

P. 3057/77

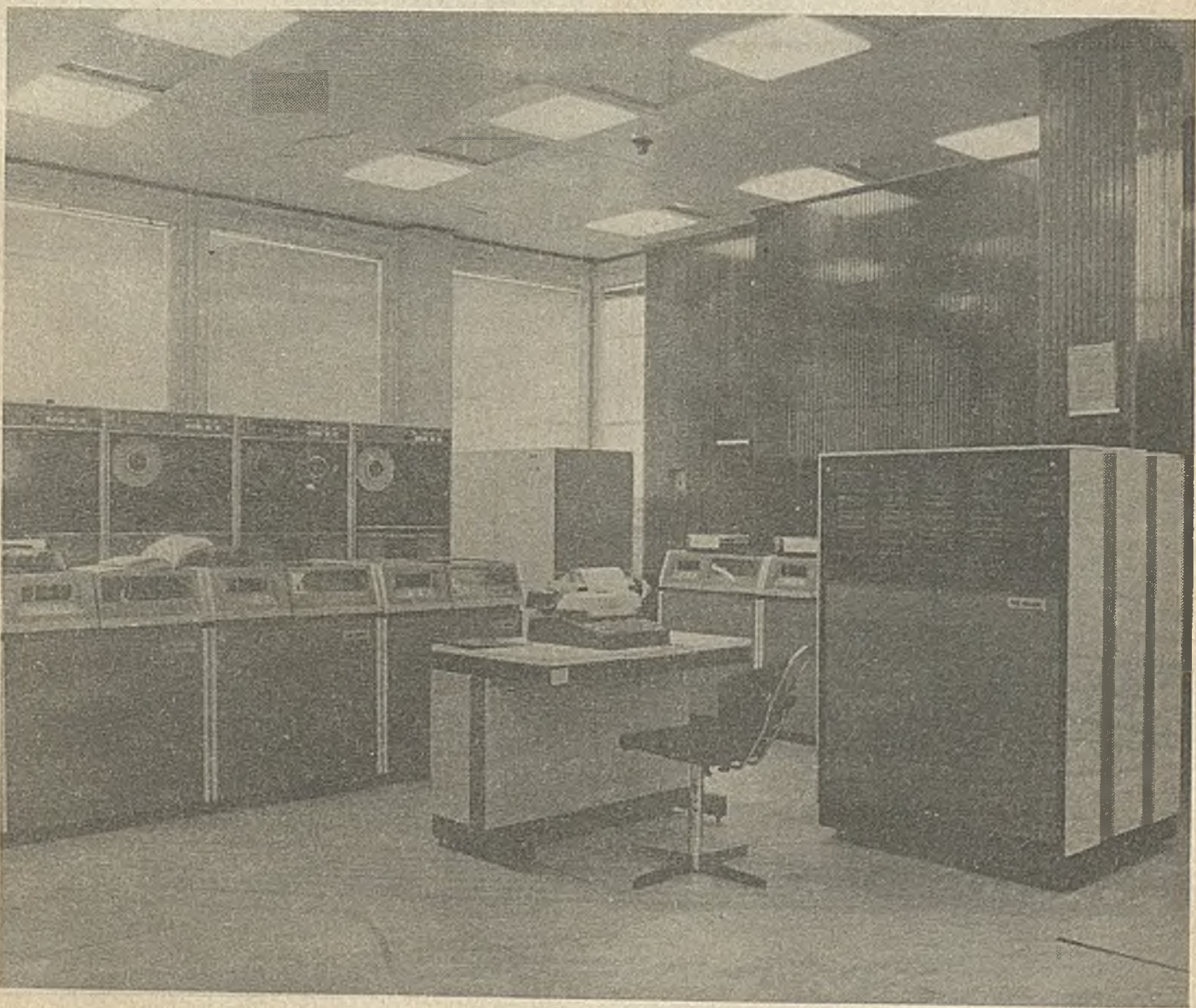


biuletyn informacyjny

3-4
'77



OBIKTOWE
SYSTEMY
KOMPUTEROWE



Zjednoczenie Przemysłu Automatyki i Aparatury Pomiarowej „MERA”
Instytut Maszyn Matematycznych „MERA IMM” Branżowy Ośrodek INTE

DWUMIESIĘCZNIK

Wydaje: CENTRUM NAUKOWO-PRODUKCYJNE TECHNIK KOMPUTEROWYCH I POMIARÓW
INSTYTUT MASZYN MATEMATYCZNYCH
BRANŻOWY OŚRODEK INFORMACJI NAUKOWEJ TECHNICZNEJ I EKONOMICZNEJ

