)
1C _H	Nadaje stan R/O dyskowi o numerze domyślnym
1D _H	Określa numery dysków o stanie R/O; zawartość pary rHL interpretujemy identycznie jak dla funkcji 18 _H
1E _H	Ustala stan pliku (R/W, R/O, DIR, SYS)
1F _H	Określa adres bloku DPB (parametry dysku) w module BIOS (funkcja wykorzystywana przez system)
20 _H	Ustala lub przydziela plikowi nowy numer użytkownika: jeśli (rE) = OFF _H , to numer jest ustalany, w przeciwnym przypadku zawartość rE jest traktowana jako nowy numer użytkownika
21 _H	Odczytuje wskazany rekord z pliku o dostępie bezpośrednim, numer rekordu należy umieścić na pozycji r bloku FCB — bajty te nie ulegają modyfikacji w czasie czytania: po wykonaniu operacji w akumulatorze A otrzymujemy; (rA) = 00 — operacja poprawna (rA) = 01 — próba czytania niezapisanego bloku (rA) = 03 — niemożliwe otwarcie określonej wersji pliku (rA) = 04 — próba dostępu do niezapisanej wersji pliku (rA) = 06 — fizyczny koniec dysku
22 _H	Zapisuje wskazany rekord na plik o dostępie bezpośrednim; wykonywana jest podobnie jak funkcja 21 _H , przy czym dodatkowo —

1	2
	(rA) = 05 oznacza niemożliwe rozszerzenie pliku (przepelnienie skrowidza)
23 _H	Umieszcza liczbę rekordów danego pliku w bajtach r tablicy FCB
24 _H	Określa bieżący numer rekordu w pliku, tzn. umożliwia przejście od czytania lub pisania sekwencyjnego do czytania lub pisania zadanego rekordu
25 _H	Inicjuje jednostki dyskowe, numery dysków są określone kolejnymi bitami pary rDE (bit 0 oznacza jednostkę A, bit 1 — jednostkę B itd.)
28 _H	Działa identycznie jak funkcja 22 _H , a ponadto zeruje każdy nowy blok

on (bez żadnych modyfikacji) wartości parametrów w kolejnych bajtach strony zerowej (tab. 1.), począwszy od adresu 81_H. W komórce 81_H zapamiętywana jest liczba istotnych znaków. Ogranicznikiem obszaru parametrów jest bajt zerowy. Moduł CCP, traktując pierwszy parametr jako nazwę pliku, tworzy 16 pierwszych bajtów bloku FCB począwszy od adresu 5C_H. O ile nie jest to koniec listy parametrów, to tworzy się znowu 16 bajtów dla drugiego FCB (od adresu 6C_H).

Programy użytkowe mogą przekazywać sterowanie trzema sposobami:

— rozkazem *RET*
 — skokiem do adresu zerowego (*JUMP 0*)
 — wywołaniem funkcji *OO (CALL OO)*.
 Ostatnie dwa sposoby powodują ponowne zainicjowanie systemu.

W tabeli 3 przedstawiono 39 funkcji, których wywołanie następuje na podstawie kodów.

* * *

W następnym numerze zostanie przedstawiona organizacja dysków, sposób generacji, a także — opinie na temat użyteczności systemu CP/M.

LITERATURA

- [1] Pinto I.: Systèmes d'exploitation: introduction à CP/M, un standard de l'industrie du micro. Minis et Micros, nr 166, 1982
- [2] Pinto I.: Système d'exploitation: CP/M et ses commandes résidentes, son assembleur et son éditeur. Minis et Micros, nr 167, 1982
- [3] Pinto I.: Système d'exploitation: CP/M et ses commandes (II). Minis et Micros, nr 168, 1982
- [4] Pinto I.: Système d'exploitation CP/M et les interfaces programmeur. Minis et Micros, nr 169, 1982

Cykl, w którym od ponad roku prezentujemy języki programowania (choć na razie były to tylko ADA i LOGLAN — pozostałe już czekają na swoją kolej), postanowiliśmy rozszerzyć o systemy operacyjne. Mamy nadzieję, że ta innowacja (jak również cały cykl) spotka się z zainteresowaniem Czytelników, tym bardziej, że pierwszym prezentowanym systemem operacyjnym jest CP/M — związany tematycznie z treścią całego numeru i uważany obecnie za faktyczny standard dla systemów mikrokomputerowych. Jest to pierwsza publikacja, jaka ukazuje się na ten temat w języku polskim.

CP/M (Control Program for Microcomputer) istnieje prawie tyle lat, ile liczy sobie mikrokomputer. W 1974 roku kalifornijczyk Gary Kildall, właściciel jednego z pierwszych mikrokomputerów (IMSAI), postanowił stworzyć dla swojego sprzętu system operacyjny. Przedsięwzięcie to wydaje się nam dzisiaj oczywiste, ale w owym czasie był to szalony pomysł. Wówczas każdy użytkownik musiał sam pisać programy obsługujące urządzenia zewnętrzne; programy przechowywano na taśmach perforowanych (nośnikach tanich, ale bardzo niewygodnych), czasami na taśmach magnetycznych. Dołączenie dysków elastycznych było więc doskonałym pomysłem. Obecnie, system CP/M jest rozpowszechniany głównie przez

firmę DIGITAL RESEARCH (Pacific Grove, California), którą założył Gary Kildall.

Prawdziwy sukces handlowy zawdzięcza jednak CP/M panu Tony'emu Goldowi, który założył w Nowym Jorku specjalny punkt sprzedaży (LIFEBOAT ASSOCIATES) programów pracujących pod nadzorem systemu CP/M, a także licznych odmian systemu przystosowanych do prawie wszystkich mikrokomputerów istniejących na rynku (opartych na mikroprocesorze INTEL 8080 lub Z80). Tony Gold założył również specjalny klub użytkowników CP/M: CPMUG. Na ogromny sukces systemu CP/M złożyły się więc nie tylko jego zalety, ale również dwa inne składniki: potrzeba oprogramowania podstawowego dla mikrokomputerów oraz pomysłowa i energiczna polityka handlowa.

W opublikowanej ostatnio bibliografii systemu CP/M Operating System: January 1978 — February 1982. Report PB82-861501. National Technical Information Service, Springfield, Virginia, February 1982) zawarto kilkadziesiąt pozycji. Popularny opis systemu znajduje się w łatwo dostępnej pracy: POWYS-LYBBE D., CP/M Operating System — the Software Bus, Microprocessors and Microsystems, Vol. 5, No. 3, 1981.

REDAKCJA

KONFERENCJE

Symposium IMACS

Międzynarodowe stowarzyszenie zastosowań matematyki i komputerów do symulacji procesów IMACS (International Association for Mathematics and Computers in Simulation) przy współpracy ze stowarzyszeniem informatyków francuskich AFCET oraz państwami wyższą szkołą mechaniki Nantes ENSM organizują w dniach 9—11 maja 1983 r. w Nantes (Francja) międzynarodowe sympozjum poświęcone metodom symulacyjnym stosowanym w automatyzacji procesów mechanicznych i energetycznych. Program sympozjum będzie obejmował cztery sesje plenarne poświęcone:

- obliczeniom równoległym
- oprogramowaniu do obliczeń symulacyjnych

- symulowaniu aspektów energetycznych, ekonomicznych i ekologicznych procesów przemysłowych
- symulowaniu przepływów.

Przewiduje się wygłoszenie ok. 60 referatów, dotyczących metod i narzędzi symulacji oraz sprzętu, oprogramowania i zastosowań symulacji w różnych dziedzinach nauki i techniki.

W czasie seminarium będą odbywać się pokazy działania sprzętu i oprogramowania symulacyjnego. Po dalsze informacje i szczegółowy program sympozjum można zwracać się pod następujący adres: AFCET — 156 Bld Péreire — F. 75017 Paris.

Operacje na liczbach całkowitych wielokrotnej precyzji

Po rocznej z górą obecności ALGORYTMÓW na łamach INFORMATYKI, postanowiliśmy wypróbować nową formułę tej rubryki. Poprzednim naszym założeniem było skondensowanie opisu i analizy algorytmów oraz przedstawienie w całości gotowych i sprawdzonych procedur. Począwszy od poniższego artykułu przyjmujemy swobodniejszą formę, kładąc nacisk na prezentację sposobu działania algorytmów i elementy ich analizy. Ze względu na ograniczoną objętość rubryki, rozszerzenie części opisowej musi pociągnąć za sobą rezygnację z zamieszczania gotowych procedur. Niemniej jednak postaramy się stosować taką notację, aby możliwe było zarówno szybkie zrozumienie zasad działania algorytmów, jak i napisanie procedur w jakimkolwiek języku programowania.

Nie chcemy podawać teraz szczegółów takiej notacji, bowiem dostosowywać ją będziemy do konkretnych problemów. Na przykład do opisu algorytmu szybkiego sortowania (ang. quick sort) czy operacji na B-drzewach najodpowiedniejsza wydaje się nam notacja rodem z książki Aho, Hopcrofta, Ullmana „The Design and Analysis of Computer Algorithms”, natomiast do obliczeń statystycznych — opis w języku zbliżonym do naturalnego. Nie będziemy oczywiście unikać gotowych procedur, o ile będą wystarczająco przejrzyste.

Poniżej prezentujemy pierwsze algorytmy w uproszczonej notacji. Prosimy czytelników o ustosunkowanie się do tej propozycji.

W bieżącym numerze prezentujemy algorytmy działań na liczbach całkowitych wielokrotnej precyzji, tzw. liczbach długich. Liczby takie to ciąg cyfr w układzie o pewnej podstawie (jej wielkość zależy od dokładności maszyny, na której algorytmy zostaną zaimplementowane). W dalszym ciągu będziemy oznaczać całą liczbę przez pojedynczą literę, a jej k -tą cyfrę przez literę z indeksem k . Przykładowo — niech podstawa wynosi 100. Wówczas liczba $a' = 1983$ ma następujące cyfry: $a_1 = 19$ i $a_2 = 83$. Przez literę z indeksem 0 oznaczać będziemy sumę cyfr w liczbie (w powyższym przykładzie $a_0 = 2$). Zakładamy też, że operujemy wyłącznie liczbami dodatnimi, modyfikacja podanych algorytmów — tak, aby uwzględnione były liczby ujemne jest prosta, a tu zaciemniałaby jedynie postać algorytmu.

MOTYWACJA

W poprzednich artykułach w tej rubryce poruszaliśmy problemy wyszukiwania informacji i sortowania. Zakładaliśmy w nich, że klucze są liczbami całkowitymi, gdy tymczasem w rzeczywistości zazwyczaj są one bardziej skomplikowane. Jednakże każdy klucz (np. alfanumeryczny) daje się przedstawić w postaci odpowiednio dużej liczby całkowitej. Niestety, większość komputerów nie daje możliwości dokonywania obliczeń na takich liczbach. Przedstawione tutaj rozwiązania umożliwią być może praktyczne zastosowanie tak ciekawych algorytmów, jak sortowanie dystrybucyjne czy przechowywanie informacji w tablicach rozproszonych.

Ponadto w normalnych obliczeniach użytkownik może się również spotkać ze zbyt małą dokładnością maszyny. W bieżącym numerze gros artykułów poświęconych jest mikroprocesorom, gdzie użytkownik może zazwyczaj używać co najwyżej liczb 16-bitowych, co często jest dla niego niewystarczające.

ALGORYTMY

Przedstawimy następujące algorytmy:

- porównywanie liczb
- dodawanie dwóch liczb
- odejmowanie
- mnożenie
- dzielenie całkowite (wraz z liczeniem reszty z dzielenia)
- potęgowanie.

Zakładamy, że liczby są przedstawione w postaci opisanej we wstępie, a zmienna p ma wartość podstawy zapisu.

Porównywanie

Procedura porównująca dwie liczby „długie” A i B daje w wyniku 1 jeżeli $A > B$, 0 — jeżeli $A = B$ i -1 — jeżeli $A < B$.

Porównaj:

```
begin
  for i := 0 to A0 do
    begin
      if Ai < Bi then begin Porównaj := -1; return end;
      if Ai > Bi then begin Porównaj := 1; return end;
    end;
  Porównaj := 0; return
end;
```

Dodawanie

Procedura *dodaj* ma jako argumenty dwie liczby długie A i B , a wynikiem jest $C = A + B$. Dla uproszczenia zapisu założyliśmy, że $A_0 = B_0$.

Dodaj:

```
begin carry := 0
for i := A0 downto 1 do
  begin
    x := Ai + Bi + carry;
    if x > p then begin x := x - p; carry := 1 end
    else carry := 0;
    Ci+1 := x
  end
end;
```

Odejmowanie

Odejmowanie liczb „długich” A i B zakłada, że $A > B$ i że $A_0 = B_0$.

Odejmij:

```
if Porównaj(A,B) = 1 then
  begin
    for i := A0 downto 1 do
      begin
        if Ai < Bi then begin
          Ai-1 := Ai-1 - 1;
          Ai := Ai + p
        end;
        Ci := Ai - Bi
      end;
    end;
```

Mnożenie

Poniżej przedstawiamy najprostszy algorytm mnożenia dwóch liczb całkowitych, mimo że znane są algorytmy teoretycznie (p. D. Knuth „The Art of Computer Programming” vol. 2, p.4.3.2). Sądzymy bowiem, że suma cyfr prak-

tycznie stosowanych liczb będzie na tyle mała, że dodatkowe koszty związane z szybkimi algorytmami udaremnią sensowność stosowania (np. przy podstawie 2^{15} ciąg liter i spacji długości 50 znaków daje się zapamiętywać w liczbie o 16 cyfrach).

Pomnóż:

```
for i:=B0 + 1 to B0 + A0 do Ci :=0;
for j:=B0 downto 1 do
begin
  if Bj=0 then Cj :=0
  else
    begin carry :=0;
      for i:=A0 downto 1 do
        begin
          t :=Ai * Bj + Ci+j + carry;
          Ci+j := t mod p;
          carry :=t div p
        end;
      Cj :=carry
    end;
end;
```

Dzielenie

W przypadku dokonywania operacji dzielenia należy osobno rozważyć przypadek, gdy dzielnik ma tylko jedną cyfrę. Wówczas algorytm dzielenia ma postać następującą:

```
Podziel: {C :=A div B; R:=A mod B}
R:=0;
for j :=1 to A0 do
begin
  Cj := (R*p +Aj) div B1 ;
  R := (R*p +Aj) mod B1 ;
end;
```

Jeżeli natomiast dzielnik ma więcej cyfr, to należy stosować bardziej złożony algorytm:

```
Podziel: {C:= A div B;R:= A mod B i niech A0 = m + n
          i B0=n}
d:= p div B1 + 1
```

```
(U1 ,..., Um+n+1) :=d*(A1 ,..., Am+n)
for j:=1 to m+1 do
begin
  if Uj=B1 then x:=p-1
  else x:= (Uj*p+Uj+1) div B1 ;
  while B2x>(Uj*p+Uj+1-x*B1)*p+Uj+1 do x:=x+1 ;
  t:=Uj-1 ;
  (Uj-1 ,..., Uj+n):=(Uj-1 ,..., Uj+n)-(B1 ,..., Bn)*x;
  Cj :=x;
  if t≠Uj-1 then
    begin
      Cj :=Cj-1;
      (Uj-1 ,... Uj+n) :=(Uj-1 ,..., Uj+n)+(B1 ,..., Bn); (*)
    end
end;
```

W miejscach zaznaczonych (*) zastosowaliśmy zapis symboliczny, aby zwiększyć czytelność algorytmu.

Potęgowanie

Przedstawiamy poniżej algorytm podnoszenia liczby A do potęgi B. Dla jasności zapisu używamy klasycznych oznaczeń działań, które należy oczywiście zamienić na wywołania odpowiednich procedur. Aczkolwiek poniższy algorytm ma już ponad 500 lat, to jednak wydaje się być optymalnym (p. D. Knuth „The Art of Computer Programming” vol. 2 p. 4.5.3)

Potęguj:

```
C :=1 ;
Z :=A
while b≠0 do
begin
  r :=B mod 2 ;
  B :=B div 2 ;
  if r=1 then C :=C*Z ;
  Z :=Z*Z
end ;
```

ANDRZEJ SZALAŚ
ZBIGNIEW SWIRSKI

SPROSTOWANIE. W poprzednim artykule tej rubryki (nr 1/83) przedstawione zostały dwa pierwsze rysunki. Przepraszamy. (Red.)

GRY

Polowanie

Kontynuując cykl prezentowania opisów gier komputerowych, chcielibyśmy przedstawić „Polowanie na wumpusy”. Wumpus jest to groźny potwór jaskiniowy, który między innymi żywi się ludźmi. Obecnie żyje on w jaskini na terenie Mordoru (J. R. R. Tolkien „Władca pierścieni”). W jaskini jest 20 grot. Każda z nich jest połączona z trzema innymi (połączenia mogą być stałe bądź generowane losowo przy każdym użyciu gry).

Poza Tobą i wumpusem w jaskini są:
— bezdenne dziury
— supernietoperze.

Jeżeli wejdiesz do groty z bezdenną dziurą, to wpadasz do tej dziury i oczywiście — przegrywasz. Jeżeli natomiast spotkasz w swojej wędrówce supernietoperza, to porywa Cię on i przenosi do innej (losowej) groty (być może z np. bezdenną dziurą).

Wejście do groty, w której jest wumpus, kończy się zawsze jednoznacznie... (pamiętaj! wumpus żywi się ludźmi). Bezdenne dziury nie są groźne dla wumpusa, bo ma on przyssawkowe łapy. Również supernietoperz nie zagraża mu, bo wumpus jest za ciężki dla niego. O wszystkich niebezpieczeństwach jesteś ostrzegany jedną lub dwie grotę wcześniej (np. komunikatem „czujesz zapach wumpusa”).

W każdym ruchu możesz albo przejść do jednej z trzech sąsiednich grot albo strzelić zatrutą strzałą. Strzała może przelecieć najwyżej przez pięć grot, jeżeli podasz numery tych grot i są one ze sobą połączone. Jeżeli nie ma odpowiednich przejść, to strzała odbija się od ściany, w losowym kierunku (może Ciebie trafić!).

Wygrwasz, jeżeli strzała trafi w wumpusa. Dodatkowym ograniczeniem jest to, że masz tylko pięć strzał. Wumpus jest leniwy i na ogół siedzi w jednej grocie (np. tej, w której jesteś, a wtedy...).

Zyczymy przyjemnego polowania.

Z. SWIRSKI

GIEŁDA INFORMACJI

O MINIKOMPUTERACH 16-BITOWYCH

Centrum Szkolenia Informatycznego (CSI) przy ZETO-Łódź organizuje w terminie 30.05—01.06. 1983 r. seminarium pn. „Wybrane problemy eksploatacyjne minikomputerów 16-bitowych”. Organizatorzy zapraszają do udziału w seminarium zainteresowanych stosowaniem: MERY-60, SM-3, SM-4 i PDP/11 (programiści, użytkownicy, obsługa techniczna).

Celem seminarium jest wymiana doświadczeń uzyskanych w procesie eksploatacji ww. sprzętu, a także umożliwienie integracji zwolenników jego zastosowań. Termin przyjęć referatów do wygłoszenia na seminarium mija 15 kwietnia br. Ich autorzy zaproszeni są na koszt organizatora.

Obrazy odbędą się w Ośrodku Szkoleniowo-Wypoczynkowym ZETO-Jelenia Góra w Jagniątkowie k. Jeleniej Góry, ul. Myśliwska 14. Koszt uczestnictwa wynosi 2800 zł od osoby. Kwotę tę należy wpłacić na konto: ZETO-Łódź NBP I O/M Łódź nr 47018-2219 i o tym fakcie powiadomić CSI pismem z imiennym zgłoszeniem.

Adres do korespondencji. 90-558 Łódź, ul. Hutora 69, tel. 647-70.

GEORGE-3

Centrum Informatyki Gospodarki Morskiej w Gdańsku oferuje użytkownikom systemu operacyjnego GEORGE-3 edytor taśm magnetycznych. Umożliwia on edycję prostych zbiorów formatu taśmy magnetycznej na taśmach fizycznych i w PZS-ie. Jest on szczególnie łatwy w użyciu, ze względu na swe duże podobieństwo do komendy EDIT w systemie GEORGE-3. Dostępna jest większość instrukcji komendy EDIT z zachowaniem tej samej nomenklatury. Dodatkowo możliwa jest edycja oznaczników użytkownika. Edytor taśm magnetycznych można przystosować do pracy bezpośrednio pod kontrolą programu nadzorczego EXECUTIVE.

Bliższych informacji udziela: Główny Technolog, Centrum Informatyki Gospodarki Morskiej, 80-890 Gdańsk, ul. Heweliusza 11, pok. 203, tel. 31-27-86.

INFORMATOR KONWERSACYJNY

Institut Organizacji Zarządzania Politechniki Warszawskiej oferuje system programów pozwalających na utrzymanie i łatwy konwersacyjny dostęp do hierarchicznie zorganizowanych zbiorów informacji tekstowej. INFORMATOR przeznaczony jest do użytkowania pod kontrolą systemu operacyjnego GEORGE-3. Główną jego cechą jest łatwość użytkowania (prosty język komend), która powoduje, że nawet mało wprawni użytkownicy są w stanie szybko

Z KRAJU

Mikroprocesorowe Szkoły Zimowe

Refleksje współorganizatora

W ramach działań przygotowujących potencjalnych klientów CEMI do użytkowania systemów mikroprocesorowych (p. [1]), w marcu 1980 r. rozpoczęto szkolenia, które nazwano Mikroprocesorową Szkołą Zimową. Zgodnie z zamierzeniami organizatorów, a ku pewnemu zaskoczeniu autora, udało się uczynić ze Szkół imprezę cykliczną. Obecnie mamy za sobą trzy szkoły [3—5], a czwarta, poświęcona systemom wielomikroprocesorowym i komputerom jednokładowym, została zaplanowana na styczeń 1983.

Formułując założenia organizacyjno-programowe Szkół Mikroprocesorowych wzorowano się na niektórych doświadczeniach w przebiegu spotkań zwanych *workshops*, które — zdaniem autora i współorganizatorów — zdaly egzamin. Ponieważ jednocześnie należało je nieco zmodyfikować, dostosowując do krajowych warunków (a podobny typ szkolenia nie jest w Polsce zbyt popularny), uznaliśmy, że celowe będzie zapoznanie czytelników INFORMATYKI z efektami prac Szkół.

Założenia organizacyjno-programowe

W prospekcie informacyjnym Mikroprocesorowej Szkoły Zimowej, w roku 1980, napisano:

„Przedsięwzięcie to jest próbą stworzenia forum aktywnej wymiany doświadczeń osób, które mają już własne osiągnięcia w dziedzinie projektowania i programowania systemów mikroprocesorowych.

Hasłem tegorocznej szkoły są narzędzia sprzętowe i programowe wspomagające procesy projektowania systemów mikroprocesorowych.

Kilka ośrodków krajowych ma w tej dziedzinie znaczące osiągnięcia; chcemy się przekonać, w jakim stopniu wymiana doświadczeń pozwoli usprawnić rozwój tych ważnych badań i prac, i na ile uda nam się przez połączenie wysiłków usprawnić proces przekazywania wyników do szerokiej praktyki społecznej. Obecnie obserwuje się proces intensywnego wzrostu zainteresowania rzeszy inżynierów mikroprocesorami i ich zastosowaniem.

(...) Chcielibyśmy, podczas Szkoły Zimowej, mieć czas na intensywną, lecz spokojną pracę. Dlatego przeznaczamy na ten cel dwa tygodnie. Narzędzia sprzętowe i programowe będą w czasie trwania Szkoły dostępne dla każdego uczestnika, aby mógł pracować za ich pomocą, rozwiązując swoje własne problemy.

To ogranicza oczywiście liczbę uczestników. Nie możemy przekroczyć liczby 30 osób, przy czym zakładamy, że

połowa to będą osoby prezentujące w czasie trwania szkoły swoje wyniki”.