KOMITET REDAKCYJNY

dr inż. Stanisława BONKOWICZ-SITTAUER, mgr inż. Roman CZAJKOWSKI,
mgr Hanna DROZDOWSKA (sekretarz redakcji), dr inż. Marek HOŁYŃSKI,
doc. dr hab. inż. Roman KULESZA (redaktor naczelny) mgr inż. Jerzy
MYSIOR, mgr inż. Józef SZMYD, mgr Robert ZAJĄC

Redaktor techniczny: Maria KOZŁOWSKA

Adres redakcji: ul. Krzywickiego 34, 02-078 Warszawa
tel. 28-37-29 lub 21-84-41 w. 431

na okładce: Maszyna Jednolitego Systemu EMC R-32 wykorzystywana w obiektowych systemach
komputerowej automatyzacji.



P. 3057/77

Biuletyn Informacyjny OBIEKTOWE SYSTEMY KOMPUTEROWE

Rok XVI

Nr 3-4

1977

Spis treści

Содержание

Contents

J. SZMYD, Z. WIŚNIEWSKI, M. WOŹNIAK - Zagadnienie budowy w warunkach krajowych komputerowych systemów sterowania przeznaczonych do automatyzacji diagnostyki samochodowej	И. ШМИД, Э. ВИСНЁВСКИ, М. ВОЗЬНЯК - Вопрос конструкции в условиях национальных компьютерных систем управления, предназначенных для автоматизации автомобильной диагностики	J. SZMYD, Z. WIŚNIEWSKI, M. WOŹNIAK - The problem of domestic building of computer control systems destined for the automotive diagnostics automation
T. CHELSTOWSKI, G. SZYMSIAK - ASTER - System terminalowy na minikomputerze serii MERA 300..	T. ХЕЛСТОВСКИ, Г. ШЫМСЯК - ASTER Терминальная система на мини-ЭВМ серии МЕРА 300	T. CHELSTOWSKI, G. SZYMSIAK - ASTER - Terminal system based on MERA 300 minicomputer ..
J. MYSTOR - Systemy operacyjne w problemowo zorientowanych systemach minikomputerowych...	И. МЫСЁР - Операционные системы в проблемно-ориентированных микрокомпьютерных системах ...	J. MYSTOR - Operating systems for the problem oriented minicomputer systems.....
S. BONKOWICZ-SITTAUER, H. GUTOWSKA - System APT (z cyklu: Programowanie obrabiarek)...	S. БОНКОВИЧ-СИТАУЭР, Х. ГУТОВСКА - Система АРТ (с цикла: программирование станков)	S. BONKOWICZ-SITTAUER, H. GUTOWSKA - APT System (from the series: Machine tool programming)
E. SZCZEP - Kompilatory skrócone	Э. ЩЕП - Перекрестные компиляторы	E. SZCZEP - Cross compilers...
T. SINKIEWICZ - Zastosowanie mikroprocesorów w monitorach ekranowych	T. СИНКЕВИЧ - Применение микропроцессоров в дисплеях	T. SINKIEWICZ - The use of microprocessors in displays....
N. BUDZYŃSKA - Mikroprocesorowy sterownik kasety CAMAC typu CMC 8080	N. БУДЗЫŃСКА - Микропроцессорное устройство управления касетой САМАС типа СМС 8080.	N. BUDZYŃSKA - The microprocessor controller of the CAMAC CMC 8080 type cassette....
K. DZIK, W. KUBERA, W. ZAPENDOWSKI - Sprzęt a oprogramowanie. O pewnych kierunkach rozwoju architektury elektronicznych maszyn cyfrowych	К. ДЗИК, В. КУБЭРА, В. ЗАПЕНДОВСКИ - Технические средства и математическое обеспечение. О некоторых направлениях развития архитектуры ЭВМ	K. DZIK, W. KUBERA, W. ZAPENDOWSKI - Hardware against software. On some development trends of the electronic digital computer architecture
Jednostki centralne Jednolitego Systemu EMC.....	Процессоры ЕС ЭВМ	The central units of the Unified Range Computers....
W. JAKACKI - Techniczne środki kontroli i diagnostyki uszkodzeń elektronicznej maszyny cyfrowej R-32	В. ЯКАЦКИ - Технические средства контроля и диагностики дефектов ЭВМ R-32	W. JAKACKI - The technical means of control and diagnosis of the R-32 computer.....
R. CZAJKOWSKI - System FELIX C32P	Р. ЧАЙКОВСКИ - Система FELIX C32P	R. CZAJKOWSKI - FELIX C32P system
Z kraju i ze świata	Новости	News

Od Redakcji

Biuletyn Informacyjny - Obiektowe Systemy Komputerowe jest kontynuacją wydawanego przez IMM do 1976 r. kwartalnika "Elektroniczna Technika Obliczeniowa - Nowości" (stąd rocznik wydawania nie odpowiada czasowi ukazania się pierwszego numeru Biuletynu). Zamierzeniem Redakcji jest omawianie w poszczególnych numerach Biuletynu określonych tematów, np. obiektowych systemów komputerowej automatyzacji (OSKA), sprzętu dla OSKA, oprogramowanie dla OSKA itp. Dlatego niniejszy numer Biuletynu zawiera głównie artykuły przeglądowe na temat sprzętu, a następny będzie poświęcony oprogramowaniu.

Zapraszamy Czytelników do aktywnego współdziałania w redagowaniu naszego czasopisma, do nadsyłania swoich uwag i propozycji oraz własnych opracowań.

mgr inż. Józef SZNYD
inż. Zbigniew WIŚNIEWSKI
mgr Małgorzata WOŹNIAK

Instytut Maszyn Matematycznych

ZAGADNIENIE BUDOWY W WARUNKACH KRAJOWYCH KOMPUTEROWYCH SYSTEMÓW STEROWANIA PRZEZNACZONYCH DO AUTOMATYZACJI DIAGNOSTYKI SAMOCHODOWEJ

1. Wstęp

Potrzeba komputeryzacji diagnostyki samochodowej wynika z przewidywanego rozwoju motoryzacji w naszym kraju w sensie ilościowym, co spowoduje powstanie zupełnie nowych problemów społecznych, organizacyjnych, technicznych i ekonomicznych. Przekroczenie pewnego progu ilościowego motoryzacji społeczeństwa powoduje, że samochód przestaje być problemem tylko jego właściciela, ale staje się poważnym problemem społecznym, zarówno z punktu widzenia produkcji jak i eksploatacji. Skala zjawiska wymaga zastosowania najnowszych metod i technik produkcji i eksploatacji dla osiągnięcia minimalizacji kosztów wytwarzania i użytkowania pojazdów traktowanych w kategoriach ogólnospołecznych. Należy więc spodziewać się również znacznego wzrostu ilościowego i jakościowego zadań stawianych przed diagnostyką i badaniami samochodów, a w szczególności:

- podwyższenia bezpieczeństwa ruchu drogowego i w konsekwencji obniżenia strat spowodowanych wypadkami drogowymi, w wyniku wykrycia, a następnie usunięcia usterek, które w dalszej pracy uszkodzonych lub rozregulowanych zespołów mogą spowodować wypadek,
- podwyższenia niezawodności samochodów (i w konsekwencji uniknięcia strat spowodowanych nieprzewidzianymi przestojami) w wyniku wykrycia, a następnie usunięcia usterek zanim spowodują one uszkodzenie współpracujących zespołów i unieruchomienie samochodu,
- ekonomicznej i racjonalnej eksploatacji samochodu przy minimalnych kosztach w następstwie wykrycia i korekty odchyleń od optymalnej regulacji zespołów,
- ochrony środowiska (czystość atmosfery, walka z hałasem), a więc obniżenia stopnia zanieczyszczenia atmosfery przez pojazd i stopnia hałaśliwości w wyniku wykrycia, a następnie usunięcia przyczyn powodujących niepełne spalanie mieszanki i wzrost hałaśliwości pojazdu,
- skrócenia czasu i obniżenia kosztów badań, obiektywność oceny stanu technicznego samochodu oraz zwiększenia przebiegów międzynaprawczych przez zapobiegnięcie demontażu zespołów koniecznych przy ocenie tradycyjnej,
- obiektywnej oceny zakresu koniecznej naprawy, a po jej wykonaniu kontroli jakości zapobiegającej niezadowoleniu klientów z prac obsługowo-naprawczych,
- znajomości stanu technicznego samochodu umożliwiającej ocenę opłacalności dalszej jego eksploatacji, celowości dokonania naprawy lub kasacji.

Stale umowocześnianie konstrukcji pojazdu z jednej strony uniemożliwia kontrolę stanu jego podzespołów w sposób tradycyjny (np. ze względu na trudność dostępu wskutek znacznego upakowania), a z drugiej strony wprowadzane są czujniki i przetworniki parametrów technicznych o konstrukcji umożliwiającej wykorzystanie ich do automatyzacji pomiarów diagnostycznych.