Drugą Szkołę zorganizowano po niepełnej 10 miesiącach, tym razem wspólnie z Centrum Szkolenia Informatycznego ZETO w Łodzi. Wiedzieliśmy, że można liczyć na uczestników poprzedniej Szkoły, którym ten typ spotkań spodobał się, ale było też jasne, że przyjadą również inni. Na podstawie wielu głosów krytycznych ustaliliśmy, że czas trwania pierwszej szkoły (dwa tygodnie) był stanowczo za długi i tym razem skrócono go do trzech dni.

Założenia tej Szkoły brzmiały następująco: „Mikroprocesorowa Szkoła Zimowa 1981 ma być spotkaniem dwóch grup ludzi: mających już kilkuletnie doświadczenie w projektowaniu, stosowaniu i nauczaniu mikrosystemów oraz tych, którzy do tej dziedziny dopiero wchodzą. Naczelnym celem szkoły jest zapoczątkowanie procesów społecznych integrujących ludzi, zajmujących się mikrosystemami, z uwzględnieniem wszystkich pozytywnych i negatywnych zjawisk, jakie wystąpiły w „klasycznej informatyce” w ostatnim trzydziestolecu jej istnienia. Równorzędnym celem Szkoły jest wzajemne poinformowanie się twórców mikroinformatyki o problemach, nad którymi pracują, zminimalizowanie dublowania prac intelektualnych i stworzenie trwale działających więzów informacyjnych między grupami mikroinformatyków, które w ciągu najbliższych lat pojawią się na terenie całej Polski. Celem przygotowawczym jest przysposobienie się do nawiązania znacznie szerszych więzów informacyjnych, obejmujących także młodzież szkolną i przedszkolną, a również nie ograniczających się do terenu naszego kraju”.

Jedno z zamierzeń przeczyło zwyczajowi rocznego lub dwuletniego cyklu odbywania konferencji. Szkoły Zimowe miały być organizowane dwukrotnie w ciągu roku: pierwsza — w styczniu, lutym lub marcu, druga (BIS) — w grudniu. W ten sposób dzieliby je okres 8—9 miesięcy. Uczestnicy referujący wstępne prace lub zamiary, na początku roku, mieliby później okazję zapoznać swoich „starych znajomych” z postępem prac. Wychodząc z tych przesłanek Mikroprocesorową Szkołą Zimową 1981-BIS zaplanowano na 15—17 grudnia 1981 r. Ze względu na rygory stanu wojennego — została przełożona i odbyła się w dniach 4—6 maja 1982 pod nazwą III Szkoły Mikroprocesorowej (bo już nie bardzo była „zimowa”).

Przed nami IV Szkoła Mikroprocesorowa (MSZ 1983), zapowiadana następująco:

„Kolejna Szkoła odbywać się będzie w czasach, gdy w światowych czasopiśmie fachowych z naszej dziedziny

jednym z dominujących tematów stały się systemy wieloprocesorowe i wielomaszynowe. Szczególną rolę w ich rozpowszechnieniu odegrały mikroprocesory, które koszty realizacji takich systemów zmniejszyły (jak chyba można szacować) o ponad rząd wielkości. Aby więc próbować nadążyć za tym wyścigiem do doskonałości „informatyko-przetwórczej” — postanowiliśmy Mikroprocesorową Szkołę Zimową 1983 poświęcić temu zagadnieniu. Ponieważ, jak Państwo pamiętają, jest zasadą Szkoły, że prezentujemy prace autorskie — musieliśmy trochę poczekać. Ale chyba dobrze, bo obecnie dysponujemy już zarówno poważnymi opracowaniami naukowymi, jak i pierwszymi, ciekawymi implementacjami wielosystemów mikroprocesorowych”.

Tematyka

Od pierwszej Szkoły podjęto wysiłki, aby każda z nich miała określony profil. Starano się wyodrębnić problemy, które na danym etapie rozwoju mikroinformatyki mogą mieć największą wartość dla uczestników Szkoły i — starannie dobierano autorów. Zawsze obowiązywała zasada, że nie wygłasza się referatów przeglądowych czy relacji „z drugiej ręki”, lecz — przedstawia to, co jest zrobione i jednocześnie może zainteresować.

Profil MSZ 1980

- Projektowanie systemów mikroprocesorowych sposobem modelowania strukturalizowanego. Przykłady stosowania tej metody. Rola środków sprzętowych typu MDS w metodzie. Perspektywy rozwoju metody.
- Systemy wspomagające proces projektowania i programowania systemów mikroprocesorowych. Osiągnięcia krajowe — perspektywy standaryzacji.
- Narzędziowe oprogramowanie skrośne. Sytuacja krajowa. Minimalne wymagania standaryzacyjne. Możliwości i środki wymuszenia standaryzacji.
- Biblioteka programów mikroprocesorowych B.I.T.E.M. Sieć B.I.T.E.M. Pierwsze wyniki. Perspektywy.

Profil MSZ 1981

- Sieć B.I.T.E.M i jej treść oraz status prawno-organizacyjny i finansowy.
- Zrealizowane (własne) mikrosystemy.
- Pakiety symulacyjne na mikroprocesorach: SIM.LIB oraz RT-SIMULA.
- Sprzętowo-programowe narzędzia wspomagające projektowanie i programowanie mikrosystemów.
- Sposób organizowania działań zastosowanych w informatyce.

Profil MSZ 1982

- Języki programowania wysokiego poziomu dla mikroprocesorów.
- Mikrokomputery jednokładowe serii 8048 i 8041.
- Biblioteka B.I.T.E.M.

Profil MSZ 1983

- Systemy wielomikroprocesorowe.
- Mikrokomputery jednokładowe.
- Pokazy programów z B.I.T.E.M i mikrosystemów autorskich.

Wyniki

Autorowi trudno zachować obiektywizm w formułowaniu wniosków, ze względu na silne zaangażowanie emocjonalne w sprawę Szkoły Mikroprocesorowej. Uważając jednak, że może ktoś inny zechce zabrać głos na temat wyników Szkół uzupełniając tę wypowiedź lub z nią polemizując, spróbuję przedstawić kilka osobistych spostrzeżeń.

- Szkoły Mikroprocesorowe stały się imprezą cykliczną. Skoro odbyły się już trzy, to należy przewidywać, że dobijemy przynajmniej do dziesiątku.
- Mamy już swoje miejsce — Łódź i gościnne gospodarza — Centrum Szkolenia Informatycznego przy ulicy Cz. Hutora 69. Blisko jest Hotel „Mazowiecki”. Jeśli nie zmienimy miejsca i gospodarza — a jeśli to będzie ode mnie zależało, to nie zmienimy — unikniemy nieszczęść, które spotykają uczestników konferencji odbywających się w miejscowościach atrakcyjnych turystycznie.
- Dwadzieścia do trzydziestu osób, które aktywnie działają w tej dziedzinie, spotkało się już trzykrotnie. Znamy swoje adresy i telefony, wiemy co kto robi i jak go odszukać. Trzeba poczekać jeszcze pięć lat, aby wszystkie pozytywne skutki takich regularnych spotkań mogły się objawić.
- Tym kilkunastu ludziom, którzy zawsze mają coś do powiedzenia w dziedzinie informatyki przygląda się zawsze co najmniej kilkanaście „nowych twarzy”, nie powinny więc występować efekty „gruśnienia” (groupthink) [2].
- W tematyce naszych spotkań, która uwzględniała wiele spraw istotnych dla informatyki, doszliśmy do mikroprocesorowych systemów wielokomputerowych i wieloprocesorowych. Goniśmy więc tego elektronicznego zająca, chyba nie najgorzej — uwzględniając, że mówimy zawsze o własnych pracach.

JERZY DAŃDA

LITERATURA

- [1] Dańda J.: Oprogramowanie systemów mikroprocesorowych — praktyczne aspekty polityki producenta półprzewodników. INFORMATYKA, nr 1/83
- [2] Janis J. L.: Victims of groupthink. Houghton-Mifflin, Boston, 1972
- [3] Mikroprocesorowa Szkoła Zimowa 1980. Materiały. ZOINTE — Instytut Technologii Elektronowej NPCP, Warszawa, 1980
- [4] Mikroprocesorowa Szkoła Zimowa 1981. Materiały. ZOINTE — Instytut Technologii Elektrycznej NPCP, Warszawa, 1981
- [5] III Szkoła Mikroprocesorowa, Materiały seminarium. Centrum Szkolenia Informatycznego ZETO, Łódź, 1982.

GIEŁDA INFORMACJI

uzyskać potrzebną informację, a po kilku kontaktach z informatorem błędnie trafiają do żadanego hasła.

Zbiory informacyjne (katalogi i hasła) mogą być łatwo tworzone, modyfikowane i usuwane za pomocą standardowych funkcji GEORGE-3. Zbiory informacyjne są oferowane razem z oprogramowaniem.

Informator może być wykorzystany do rozpowszechniania wśród użytkowników informacji bieżących, a także zastosowany jako część składowa systemów przetwarzania danych — jeżeli występuje potrzeba szybkiego udostępniania informacji (np. o typach transakcji, poprawnych formatach komend, postaciach wydawnictw).

Blizszych informacji udzielają: mgr Piotr Zajęczkowski i mgr inż. Dariusz Adamowski, IOZ PW, ul. Narbutta 85, 02-521 Warszawa, tel. 49-98-71 w. 569. Informator można obejrzeć w Instytucie od poniedziałku do piątku w godz. 8.30—15.30.

GEORGE-3 - instalacja i wdrażanie

ZETO-Wrocław oferuje swoje usługi przy instalowaniu i wdrażaniu do eksploatacji systemu operacyjnego GEORGE-3. Biorąc pod uwagę sprzęt i wymagania zgłaszającego się ośrodka, zapewnia:

- opracowanie projektu systemu
- generowanie systemu operacyjnego
- instalowanie Pamięci Zbioru Systemu (PZS)
- uruchomienie pracy lokalnej
- testowanie systemu
- uruchomienie pracy zdalnej (MOP)
- opracowanie dokumentacji systemu
- szkolenie (operatorów, projektantów i programistów, kierowników instalacji)

Zlecając ZETO-Wrocław wdrożenie systemu GEORGE-3, zleciodawca uzyskuje w okresie ok. 1 miesiąca (w tym szkolenie) instalację gotową do normalnej eksploatacji, dysponującą sprawdzonym oprogramowaniem systemowym (użytkowym) oraz przeszkoloną kadrą.

Wdrożenie systemu GEORGE-3 przez ZETO-Wrocław wymaga od zamawiającego:

- posiadania sprawnej technicznie instalacji
- posiadania nośników s.o. GEORGE-3 i egzekutorów EGRM i EWGN (ZETO-Wrocław może — o ile uzyska odpowiednie upoważnienie — pośredniczyć w uzyskaniu poprawnych wersji systemu GEORGE-3 i egzekutora EWGN)
- udostępnienia instalacji na prace wdrożeniowe (ok. 20 godz.)
- pokrycia kosztów delegacji
- współpracy z zespołem wdrażającym, zarówno na etapie przygotowań, jak i w trakcie wdrażania.

Zgłoszenia przyjmuje kierownik Działu Marketingu, mgr Tadeusz Czerniewski: ZETO-Wrocław, ul. Ofiar Oświęcimskich 7/13, 50-069 Wrocław; tel. centr. 445-431 do 37; telex 0712533 ZETO PL.

Najnowsze tendencje w dziedzinie komputerowych systemów sterowania

Przy bardzo utrudnionym dostępie do literatury fachowej, warto omawiać prace konferencyjne¹⁾, nawet jeśli istnieje obawa, że omówienie dotrze do Czytelników z rocznym opóźnieniem. Jest to uzasadnione tym bardziej, że w 1981 roku odbył się Światowy Kongres Międzynarodowej Federacji Sterowania Automatycznego (IFAC), organizacji odgrywającej wiodącą rolę na arenie międzynarodowej²⁾ w dziedzinie komputerowych systemów sterowania (Kyoto, Japonia, 24—28 sierpnia).

Choć na ogół, w tzw. latach kongresowych (Kongresy IFAC odbywają się co trzy lata) nie organizuje się innych konferencji pod patronatem tej organizacji, w roku 1981 uczyniono wyjątek od tej reguły organizując towarzyszące Kongresowi dwa spotkania robocze, tzw. *workshops*, ściśle związane z interesującą nas tematyką:

● *3rd IFAC Workshop on Distributed Computer Control Systems (DCCS)*, w Pekinie, 15—17 sierpnia

● *11th IFAC/IFIP Workshop on Real Time Programming (RTP)*, w Kyoto, 31 sierpnia — 2 września

Kongres IFAC był molochem, na którym przedstawiono ponad 600 referatów. W takiej masie prac niezmiernie trudno jest uchwycić właściwe tendencje, tym bardziej, że te najnowsze zostały bardzo słabo zarysowane. Wydaje się, że oba spotkania towarzyszące Kongresowi (tzn. DCCS i RTP) były pod tym względem bardziej czytelne. Tematyka tych spotkań odpowiada naturalnemu podziałowi zagadnień na sprzętowe i programowe. Utrzymując ten podział, omówimy także bardzo ważny, wyłaniający się obecnie problem — udział człowieka w komputerowych systemach sterowania oraz skutków, jakie przynosi człowiekowi ta nowa technika pracy. Na zakończenie przedstawimy — na przykładzie Japonii — zastosowania komputerowych systemów sterowania.

SYSTEMY ROZŁOŻONE PRZESTRZENNIE

Jest oczywiste, że prezentację nowych tendencji w budowie sprzętu należy rozpocząć od mikroprocesorów. Niemniej mikroprocesory, same w sobie, niczego nie znaczą; istotne są ich funkcje — pojedyncze, jako jednostek

wbudowanych w inne urządzenia (pełniące rolę części cyfrowej), lub bardziej złożone, w zestawach wieloprocessorowych, najczęściej rozłożonych przestrzennie.

Według H. Hallinga, RFN (RTP), obecnie nie ma większych systemów sterowania, które nie byłyby rozłożone przestrzennie. Mówiąc o sprzętowej stronie zagadnienia, należy rozważyć kilka elementów takiego systemu różniących się funkcjonalnie, a więc: część sprzęgającą z obiektem, sprzęgającą z operatorem, bazę danych oraz magistralę komunikacyjną.

Wprawdzie ciągle aktualnym problemem jest standaryzacja urządzeń sprzęgających komputery z obiektem i operatorem, jednak istotniejsza dla systemów rozłożonych przestrzennie jest wzajemna komunikacja zestawów. Prace nad normalizacją w tym zakresie, choć rozpoczęte stosunkowo niedawno, są prowadzone bardzo intensywnie.

Najważniejszy, ze względu na możliwe konsekwencje po jego zatwierdzeniu, jest tzw. projekt IEEE P-802, dotyczący lokalnych sieci komputerowych — *Local Area Network Standard* (T. J. Harrison, G. J. Clancy, USA, DCCS). Istota tego projektu zawiera się w opracowaniu zaleceń normalizacyjnych dla dwóch najniższych poziomów tzw. siedmiopoziomowego modelu ISO/OSI (ang. *open system interconnection*³⁾), tzn. poziomu fizycznego (ang. *physical level*) i poziomu łącza (ang. *data link level*). W projekcie P-802 poziomowi łącza odpowiadają dwie warstwy: sterowanie łączem logicznym (ang. *logical link control*) i sterowanie dostępem do ośrodka transmisji (ang. *media access control*).

Zasadniczym fragmentem powyższego projektu jest uzgodnienie dwóch dopuszczalnych metod dostępu do ośrodka, tzw. metody dostępu wspólnego z wykrywaniem kolizji (ang. *carrier sense multiple access with collision detection*, CSMA/CD) i metody dostępu oznakowego (ang. *token access*). Metoda CSMA/CD polega głównie na tym, że stacja może wysłać informację dopiero po każdorazowym sprawdzeniu, że linia jest wolna (ang. *idle*).

3) Zimmermann H.: The ISO Model of Architecture for Open System Interconnection. IEEE Trans. Communication, Vol. 28, p. 425, 1980

Zimmermann H.: Progression of the OSI Reference Model and Its Applications. Proc. National Telecommunications Conf., New Orleans, Louisiana, USA, 29 Nov. — 3 Dec. 1981;

Data Processing — Open System Interconnection — Basic Reference Model. Computer Networks, Vol. 5, No. 3, April 1981

Pierwszą siecią lokalną działającą według tej zasady, jest tzw. ETHERNET firmy XEROX⁴⁾. W metodzie dostępu oznakowego kontrola nad ośrodkiem transmisji jest przekazywana w sposób uporządkowany od stacji do stacji, za pomocą tzw. oznaki (ang. *token*), która upoważnia do transmisji przez określony czas, po czym oznaka jest przekazywana następnej stacji w tzw. pętli logicznej.

Obie przedstawione metody dostępu dają możliwość żądania transmisji, co oznacza zdecentralizowany przydział ośrodka. Ma to istotne znaczenie, gdyż powszechnie stosowane w telekomunikacji metody zwielokrotnienia łącza przez podział czasu lub podział częstotliwości (TDM, FDM, ang. *time division multiplexing*, *frequency division multiplexing*) są nieefektywne⁵⁾.

Wybór jednej z dwóch metod dostępu jest również ważny, gdyż są one wzajemnie niezgodne. Poza wyborem metody dostępu, użytkownicy sieci lokalnych muszą dokonać wyboru ośrodka transmisji. Obecnie rozważane jest użycie kabli współosiowych⁶⁾ (wąsko- i szerokopasmowych, ang. *baseband*, *broadband*), a w przyszłości także — par skręconych i włókien światłowodowych (ang. *fiber optic cable*).

Wydaje się, że konieczne jest przyjęcie takiego kierunku standaryzacji, aby różne skomputeryzowane urządzenia, pochodzące od różnych producentów, można było łączyć we wspólną sieć. Należy uważnie śledzić rozwój prac za granicą, aby nie popełnić błędów przy wyborze krajowych rozwiązań. Pouczający jest tu przykład systemu CAMAC, który — mimo że ma wielu przeciwników, a do wielu zastosowań się nie nadaje — jest jedynym systemem cyfrowym, produkowanym w Polsce, umożliwiającym użytkownikowi samodzielne utworzenie zestawu. Ważne jest, aby wybrać taki standardowy system rozłożony przestrzennie, który będzie można eksploatować i stopniowo rozbudowywać przez wiele lat.

Wiele obecnie prowadzonych prac dotyczy architektury i organizacji sieci komputerowych przeznaczonych do sterowania, lecz są to na ogół roz-

1) Czytelnicy zainteresowani tą tematyką mogą sięgnąć do poprzednich artykułów z tego cyklu — INFORMATYKA nr 9/1980 i 11—12/1981

2) Istotnym uzupełnieniem tego artykułu jest informacja ze spotkania Komitetu ESONE i stowarzyszenia ECA — p. INFORMATYKA nr 5—6/1981

4) Shoch J. F., Dalal Y. K., Redell D. D., Crane R. C.: Evaluation of the ETHERNET Local Computer Network. Computer, Vol. 15, No. 8, p. 10, 1982

5) Stieglitz M.: Local Network Access Tradeoffs. Computer Design, Vol. 20, No. 10, p. 163, 1981

6) Hopkins G. T., Meisner N. B.: Choosing between Broadband and Baseband Local Networks. Mini-Micro Systems, Vol. 15, No. 6, p. 89, 1982

wiązania jednostkowe, nie zawsze warte naśladowania (I. Ilzyna, ZSRR; J. Lan, Chiny; L. F. Natiello, USA; DCCS). Choć zastosowania wymagające użycia struktury sieciowej mogą być bardzo różnorodne, np. walcownia blach (W. E. Miller, USA), elektrownia konwencjonalna lub jądrowa (P. M. Larsen, Dania; C. A. J. Payne i in., USA), system dystrybucji mocy (A. C. M. Chen, USA) — potrzeby użytkowników nie są aż tak zróżnicowane, aby nie można wydobyc cech wspólnych sieci lokalnych w celu standaryzacji rozwiązań.

ZAGADNIENIA OPROGRAMOWANIA

Do niedawno odkrytych prawd, która dociera także do świadomości użytkowników, należy i taka, że programowanie komputerowych systemów sterowania nie polega jedynie na pisaniu i uruchamianiu programów. W procesie wytwarzania oprogramowania wyróżnia się kilka faz mających wpływ na ostateczną jakość swoistego produktu, jakim jest program. W zależności od tego, jak podzielimy tzw. czas życia programów, można wyróżnić szereg faz rozwoju oprogramowania, np.: specyfikacja, projektowanie, kodowanie, uruchamianie, pielęgnacja.

Uważa się, że żaden większy program nie jest wolny od błędów — nawet po uruchomieniu. Niemniej błędy popełnione we wczesnym okresie wytwarzania programu są trudniejsze do wykrycia i najczęściej ich usunięcie jest możliwe dopiero przy testowaniu, a nawet w trakcie działania programu. Jest to bardzo niekorzystne, gdyż koszty ponoszone wskutek błędnego oprogramowania są tym większe, im później błędy zostaną wykryte. Z tego względu coraz więcej uwagi poświęca się wczesnym fazom wytwarzania programu, przed samym kodowaniem, tzn. — projektowaniu i specyfikacji (specyfikacją nazywa się opis obiektu, określający jego interesujące właściwości). Trudną sprawą jest formalizacja obu procesów, tzn. — specyfikacji wymagań użytkowych, które powinien spełniać system (program) i projektowania programu odpowiadającego tym wymaganiom. Warto więc zwrócić uwagę na prace, które wnoszą wiele istotnych elementów do zrozumienia tych zagadnień.

Bardzo ważnym faktem jest określenie formalnego języka specyfikacji, gdyż dotychczas stosowane języki naturalne, co najwyżej uzupełnione reprezentacją graficzną, nie są wystarczającym narzędziem do zwiększenia niezawodności systemów (niezawodności: oprogramowania). Trudność we wprowadzeniu formalnych języków specyfikacji polega na tym, że muszą one mieć prostą budowę i być łatwe do opanowania przez osoby nie mające wykształcenia informatycznego. Są to warunki konieczne, ponieważ język specyfikacji jest podstawowym środkiem porozumienia między producentem (informatykiem) a użytkownikiem systemu.

Specyfikacja wymagań powinna być przeprowadzona na kilku poziomach abstrakcji (J. Ludewig, H. Sandmayr, Szwajcaria, DCCS). Przykładowo — najwyższy poziom powinien zawierać następujące informacje:

- jakie sygnały wejściowe (do komputera) będą dostępne i jak długo
- jakie sygnały wyjściowe (z komputera) muszą być dostarczane i przez jak długi czas
- jakie jest dopuszczalne opóźnienie sygnału wyjściowego względem wejściowego (czas reakcji)
- jaka musi być niezawodność systemu (w sensie określonego wskaźnika).

Konieczne jest oddzielenie specyfikacji wymagań od specyfikacji projektu systemu (ang. *design specification*). Najlepiej byłoby, gdyby użytkownik nie określał, jak mają być zrealizowane jego wymagania funkcjonalne, tzn. aby nie definiował środków realizacji. Choć bardzo często system komputerowy mający wykonać określone zadanie jest już zainstalowany, znacznie korzystniej jest, jeżeli decyzję tę podejmuje projektant na podstawie specyfikacji wymagań, dostarczonej przez użytkownika. Zwracali na to uwagę także polscy autorzy⁷⁾.

Użyteczność języka specyfikacji jest pełna wtedy, gdy dysponujemy zestawem różnorodnych narzędzi programowych języka. Mówiąc krótko, niezbędne jest tzw. środowisko programowe dla języka. Wtedy można mówić o istnieniu tzw. systemu specyfikacji (ang. *specification system*). Jednym z pierwszych takich systemów był — dziś już dobrze znany — tzw. PSL/PSA⁸⁾. Inny system, ESPRESO, przeznaczony do specyfikacji oprogramowania na potrzeby sterowania, powstał w Karlsruhe (RFN) i jest rozwijany w firmie BROWN-BOVERY AND CO. Taki system, a w szczególności język, musi także podlegać prawom specyfikacji, co prowadzi do tzw. metaspecyfikacji (J. Ludewig, Szwajcaria, RTP). Bardzo interesujące są prace prowadzone nad wykorzystaniem języka ADA do specyfikacji⁹⁾ (S. Savoy, Francja, RTP).