Konieczność automatyzacji badań diagnostycznych pojazdu wynika nie tylko z masowości samochodu i różnorodności zadań, jakie się przed diagnostyką stawia, ale i z kwalifikacji właścicieli pojazdów i personelu stacji obsługi. Można się spodziewać, że w przyszłości będzie coraz większy procent właścicieli pojazdów nie posiadających żadnych kwalifikacji (ani chęci) do wykonywania nawet drobnych napraw i regulacji, czy też dokonywania nawet wstępnych przeglądów technicznych.

Wyszkolenie dobrego diagnosty na aparaturę dotychczas stosowaną jest długotrwałe, a równocześnie nie gwarantuje standaryzacji procesu i wyników diagnozy ze względu na znaczny subiektywizm metod. Problem kadr diagnostycznych dla nowopowstałych jak i rozbudowywanych stacji może być rów-

niez rozwiązany jedynie przez optymalne (w aspekcie techniczno-ekonomicznym) zautomatyzowanie procesu badań i stawiania diagnozy.

Stopień automatyzacji badań zależy jest od postępów w modernizacji (i elektronicznej) samochodu i rozwoju skomputeryzowanej aparatury kontrolno-pomiarowej.

Pełna automatyzacja badań nie będzie rozwiązaniem optymalnym, ponieważ zautomatyzowanie niektórych prostych czynności kontrolnych powoduje przedłużenie czasu kontroli i pracochłonności badania spowodowanych koniecznością instalowania w samochodzie członów wykonawczych i czujników przed badaniami. Ponadto nadmierna rozbudowa układów sterowania zmniejsza niezawodność całego systemu kontroli.

Przykładem czynności kontrolnych niemożliwych do zautomatyzowania jest np. ocena stopnia zaawansowania korozji. Racjonalna automatyzacja wymaga stosowania metod i programów badań optymalnych pod względem podatności do zautomatyzowania i powinna obejmować tylko te czynności kontrolne, których zautomatyzowanie spowoduje istotne zmniejszenie czasu i kosztów badań. Pozostałe czynności kontrolne niepodatne do zautomatyzowania powinny być nadal wykonywane dotychczasowymi metodami. Dlatego celowe jest stopniowe wprowadzanie automatyzacji obejmującej początkowo tylko pomiary kontrolowanych wielkości fizycznych, a następnie zastępującej pracę operatora.

Najwyższy stopień automatyzacji - komputeryzacja obejmuje analizę wyników badań, diagnozę dotyczącą przyczyn wykrytych niedomagań oraz zakres, termin i koszt koniecznej naprawy.

W krajach o bardziej niż u nas rozwiniętym przemyśle motoryzacyjnym czynione są próby automatyzowania diagnostyki samochodowej. Zagadnienie to nie jest jeszcze opanowane, ale można przewidywać, że jest to właściwy kierunek rozwoju diagnostyki samochodowej w przyszłości.

Przy obecnym tempie rozwoju motoryzacji w Polsce, zbyt późne wprowadzenie automatyzacji badań diagnostycznych będzie wymagało zakupu licencji lub gotowych urządzeń z zagranicy. Dlatego celowe jest podjęcie w kraju prac nad stopniowym automatyzowaniem poszczególnych badań diagnostycznych samochodu.

2. Automatyzacja diagnostyki samochodowej - stan obecny i tendencje rozwojowe na świecie

● Przykłady realizacji urządzeń do zautomatyzowanej diagnostyki samochodowej

Znane urządzenia zautomatyzowanej diagnostyki samochodowej można podzielić na następujące grupy wg stopnia automatyzacji badań:

- sterowane elektrycznie zestawy konwencjonalnej aparatury diagnostycznej (Elektroniczne Centrum BMW - 1968 r.),
- zautomatyzowane zestawy diagnostyczne (Hofmann M-721),
- skomputeryzowane stanowiska diagnostyczne (VW, Fiat),
- skomputeryzowane linie diagnostyczne (Laycock, stacja w Düsseldorfie).

Sterowane elektrycznie zestawy konwencjonalnej aparatury diagnostycznej. Zestawy odpowiednio przystosowanej konwencjonalnej aparatury diagnostycznej zgrupowane są w jednym centralnym stanowisku diagnostycznym. Operator wykonuje kolejno badania kontrolne wg formularza i dokładnie go wypełnia. Przykładem takiego rozwiązania jest wybudowane w 1968 r. Centrum Elektroniczne BMW umożliwiające kontrolę 32 parametrów silnika.