Konieczność gruntownej zmiany podejścia do programowania komputerowych systemów sterowania wynika z guzej zawodności oprogramowania nawet w tak krytycznych zastosowaniach, jak loty kosmiczne i technika wojskowa. Obecnie nie ma dostatecznej ogólnej teorii, na której można by oprzeć projektowanie niezawodnych systemów sterowania. Podejmuje się

jednak wiele prób opisu teoretycznego i realizacji praktycznej tzw. systemów bezpiecznych, ang. *safe systems* (R. M. Konakovskiy, RFN, Kongres).

ROLA CZŁOWIEKA W KOMPUTEROWYCH SYSTEMACH STEROWANIA

Jeżeli komputeryzacja staje się tak powszechna, że sprzęt zastępuje człowieka nawet w najbardziej skomplikowanych zastosowaniach, a oprogramowanie zapewnia komputerom wystarczającą inteligencję, to nasuwa się pytanie — co robi człowiek. Jest to kwestia niezwykle ważna — człowiek zastępowany przez komputery nie może być wypierany, jego predyspozycje powinny być wykorzystane przy wykonywaniu nowych, może bardziej odpowiedzialnych zadań.

Komputery mogą stać się użytecznym narzędziem w ręku operatora systemu, dostarczając mu pełniejszej (a jednocześnie bardziej skondensowanej) informacji, która umożliwia dokładniejszą ocenę przebiegu procesu i lepszy nadzór (M. Cuenod, Turcja, DCCS). Operator wyposażony w inteligentnego pomocnika ma pełniejszą kontrolę nad procesem i może choćby przewidzieć sytuacje awaryjne lub zapobiec im. Zatem operator jest obecnie traktowany jako integralny element systemu sterowania, na co miały wpływ przede wszystkim doświadczenia z wypadków w elektrowniach jądrowych i — katastrof lotniczych¹⁰⁾. W wielu przypadkach, odpowiednio przeszkolona obsługa mogła zmniejszyć rozmiary szkód lub całkowicie zapobiec awariom (G. Johanson, RFN, Kongres).

Diagnostyka sytuacji nienormalnych jest prawie wyłącznie domeną człowieka. Jednak dla pełnego, efektywnego wykorzystania operatora w procesie sterowania obiektami, niezbędny jest opis matematyczny jego zachowania. Nowym narzędziem, które — jak się wydaje — może adekwatnie ująć złożoność ludzkich zachowań, jest teoria zbiorów rozmytych¹¹⁾ (W. B. Rouse, A. M. Hunt, USA, Kongres).

Najpełniejszy, znany mi przegląd zagadnień współpracy człowieka z komputerem, który warto polecić uwadze Czytelników, przedstawili znakomici specjaliści tej dziedziny, William Rouse i Thomas Sheridan¹²⁾

10) Livingstone W. L.: The Impact of Three Mile Island on Process Control Computer Technology. *Control Engineering*, Vol. 27, No. 11, p. 87, 1980

11) Czogała E., Pedrycz W.: Elementy i metody teorii zbiorów rozmytych. *Skr. uczeln.* nr 989, Politechnika Śląska, Gliwice, 1980

12) Rouse W. B.: Human-Computer Interaction in the Control of Dynamic Systems. *ACM Computing Surveys*, Vol. 13, No. 1, pp. 71-99, March 1981;

Sheridan T. B.: Theory of Man-Machine Interactions as Related for Computerized Automation. *Proc. 6th Advanced Control Conf.*, Lafayette, Indiana, USA, April 1980

7) Górski J., Sołtysik J., Szejko S.: Problemy budowy komputerowych systemów automatyzacji pomiarów i sterowania. *Prace VIII Krajowej Konferencji Automatyki*, Szczecin, 16-17 września 1980, str. 75, tom 2

8) Teichroew D., Hershey III E. A.: PSL/PSA — A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems. *IEEE Trans. SE*, Vol. 3, p. 41, 1977

9) Goldsack S. J., Kramer J.: Invariants in the Application-oriented Specification of Control Systems. *Automatica*, Vol. 18, No. 1, pp. 71-76, 1982

ZASTOSOWANIA KOMPUTERÓW DO STEROWANIA (NA PRZYKŁADZIE JAPONII)

Zastosowanie komputerów do sterowania wiąże się przede wszystkim ze wzrostem wydajności procesów technologicznych, polepszeniem jakości produktów oraz z poprawą warunków pracy. W kraju przodującym technologicznie, jakim jest Japonia, potrzeby te są tak silne, że wywołują nieustanny postęp i rozwój techniki komputerowej w kierunku zwiększenia szybkości komputerów, zwiększenia niezawodności sprzętu i oprogramowania, zmniejszenia kosztów produkcji oprogramowania, wytwarzania wszechstronniejszych narzędzi programowych i opracowania inteligentniejszych algorytmów użytkowych (H. Kuwahara, T. Miura, Japonia, RTP).

Pod koniec lat siedemdziesiątych w przemyśle japońskim pracowało prawie 4000 komputerów. Najbardziej stabilne, choć nie najliczniejsze zastosowania dotyczą takich dziedzin gospodarki, jak: wytwarzanie energii elektrycznej (2,5% ogółu), produkcja stali (3,1%), przemysł chemiczny (3,3%). Pełna automatyzacja instalacji w tych dziedzinach nie należy do rzadkości. Niezależnie od tego, najpowszechniejsze jest zastosowanie komputerów w automatyzacji procesów wytwórczych (17% ogółu zastosowań) oraz instalowanie komputerów w produktach podlegających sprzedaży (ang. *original equipment manufacturer's applications* — OEM, 29,8%). Obecnie ekspansja komputerów w gospodarce japońskiej następuje w kierunku takich zastosowań, jak: automatyzacja laboratoriów (10,8%), medycyna (8,5%) i telekomunikacja (9%).

Wymienione zastosowania zwiększają wymagania stawiane systemom komputerowym, co prowadzi do podejmowania prób wykorzystania najnowszych osiągnięć tej techniki. Przykładowo — od trzech lat w zakładach NIPPON STEEL CORPORATION i HITACHI LTD. działają rozłożone przestrzennie systemy sterowania oparte na przesyłaniu informacji przez włókna światłowodowe, a także — z wykorzystaniem satelitów. Natomiast w celu zwiększenia tzw. produktywności w dziedzinie oprogramowania (ang. *programmer's productivity*) w firmie HITACHI LTD. wykorzystuje się tzw. stacje wspomaganie (ang. *software workbench station*), sprzężone z centrum produkcji oprogramowania, a wyposażone w zestaw narzędzi programowych (nieustannie uzupełniany nowymi), wspomagających wytwarzanie oprogramowania w pełnym cyklu życia programów, tzn. od fazy specyfikacji do pielęgnacji programów.

Wzorcowym przykładem kompleksowego zastosowania komputerów do sterowania jest tzw. KIIS (*Kimitsu Integrated Information System*, NIPPON STEEL CORPORATION), gdzie najnowsze osiągnięcia techniki komputerowej, zarówno w zakresie architektury systemu, jego organizacji, metod oprogramowania, jak i współpracy człowieka z komputerem, zastosowano

do sterowania różnorodnych procesów hutniczych (H. Mitsuda i in., Kongres, K. Inoue i in., RTP).

Potrzeby gospodarki w zakresie nowych metod techniki komputerowej wpływają nie tylko zewnętrznie na rozwój tej techniki, ale stymulują rozwój technologii, wymuszają głębsze wnikanie w procesy warunkujące postęp technologiczny. Działania pojedynczych firm byłyby jednak bardzo nieefektywne, gdyż przedsięwzięcia tego rodzaju wymagają ogromnych nakładów i ścisłej koordynacji. Z powyższych względów, realizację szeroko zakrojonego projektu, znanego pod nazwą „Systemy komputerowe piątej generacji”^{12, 13} rozpoczęto pod patronatem rządu Japonii. Podejmowane są również próby nawiązania współpracy międzynarodowej¹⁴.

Celem przedsięwzięcia jest zintegrowanie badań, dotąd luźno powiązanych zastosowaniami, w czterech zasadniczo oddzielnych kierunkach:

- technologii układów o bardzo dużym stopniu scalenia (VLSI)
- rozłożenia przestrzennego procesów obliczeniowych
- języków programowania bardzo wysokiego poziomu
- sztucznej inteligencji (ang. *expert systems, knowledge-based systems*). Przewiduje się, że komputery lat dziewięćdziesiątych, komputery piątej generacji, będą zbudowane z elementów składowych, umożliwiających — jak klocki LEGO — tworzenie i dowolne zestawianie jednostek sprzętowych i modułów programowych, dostosowanych do wymagań użytkowych. Komputery będzie można, z kolei, łączyć w sieci informacyjne na kształt współczesnych sieci telefonicznych. Oczywiście, podstawą tak ambitnego przedsięwzięcia, muszą być nowe koncepcje teoretyczne, co oznacza zdecydowane odejście od architektury von Neumanna.

* * *

Przeniknięcie mikroprocesorów do systemów sterowania jest faktem nie podlegającym dyskusji. W efekcie następuje rozczłonkowanie tych systemów na podsystemy, układy lub jednostki wykonujące pojedyncze, bardziej wyspecjalizowane funkcje. Konsekwencją tego faktu jest powstanie zupełnie nowych zagadnień organizacji takich systemów. Decydującym elementem sprawnego i niezawodnego działania komputerowych systemów sterowania jest ich oprogramowanie. Dlatego tak duże nakłady przeznaczają się na zwiększenie tzw. produktywności

ności programistów (tj. zdolności do efektywnego wytwarzania dobrego jakościowo oprogramowania) i zapewnienie trwałości, odporności i niezawodności programów (kierunek ten dość skutecznie omija Polskę). Pewnym remedium na złożoność zarówno sprzętu, jak i oprogramowania — jest normalizacja, prowadząca do modularności komputerów i programów.

Podsumowując przedstawione uwagi, należy stwierdzić, że zagadnienie wykorzystania komputerów do sterowania stało się problemem o znaczeniu strategicznym. Świadczy o tym choćby takie przedsięwzięcie, jak projekt opracowania języka ADA i jego środowiska programowego, finansowany początkowo przez jedno — ale jakże potężne — ministerstwo, tj. Departament Obrony USA, a obecnie — także przez ministerstwa obrony lub przemysłu najbardziej rozwiniętych krajów zachodnich. Świadczy o tym również przystąpienie do stworzenia systemów komputerowych piątej generacji — określanej jako największy i najbardziej obiecujący projekt komputerowy dotychczas rozpoczęty („*the most challenging and extensive computer project undertaken to date*”), co zostało zainicjowane i jest finansowane przez rząd Japonii.

Trzy programy badawcze dotyczące komputerowych systemów sterowania, na które Federalne Ministerstwo Rozwoju Technologii RFN (BMFT) przeznaczyło w latach 1973—79 od 1,5 do 3 mld marek omówiono *in extenso* na Kongresie (P. Brödner, T. Martin). Lektura tego artykułu jest niezmiernie pouczająca.

Wydaje się, że nawet ten fragmentaryczny obraz nowych tendencji na świecie wskazuje przynajmniej część kierunków — zarówno badawczych, jak i aplikacyjnych, których podjęcia nie można odkładać. Wszyscy zdajemy sobie sprawę, że odległość, jaka nas dzieli od czołówki, stale się powiększa. Na pocieszenie można powiedzieć, że CEMI rozpoczęło produkcję mikroprocesorów (wiem to stąd, że pojawiły się już one na warszawskim pchlim targu).

Prace z omawianych konferencji IFAC zostały wydane w formie książkowej przez wydawnictwo Pergamon Press:

- T. Hasegawa, ed., *Real Time Programming 1981*
- W. E. Miller, ed., *Distributed Computer Control Systems*.

Zbiór preprintów Kongresu IFAC znajduje się w bibliotece Instytutu Badań Systemowych PAN, ul. Newelska 6, 01-447 Warszawa (sygnatura nr 35757). Warto również zacytować, trudniejsze do zdobycia, wydawnictwo japońskie nt. komputerów piątej generacji: *Proceedings of International Conference on Fifth Generation Computer Systems*, Tokyo, 19—22 October 1981, Japan Information Processing Development Center.

¹² Treleaven P. C., Gouveia Lima I.: Japan's Fifth Generation Computer Systems. *Computer*, Vol. 15, No. 8, p. 79, 1982

¹³ Sobczyk M. (oprac.): Japonia — piąta generacja. *INFORMATYKA*, nr 7, str. 32, 1982

¹⁴ Yasaki E. K.: Tokyo Looks to the 90's. *Datamation*, Vol. 28, No. 1, p. 110, 1982

Słowniczek mikroprocesorowy (cz. 2)

ODŚWIEŻANIE (*refreshment*) — odnawianie zawartości pamięci dynamicznej, wymagane — wg danych podawanych przez producentów — nie rzadziej niż co 2 ms. Ze względu na odświeżanie, które z reguły jest realizowane sprzętowo, asynchronicznie w stosunku do rytmu wykonywania programu, czas dostępu¹⁾ pamięci dynamicznych nie jest stały, co wymaga generowania potwierdzenia.

OPROGRAMOWANIE UKŁADOWE (*firmware*) — 1) mikroprogramy umieszczone w ROM, 2) inne ustalone „firmowe” oprogramowanie podstawowe, jak np. edytor tekstowy i assembler, umieszczone w ROM.

PAMIĘĆ DOMENOWA (*bubble memory*, niepopr. pamięć pęcherzykowa, bąbelkowa) — pamięć, w której bity informacji są reprezentowane przez obecność albo brak domen cylindrycznych w określonych miejscach cienkiej warstwy magnetycznej. Domeny są tworzone przez stałe pole magnetyczne, przy czym przemieszczają się na zasadzie przesuwania zawartości rejestru szeregowego, o typowej częstotliwości przesuwu równej 100–500 kHz. Typowy średni czas dostępu pamięci domenowej jest równy 2–4 ms.

PAMIĘĆ DYSKIETKOWA — pamięć dyskowa wyposażona w dyski elastyczne, najczęściej 1–2 jednostki napędu, oraz niezbędne układy elektroniczne. Pamięci dyskietkowe są powszechnie stosowane w systemach mikrokomputerowych ze względu na stosunkowo małe rozmiary, niskie ceny oraz wymienne nośniki.

PIÓRO ŚWIETLNE — sonda fotoelektryczna sprzężona ze sterownikiem monitora ekranowego, służąca do identyfikacji wybranego obszaru lub punktu na ekranie. Skorelowanie chwili reakcji sondy na pojawienie się plamki z sygnałami synchronizacji monitora umożliwia określenie współrzędnych punktu na ekranie. Pióro świetlne jest wykorzystywane przy organizowaniu współpracy człowiek-maszyna, zwłaszcza przy projektowaniu wspomaganym komputerowo.

PLA (*Programmable Logic Array*) — matryca logicznych bramek iloczynu i sumy (wykonana w technologii MOS lub bipolarnej), których połączenia wykonuje się przez zmianę masek w trakcie produkcji lub (rzadziej) np. przez wypalanie za pomocą lasera. PLA konstrukcyjnie przypomina pamięć ROM, a funkcjonalnie w pewnym zakresie stanowi jej alternatywę, zwłaszcza w przypadku konwersji kodów oraz dekodowania rozkazów. W porównaniu do zestawów bramek TTL charakteryzuje się mniejszą mocą rozpraszaną i powierzchnią montażu oraz nieco większą niezawodnością i mniejszą szybkością działania.

POBRANIE (*fetch*) — cykl maszynowy odczytu kodu rozkazu. W zależności od formatu rozkazu pobranie może być powtarzane, przykładowo w mikroprocesorze INTEL 8080 — 1–3 razy dla każdego rozkazu.

POLLING — metoda organizowania obsługi urządzeń zewnętrznych, polegająca na programowym sprawdzaniu ich stanu (dokładniej — zawartości odpowiednich portów) w rytmie pracy mikrokomputera. Jest prostsza od metody przerwań, aczkolwiek mniej przystosowana do pracy w czasie rzeczywistym.

PORT — miejsce (droga) przez które mikroprocesor komunikuje się z urządzeniem zewnętrznym w celu transmisji danych. W zasadzie, z programowego punktu widzenia, port jest adresowalnym, równoległym 8-bitowym rejestrem.

¹⁾ Terminy zaznaczone rozstrzelonym drukiem są wyjaśnione w Słowniczku — przyp. red.

POTWIERDZENIE (*acknowledge*) — sygnał sprzęgu oznaczający poprawną reakcję układu lub urządzenia na odpowiedni sygnał sterujący. Sygnały potwierdzenia stosuje się w przypadku dwu lub kilku współpracujących modułów o różnej szybkości lub różnych częstotliwościach zegara — działających niezależnie w sensie wewnętrznych rytmów pracy, tzn. asynchronicznie.

Przykładowo — mikroprocesor wysyła potwierdzenie przyjęcia sygnału przerwania z urządzenia zewnętrznego, co oznacza przejście do odpowiedniego podprogramu obsługi lub — oczekuje potwierdzenia (przyjęcia bądź wysłania danych) ze strony sterowanego urządzenia zewnętrznego, względnie pamięci. Brak potwierdzenia w tym drugim przypadku powoduje np. przejście mikroprocesora INTEL 8080 do stan WAIT.

PROGRAMATOR — urządzenie wpisujące dane lub programy do pamięci PROM (EPROM).

PROGRAMOWALNY UKŁAD — sterownik lub inny układ (np. wejściowo-wyjściowy) zawierający jeden lub kilka portów, którego właściwości lub konfiguracja są definiowane lub zmieniane programowo przez mikroprocesor za pomocą odpowiednich słów kontrolnych.

PROM (*Programmable ROM*) — pamięć typu ROM, której zawartość (najczęściej programy lub tablice stałych) jest wpisywana przez projektanta konkretnego systemu mikrokomputerowego, tzn. przez użytkownika, a nie przez producenta układów scalonych. Najbardziej typowe pamięci PROM mają pojemności równe 8–16 K bitów w organizacji 1–2Kx8 bitów.

PUNKT WSTRZYMANIA (*breakpoint*) — miejsce w programie wyspecyfikowane przez operatora lub programistę (najczęściej przez podanie adresu wybranego rozkazu). W miejscu tym wykonywanie programu może zostać wstrzymane — bądź na stałe, np. w celu zbadania określonych warunków, bądź tylko chwilowo, np. w celu odnotowania informacji o stanie mikrokomputera i realizacji programu (por. debugging, system wspomaganie).

RAM (*Random Access Memory*) — pamięć o tzw. swobodnym dostępie, umożliwiająca zarówno zapis, jak i odczyt danych (swobodny dostęp — w sensie możliwości zaadresowania dowolnej komórki przy zachowaniu jednakowego, a zarazem krótkiego czasu dostępu — jest również cechą pamięci typu ROM). Rozróżniamy statyczne pamięci RAM, w zasadzie przechowujące zawartość przez dowolnie długi czas (aczkolwiek pamięci tzw. nietrwale tracą zawartość po wyłączeniu napięć zasilających) oraz dynamiczne pamięci RAM wymagające odświeżania zawartości. Pojemność współczesnych statycznych pamięci RAM jest równa najczęściej 4K bitów (w organizacji 1Kx4), a dynamicznych — 16K bitów (w organizacji 16Kx1).

ROM (*Read-Only Memory*) — pamięć stała umożliwiająca tylko odczyt, której zawartość jest wpisywana w trakcie produkcji. Pamięci ROM są znacznie tańsze od EPROM, jeżeli produkuje się je w dłuższych seriach (orientacyjnie — powyżej 1000 sztuk), dlatego do pamięci ROM wprowadza się albo typowe dla różnych systemów tablice stałych (np. tzw. generator znaków) i programy systemowe w rodzaju monitora lub assemblera, albo indywidualne programy dla stosunkowo małych i masowo sprzedawanych mikrokomputerowych systemów powszechnego użytku.

SCROLLING — przesuwanie obrazu na ekranie monitora, z reguły w kierunku pionowym, połączone ze znikaniem linii na odpowiednim brzegu ekranu oraz z pojawianiem się nowych linii na przeciwnym brzegu.

SEKWENSER — (*sequencer*) w mikroprocesorze segmentowym układ generacji adresu następnego wykonywanego mikrorozkazu, spełniający rolę kontrolno-sterującej części procesora.

SINGLE-CHIP MICROCOMPUTER — p. mikrokomputer jednoukładowy

SPRZĘG (*interface*) — granica między współpracującymi modułami, a właściwie definiująca tę granicę zasady współpracy i (lub) protokoły transmisji danych, a także zasady łączenia tych modułów oraz wykazy i parametry wymienianych między nimi sygnałów. Sprzęg jest czasami utożsamiany z układami realizującymi te zasady — dotyczy to zwłaszcza programowalnych układów wejściowo-wyjściowych umożliwiających dołączenie urządzeń zewnętrznych do szyn systemowych. Sprzęgi standardowe umożliwiają tworzenie modularnych systemów mikrokomputerowych, a zwłaszcza dołączanie wymiennych modułów, często opracowywanych i produkowanych w różnych firmach (p. arbitraż, multibus, S-100, transmisja asynchroniczna).

STEROWNIK — układ umożliwiający współpracę mikroprocesora z urządzeniami zewnętrznymi (np. z pamięcią dyskietskową) lub układ kontrolujący pracę całego mikrokomputera (np. sterownik przerw, sterownik odświeżania pamięci dynamicznych). Sterowniki są coraz częściej programowalnymi układami scalonymi, a nawet są wyposażone w mikroprocesory — jak np. układ UPI 8041.

SYSTEM MIKROKOMPUTEROWY — sprzęt (mikrokomputer z urządzeniami zewnętrznymi) oraz pełne oprogramowanie (podstawowe i użytkowe), a w niektórych przypadkach także — człowiek-operator. System mikrokomputerowy realizuje konkretne algorytmy użytkowe, tzn. określony cel.

SYSTEM WSPOMAGANIA (*development system* — system uruchomieniowy) — system mikrokomputerowy służący do uruchamiania i badania innych, użytkowych systemów mikrokomputerowych. Poza właściwym uruchamianiem, m.in. łącznie z tłumaczeniem oraz ingerowaniem w wykonywanie uruchamianego programu (jak np. wstrzymywanie, notowanie historii wykonywania programu, dostęp do pamięci i rejestrów roboczych) najważniejszą funkcją systemu wspomaganego jest możliwość uruchamiania programów w pamięci RAM (tzn. przed wpisaniem do PROM), ale w warunkach maksymalnie zbliżonych do docelowych, tzn. na docelowym sprzęcie i z szybkością zbliżoną do docelowej (por. ICE).

SZYNA — zespół linii danych lub linii adresowych, lub linii kontrolno-sterujących, służących do przesyłania sygna-

łów między modułami systemu dołączonymi równolegle. Ze względu na ograniczoną liczbę końcówek układów scalonych (najczęściej nie przekracza ona 40), w niektórych najnowszych mikroprocesorach i sprzętach standardowych występują wspólne linie adresów i danych wykorzystywane na zasadzie przełączania (technika multipleksowa).

S-100 — standardowy sprzęg wewnątrzkasety wprowadzony początkowo w mikrokomputerze ALTAIR firmy MITS, starszy od sprzęgu multibus, ale również przeznaczony przede wszystkim dla mikroprocesora INTEL 8080. Jest szeroko stosowany w USA, zwłaszcza w komputerach osobistych różnych firm.

TRANSMISJA ASYNCHRONICZNA — pojęcie stosowane w przypadku transmisji szeregowej. Jest to transmisja, w której chwile występowania transmitowanych znaków mogą być w zasadzie dowolne, a odstępy czasu między tymi znakami — nierównej długości. Sposób kodowania znaków przez nadajnik umożliwia ich przyjmowanie przez odbiornik. W systemach mikrokomputerowych transmisja asynchroniczna jest organizowana przez specjalne sterowniki, które umożliwiają transmisję dwukierunkową zgodnie z przyjętym standardem np. V24, RS232. Przykładami takich sterowników są: USART 8251 firmy INTEL i ACIA MC6850 firmy MOTOROLA.

UKŁAD TRÓJSTANOWY (ang. *three-state*) — układ, którego wyjścia, poza normalnymi w technice komputerowej dwoma poziomami napięcia odpowiadającymi zeru i jedynce logicznej, przyjmują trzeci, nieoznaczony stan charakteryzujący się dużą impedancją. Układ trójstanowy nie jest układem logiki trójwartościowej, jest raczej wyłączanym układem dwustanowym. Układy trójstanowe umożliwiają dołączenie do szyn wielu źródeł sygnałów napięciowych, z których tylko jedno może być w danej chwili aktywne.

URUCHAMIANIE (*debugging*) — wykrywanie i usuwanie błędów oprogramowania lub sprzętu, a także uszkodzeń sprzętu. Uruchamianie wymaga stosowania specjalnych narzędzi programowych i (lub) sprzętowych, m.in. wykorzystania punktów wstrzymania (por. system wspomaganego).

WEJŚCIOWO-WYJŚCIOWY (*Input-Output, we-wy*) — ogólne określenie dotyczące komunikowania się mikrokomputera z otoczeniem przez porty, do których są dołączone urządzenia zewnętrzne. Przykładowo — istnieją rozkazy wejściowo-wyjściowe, transmisja danych wejściowo-wyjściowych, układy wejściowo-wyjściowe.

MAREK TADEUSZ JANKOWSKI

KSIAŻKI NADESLANE

WYDAWNICTWA NAUKOWO-TECHNICZNE

Władysław M. Turski: Metodologia programowania. Wydanie drugie, nakład 5000 egz., 262 str., cena 120 zł, seria BIO, Warszawa 1982.

Książka poświęcona jest metodologicznym aspektom konstrukcji programów i obejmuje problemy związane ze strukturami danych, modularnością oprogramowania oraz organizacją pracy zespołów programistów. Wydanie to jest rozszerzone w stosunku do poprzedniego głównie o logiczne podstawy programowania, umożliwiające ścisłe formułowanie i dowodzenie własności struktur danych i programów.

Lech Banachowski, Antoni Kreczmar: Elementy analizy algorytmów. Nakład 5000 egz., 187 str., cena 120 zł, seria BIO, Warszawa 1982.

Książka stanowi wprowadzenie w problematykę semantycznej poprawności algorytmów oraz ich złożoności obliczeniowej. Przedstawione pojęcia ilustrowane są dużą liczbą przykładów. Autorzy między innymi przeprowadzili dyskusję takich powszechnie spotykanych w praktyce zadań nie-liczbowych przetwarzania danych, jak: sortowanie, selekcja i znajdowanie klas równoważności.

Edwin E. Klingman: Projektowanie systemów mikroprocesorowych. Nakład 10 000 egz., 526 str., cena 260 zł, Warszawa 1982.

Recenzję tej książki opublikowaliśmy w poprzednim numerze.

Grzywak A., Suchorończak Z.: Rozwój systemów mikrokomputerowych na przykładzie MERY-6) (SM 1633)

INFORMATYKA 1983, nr 2, s. 3

Charakterystyka systemu mikrokomputerowego MERA-60. System ten został włączony do rodziny małych maszyn cyfrowych Jednolitego Systemu jako SM-1633. Podano opis sprzętu i oprogramowania, wyniki badań międzynarodowych oraz koncepcję przekształcenia tego modelu w system następnej generacji MERA-60/256.

Jagiello S., Koźmiński A., Rzymkowski K.: Wybrane mikrokomputerowe bloki sterujące systemu CAMAC i mikrokomputery o przeznaczeniu uniwersalnym

INFORMATYKA 1983, nr 2, s. 9

Charakterystyka mikrokomputerowych sterowników kaset CAMAC opracowanych w Instytucie Badań Jądrowych w Świerku. Omówiono architekturę, funkcję, oprogramowanie oraz zastosowania sterowników.

Pieńkos J. i inni: Modułowy system sterowników mikroprocesorowych MIKROSTER

INFORMATYKA 1983, nr 2, s. 13

Charakterystyka modułowego systemu sterowników mikroprocesorowych umożliwiających konstruowanie sterowników programowanych oraz budowę systemów wspomaganego projektowania oprogramowania użytkowego. Omówiono architekturę systemu oraz funkcje i parametry poszczególnych rodzajów pakietów, a także zestaw projektowo-uruchomieniowy oparty na modułach tego systemu i jego oprogramowanie. W zakończeniu podano perspektywę rozwoju systemu oraz zamierzenia dotyczące rozpoczęcia produkcji.

Woźniak A. i inni: Modułowy system mikroprocesorowy MSM

INFORMATYKA 1983, nr 2, s. 17

Charakterystyka modułowego systemu mikroprocesorowego opracowanego w Instytucie Informatyki Politechniki Warszawskiej. System ten, przeznaczony do projektowania i uruchamiania układów i systemów mikroprocesorowych, zastosowano w procesie dydaktycznym oraz do zaprojektowania urządzeń specjalizowanych. Omówiono poszczególne moduły systemu, jego oprogramowanie, a także dalszy rozwój konstrukcji i zastosowań.

Binkowski W., Furyk M., Przybylski A.: Alfaskop ALFA 31M jako urządzenie wejścia-wyjścia systemu mikrokomputerowego

INFORMATYKA 1983, nr 2, s. 22

Zastosowanie alfaskopu ALFA 31M, przystosowanego do współpracy z komputerem ODRA 1305, w małym systemie mikrokomputerowym. Podano sposób realizacji sprzęgu oraz możliwości funkcjonalne i oprogramowanie alfaskopu.

Jankowski M. T., Wertel J.: Jednopakietowy inteligentny sterownik graficzny MSG-3C

INFORMATYKA 1983, nr 2, s. 24

Charakterystyka jednopakietowego sterownika do graficznego obrazowania danych w systemach mikrokomputerowych. Sterownik został opracowany i zbudowany w Instytucie Technologii Elektronowej w Warszawie z układów produkcji krajowej. W rozwiązaniu uwzględniono możliwość użycia sterownika w systemie mikrokomputerowym MERA ZSA MIKRO-80.

Sobczyk M.: Dyskowa system operacyjny CP/M. Część 1

INFORMATYKA 1983, nr 2, s. 25

Pierwsza część charakterystyki systemu operacyjnego CP/M, powszechnie stosowanego w systemach mikrokomputerowych. Omówiono budowę systemu, pliki, polecenia rezydentne i nierezydentne oraz zasady pisania programów użytkowych.

Гжывак А., Сухороньчак З.: Развитие систем микро ЭВМ на примере MERA-60 (SM-1633)

INFORMATYKA 1983, № 2, стр. 3

Характеристика системы микро ЭВМ MERA-60. Система эта была включена в семейство малых вычислительных машин Единой Системы как SM-1633. Описано оборудование и программное обеспечение, результаты международных исследований и концепцию преобразования этой модели в систему следующего поколения MERA-60/256.

Ягелло С., Козьминьски А., Жымковски К.: Выбранные микро ЭВМ блоки контролирующие системы CAMAC и микро ЭВМ предназначены для универсального употребления

INFORMATYKA 1983, № 2, стр. 9

Характеристика микро ЭВМ контроллеров крейтов CAMAC, разработанных Институтом Ядерных Исследований в Сверку. Обсуждено структуру, функцию, программное обеспечение и применение контроллеров.

Пенькос И., и другие: Модульная система контроллеров микропроцессорных MIKROSTER

INFORMATYKA 1983, № 2, стр. 13

Характеристика модульной системы контроллеров микропроцессорных, делающей возможным конструирование контроллеров программных и строительство систем вспомоществования проектирования утилитарного программного обеспечения. Обсуждено структуру системы, функции и параметры отдельных родов пакетов, а также состав проектно-внедряющий, опирающийся на эту систему и его программное обеспечение. В заключении определено перспективы развития системы и намерения касающиеся приведения в движение производства.

Возняк А., и другие: Модульная микропроцессорная система MSM

INFORMATYKA 1983, № 2, стр. 17

Характеристика модульной микропроцессорной системы разработанной Институтом Информатики Варшавского Политехнического Университета. Система эта, предназначена для проектирования и отладки программ и систем микропроцессорных, дидактики и для составления проекта специализированных оборудований. Обсуждено отдельные модули системы, её программное обеспечение, а также дальнейшее развитие конструкций и применений.

Бинковски В., Фурьк М., Пжибыльски А.: Буквенно-цифровой дисплей ALFA 31M — как входно-выходное устройство системы микро ЭВМ

INFORMATYKA 1983, № 2, стр. 22

Применение буквенно-цифрового дисплея ALFA 31M, предназначенного для сотрудничества с ЭВМ ODRA 1305, в малой системе микро ЭВМ. Указан способ реализации связи, а также функциональные возможности и программное обеспечение буквенно-цифрового дисплея.

Янковски М. Т., Вертель И.: Однопакетный сообразительный графический контроллер MSG-3C

INFORMATYKA 1983, № 2, стр. 24

Характеристика однопакетного контроллера для графического представления данных в системах микро ЭВМ. Контроллер был разработан и построен Институтом Электронной Технологии в Варшаве из схем отечественного производства. В решении учтено возможность применения контроллера в системе MERA ZSA MIKRO-80.

Собчык М.: Дисковая операционная система CP/M. Часть I

INFORMATYKA 1983, № 2, стр. 25

Первая часть операционной системы CP/M, всеобщей применяемой в системах микро ЭВМ. Обсуждено структуру системы, массивы, рекомендации, а также принципы лисания утилитарных программ.

Grzywak A., Suchorończak Z.: Microcomputer systems development on example of MERA-6) (SM-1633)

INFORMATYKA 1983, No. 2, p. 3

Characteristics of the MERA-60 microcomputer system. The system was incorporated into the small computers family of the Unified System as SM-1633 model. Hardware and software description, results of international tests and idea for conversion to MERA-60/256 model, a next generation system, are presented.

Jagiello S., Koźmiński A., Rzymkowski K.: Selected microcomputer control blocks of the CAMAC system and general purpose microcomputers

INFORMATYKA 1983, No. 2, p. 9

Characteristics of microcomputer controllers for CAMAC crates designed by the Nuclear Research Institute in Świerk. Architecture, functions, software and applications of these controllers are discussed.

Pieńkos J. and others: MIKROSTER — a modular system of microprocessor controllers

INFORMATYKA 1983, No. 2, p. 13

Characteristics of the microprocessor controller modular system for constructing programmed controllers and building development systems for application software. The system's architecture, functions and parameters of separate board types, as well as configuration for designing and testing, based on the system modules, and its software are discussed. In the conclusion prospects of the system development and plans for manufacturing are presented.

Woźniak A. and others: MSM — a modular microprocessor system

INFORMATYKA 1983, No. 2, p. 17

Characteristics of the modular microprocessor system designed by the Institute of Computer Science of the Warsaw Technical University. The system, devoted for designing and testing of microprocessor circuits and systems, was applied in didactic process and for designing of specialized devices. Individual modules of the system and its software, as well as future development of hardware and applications are presented.

Binkowski W., Furyk M., Przybylski A.: ALFA 31M alphanumeric display as input/output for microcomputer system

INFORMATYKA 1983, No. 2, p. 22

Application of the ALFA 31M alphanumeric display, which is adjusted for cooperation with ODRA 1305 computer, in a small microcomputer system. The way for interface realization, as well as display's functional facilities and software are presented.

Jankowski M. T., Wertel J.: MSG-3C — a intelligent graphic controller

INFORMATYKA 1983, No. 2, p. 24

Characteristics of the single board controller for graphic data imaging in microcomputer systems. The controller was designed and built by the Electron Technology Institute in Warsaw using home made chips. In the solution the possibility of the application in MERA ZSA MICRO-80 microcomputer was allowed.

Sobczyk M.: CP/M — a disc operating system. Part 1

INFORMATYKA 1983, No. 2, p. 25

First part of presentation of the CP/M operating system, which is generally applied in microcomputer systems. Structure of the system, files, resident and non-resident commands, as well as writing rules for application programs are discussed.

Grzywak A., Suchorończak Z.: Entwicklung von Mikrorechnersystemen auf Beispiel von MERA-6) (SM-1633)

INFORMATYKA 1983, Nr. 2, S. 3

Eine Charakteristik des MERA-60 Mikrorechnersystems. Dieses System wurde in die ESER-Kleinrechnerfamilie als SM-1633 eingeschlossen. Es wurde Hard- und Softwarebeschreibung, Ergebnisse der internationalen Prüfungen, sowie die Konzeption von Umgestaltung dieses Modells zu einem System folgender Generation MERA-60/256, angegeben.

Jagiello S., Koźmiński A., Rzymkowski K.: Ausgewählte Mikrorechnersteuerblöcke des CAMAC-Systems und universelle Mikrorechner

INFORMATYKA 1983, Nr. 2, S. 9

Eine Charakteristik von Mikrorechnersteuerblöcken für CAMAC-Kassetten, die im Institut für Kernforschung in Świerk entwickelt wurden. Es wurde die Architektur, Funktionen, Software und Anwendungen dieser Steuereinheiten besprochen.

Pieńkos J. und andere: MIKROSTER — ein Modulare System von Mikroprozessorsteuereinheiten

INFORMATYKA 1983, Nr. 2, S. 13

Eine Charakteristik des Modulare Systems von Mikroprozessorsteuereinheiten, die die Konstruierung der programmierten Steuereinheiten und den Bau von Unterstützungssystemen für Anwendungssoftwareprojektion ermöglichen. Es wurde die Architektur des Systems, Funktionen und Parameter der einzelnen Schaltkreisgruppen, sowie auf Basis der Systemmodule gebaute Projektierungs- und Testungskonfiguration und ihre Software besprochen. Im Endteil des Artikels wurden die Entwicklungsaussichten des Systems und die Pläne angesichts des Herstellungsanfangs angegeben.

Woźniak A. und andere: MSM — ein Mikrorechnermodularsystem

INFORMATYKA 1983, Nr. 2, S. 17

Eine Charakteristik des Mikrorechnermodularsystems, das im Institut für Informatik der Warschauer Technischen Universität entwickelt wurde. Das System, vorgesehen für Projektierung und Testung von Mikrorechner-Schaltkreisen und -Systemen, wurde im didaktischen Prozess und bei der Projektierung von Spezialeinrichtungen, verwendet. Es wurden die einzelnen Module des Systems, und seine Software, sowie weitere Entwicklung von Konstruktion und Anwendungen, angegeben.

Binkowski W., Furyk M., Przybylski A.: Alphanumerisches Anzeigegerät ALFA 31M als Ein- und Ausgabe eines Mikrorechnersystems

INFORMATYKA 1983, Nr. 2, S. 22

Eine Anwendung des alphanumerischen, zur ODRA 1305 Rechner angepassten Anzeigegerätes, in einem kleinen Mikrorechnersystem. Es wurden Realisierungsweise der Schnittstelle, sowie Funktionsmöglichkeiten und Software des Anzeigegerätes, angegeben.

Jankowski M. T., Wertel J.: Intelligente graphische Einzelplattesteuereinheit MSG-3C

INFORMATYKA 1983, Nr. 2, S. 24

Eine Charakteristik der Einzelplattesteuereinheit zur graphischen Darstellung von Daten in Mikrorechnersystemen. Die Steuereinheit wurde im Institut für Elektronische Technologie in Warschau mit Verwendung von einheimischen Schaltkreisen erarbeitet und gebaut. In der Lösung wurde die Anwendungsmöglichkeit dieser Steuereinheit im Mikrorechnersystem MERA ZSA MIKRO-80 berücksichtigt.

Sobczyk M.: CP/M — ein Plattenbetriebssystem. Teil 1

INFORMATYKA 1983, Nr. 2, S. 25

Erster Teil der Charakteristik von CP/M Betriebssystem, das allgemein in Mikrorechnersystemen verwendet wurde. Es wurde Systemsbauweise, Dateien, residente und nicht-residente Befehle, sowie Grundsätze für Erarbeitung von Anwendungsprogrammen besprochen.

Małe w informatyce

Błędna decyzja. Kto ponosi winę? Szukamy... i znajdujemy. Oczywiście informatyków. To oni wszystkiemu winni.

Czytelniku, masz wątpliwości? O, to już nieladnie, a nie wiesz, że nasz Wielki lub Mały Decydent pracuje po 16 godzin dziennie, a czasem i dłużej. Śni i pracuje. Optymalizuje, reorganizuje, uszczęśliwia... A że efekt jest mizerny? — To już wina tych, co dostarczyli informacje.

Mieliśmy przecież tak wspaniałe koncepcje i makietki różnych systemów: globalnych, ogólnokrajowych, rządowych. I co? Wstyd! Nie ostrzegły, jak ten strażnik z Wieży Mariackiej, że kryzys nadchodzi.

Spójrzmy na to nieco inaczej: w swoim czasie Jerzy Gwiazdziński celnie, w syntetyczny sposób pokazał¹⁾, że informatyka — relatywnie — nie była zbyt kosztowna. Jeśli zaś porównać świetnie dobrane mierniki planowanych nakładów z ich faktyczną realizacją (to już z innych źródeł), okaże się, że na informatykę środków w większości przypadków nie marnowano.

Zastanówmy się, czy informatyka rzeczywiście nie dała dla podniesienia sprawności przedsiębiorstw. Wiemy przecież, że tak nie jest i że większość systemów pracuje zupełnie nieźle, spełniając założone wymagania. I można chyba stwierdzić, że — przy dysponowanych środkach — były one z reguły zaprojektowane w sposób optymalny.

Już widzę wyciągnięte dłonie: — Pokaż pan te systemy! Przyjmij zakład, że takich systemów nawet z ogarkiem nie znajdziesz!

Kto informatyków powinien chwalić — użytkownicy? Ależ, czy oni się znają na naszych wspaniałych produktach? A poza tym, mają przecież do nas ten sam stosunek jak do innych grup zawodowych chcących im ułatwić życie — często wbrew ich woli (ukrytej głęboko, bardzo głęboko).

I tak mówiąc między nami — które przedsiębiorstwo będzie stosowało kłopotliwy „klucz do dobrobytu”, jeżeli wystarczy mu wytryszek: podniesienie ceny wyrobu?

Myślę więc o zespole Profesora Gościńskiego. Panie Profesorze, zreformujcie reformę tak, aby informacja szybka, rzetelna i wszechstronna była w cenie, by była poszukiwana. Aby sloganowe „Trzy S” stały się faktem. Mógłbym wówczas już spokojnie patrzeć w przyszłość i zawiesić poszukiwanie kursu na czeladnika murarskiego. Informatyka będzie potrzebna i efektywna. Konkurencja — szybko i skutecznie wyeliminuje tu wszystkie knoty. Oczywiście, jeżeli uda się wyciszyć tych, którzy domagają się dla siebie priorytetów. Jeżeli tylko na to pozwolimy i nie będziemy krepowali jej zakazami i nakazami — zweryfikuje nasze produkty, które obecnie spoczywają z pięknymi na-

zwami we wspaniałych importowanych okładkach. I często to tylko je między sobą różni. Czy można kogoś winić? O tak, każdy z nas wymieni wiele nazwisk i instytucji. Niektóre z nich pojawić się mogą na wielu listach. Niemniej barierę szkodliwej działalności można tylko pokonać przez wprowadzenie działającego mechanizmu ekonomicznego. Mechanizmu, który — poza wszystkim — zmusi przedsiębiorstwo do szukania skutecznych sposobów wynagradzania pracowników.

Informatyka będzie nadal działać w zmieniających się warunkach. Już dziś musimy więc przygotować oferty dla przedsiębiorstw przyszłości. I dlatego też muszą istnieć przedsiębiorstwa pracujące dla informatyki.

Dawno temu byłem sekretarzem konkursu. Był to bardzo „Ważny Konkurs” pod bardzo „Ważnym Patronatem”. Otrzymało, jak zwykle, wiele różnych prac. Jedną z nich przedstawiała koncepcję poprawy złej sytuacji w sprzęcie komputerowym. Kandydat do premii proponował proste wyjście z sytuacji. Zebrał z zagranicznych pism oferty pracy dla informatyków wraz z proponowanym wynagrodzeniem. Postulował wysłanie naszych ludzi na te wakaty, opodatkowanie ich, a za uzyskane w ten sposób pieniądze — zakupienie komputerów. Na tych komputerach miało być szkolić następnych fachowców. Tych z kolei wysłać na następne wakaty itd.

W sądzie konkursowym było nam wówczas wesoło. Ale zastanówmy się. Pomysł sam w sobie wart jest przemyslenia. Eksport to fantastyczna szkoła i egzamin dla informatyków. Tak, ale jest jedno zastrzeżenie: eksport kadry często przypomina wyszukiwanie rodzynków w cieście. Dlatego też należy stworzyć takie warunki, aby eksportować w pierwszym rzędzie systemy — jako produkt. Wtedy mogą być wpływy większe, nie wspominając o chwale dla zespołu i kraju.

Wiem, że niektórzy uśmiechną się: ot, następne pobożne życzenie. Ale jestem przekonany, że należy stworzyć tylko odpowiednie warunki ekonomiczne i... Nasuwają się jednak dalsze wątpliwości. Jakie organizacyjne ramy stosować? Czy duże gigantyczne przedsiębiorstwa informatyczne, czy też małe, elastyczne, dostosowujące się szybko do zmieniającego się popytu?

I jeszcze jedno. Każdy z nas wie, że na jednego projektanta-programistę przypada cały łańcuch pośredników. Łańcuch ten jest tym dłuższy, im użytkownik atrakcyjniejszy. Jeżeli zatem porównamy — jak czynią to ekonomiści — całość kosztów do góry lodowej — to koszty bezpośrednie, a ściślej koszty, które powinny być poniesione, aby produkt był dobry, odpowiadają czubkowi tej góry. Koszty administracji, pośpiechu, pośredników, to zaś jej podstawa, często dobrze ukryta. O opłacalności przedsięwzięcia decyduje jednak wielkość całej góry.

I w informatyce małe jest piękne?

¹⁾ Problemy rozwoju informatyki, Gospodarka Planowa nr 7-8 1975, s. 417

Zasady prenumeraty

Zamówienia i przedpłaty na prenumeratę **INFORMATYKI** przyjmuje: Wydawnictwo **SIGMA**, skrytka 1004, 00-950 Warszawa; konto bankowe: nr 1036-7490-139-11 III O/M NBP Warszawa.

Instytucje przesyłają zamówienia, zawierające: liczbę zamawianych egzemplarzy, okres prenumeraty, pełny adres zamawiającego, adresy odbiorców oraz numer konta bankowego zamawiającego. Dopisek **PRENUMERATA STAŁA** przy zamówieniu całorocznym oznacza, że zamawiający nie musi już ponawiać zamówień, lecz jedynie dokonywać corocznie przedpłaty wg obowiązujących cen.

Prenumeratorzy indywidualni dokonują przedpłaty przekazem na ww. konto, podając

na odwoicie odcinka: liczbę egzemplarzy oraz okres prenumeraty.

Terminy przedpłat:

do 25 listopada — I kwartał, I półrocze, cały następny rok (w tym prenumerata stała);

do 10 marca — II kwartał;

do 10 czerwca — III kwartał, II półrocze;

do 10 września — IV kwartał.

Prenumerata kwartalna — 225 zł, półroczna — 450 zł, roczna — 900 zł.

Prenumerata ze zleceniem wysyłki za granicę jest dwukrotnie droższa.

Dodatkowych informacji udziela:

Dział Handlowy Wydawnictwa **SIGMA**, Warszawa, ul. Mazowiecka 12, tel. 26-80-16.

W numerze 3/83 przedstawimy:

Owczarczyk J.: Rastrowe systemy graficzne

Sobczyk M.: Dyskowy system operacyjny CP/M (cz. 2)

Rolland C., Stępniewski J.: Model dynamiki systemu informatycznego

Grabowski J.: Przegląd mikroprocesorów 16-bitowych (1). ZILOG

Drewniak W.: Określenie liczby etatów obsługi technicznej w ośrodku obliczeniowym

W najbliższych numerach:

● **TECHNIKA MIKROPROCESOROWA** (cz. 3)

● Raport dla Klubu Rzymskiego: Mikroelektronika i społeczeństwo. Na dobre i złe (fragmenty)

● Programowanie procesów współbieżnych

● Klasy w **LOGLANIE**