Zautomatyzowane zestawy diagnostyczne. Przykładem takiego rozwiązania jest urządzenie firmy Hofmann M-721 przewidziane do kontroli samochodów zarówno nieprzystosowanych jak i przystosowanych fabrycznie do zautomatyzowanej diagnostyki. Urządzenie to składa się z 2 podstawowych zespołów połączonych ze sobą elektrycznie za pomocą złącz wtykowych:

- zespół pomiarowo-sterujący odpowiadający nowoczesnemu diagnostykowi z elektronicznie wybieranym programem; zespół ten może pracować samodzielnie zastępując w pełni konwencjonalny diagnostyk,
- zespół programująco-rejestrujący składający się z czytnika kart perforowanych (odpowiadających typowi badanego samochodu) i z drukarki. Zestaw ten umożliwia wykonanie ok. 150 rodzajów badań silnika, hamulców i układu jezdnego.

Skomputeryzowane stanowiska diagnostyczne. Zakres automatyzacji obejmuje kontrolę podatnych do zautomatyzowania parametrów, porównanie wyników badań oraz wydruk wyników z adnotacją "dobrze" - "źle".

Parametry trudno do zautomatyzowanej kontroli są sprawdzane za pomocą prostych przyrządów lub metodami wizualnymi, a wyniki kontroli są ręcznie wprowadzane do stanowiska. Takie rozwiązanie zastosowała firma "Volkswagen-Werke". Stanowisko diagnostyczne jest przewidziane do kontroli samochodów przystosowanych fabrycznie do zautomatyzowanej diagnostyki. System kontroluje 88 parametrów (z tego 25 automatycznie) głównie silnika i instalacji elektrycznej. Stanowisko składa się z następujących podstawowych zespołów:

- maszyna cyfrowa,
- czytnik kart perforowanych,
- diagnostyczne przyrządy pomiarowe przystosowane do współpracy z maszyną cyfrową (geometria kół, ustawienie świateł itp.),
- drukarka,
- uchwyt zdalnego sterowania

Skomputeryzowane linie diagnostyczne. Skomputeryzowane linie diagnostyczne składają się z kilku stanowisk sterowanych elektroniczną maszyną cyfrową. Każde stanowisko jest przystosowane do kontroli jednej grupy pokrewnych parametrów. Na linii bada się jednocześnie po jednym samochodzie na każdym stanowisku, co zmniejsza średni czas badań samochodu proporcjonalnie do liczby stanowisk. Automatyzacja obejmuje podatne do zautomatyzowania czynności kontrolne, porównanie wyników badań oraz wydruk wyników z oceną "dobrze" - "źle".

Przykład skomputeryzowanej linii diagnostycznej składającej się z 5 stanowisk firmy Laycock opisano niżej. Każde stanowisko jest obsługiwane przez jednego operatora, czas badań - 7 minut. W pamięci elektronicznej maszyny są gromadzone informacje otrzymywane od operatorów oraz z urządzeń automatycznej kontroli. Na linii wykonuje się następujące główne czynności kontrolne:

- stanowisko 1 - kontrola świateł oraz oględziny hamulców, ogumienia, układu przeniesienia napędu, mechanizmu kierowniczego, układu odprowadzania spalin, podwozia (ogumiska korozji),
- stanowisko 2 - kontrola hamulców,
- stanowisko 3 - kontrola geometrii kół i luzów układu kierowniczego,
- stanowisko 4 - kontrola wyważenia kół,
- stanowisko 5 - kontrola dynamiki pojazdu na hamowni podwoziowej (przyspieszenie i moment napędowy na kołach) oraz automatyczną kontrolą silnika przy automatycznej zmianie obrotów wału korbowego (obwody zasilania, zapłonu, rozruchu, stan świec, analiza spalin i moc silnika).

Wyniki badań otrzymuje się na końcu linii w formie wydrukowanej karty diagnostycznej.

● Przewidywano tendencje rozwojowe w automatyzacji diagnostyki samochodowej

Postęp w tej dziedzinie będzie w znacznej mierze uwarunkowany tempem elektronicznej diagnostyki samochodu i to zarówno w zakresie wprowadzenia nowych sprawnych i niezawodnych typów przetworników wielkości fizycznych na wielkości elektryczne, jak i elektronicznej instalacji układów: zapłonowego, paliwowego, hamulcowego i innych. Należy się spodziewać, że coraz więcej wytwórców będzie wyposażać pojazdy w centralne gniazdo grupujące sygnały od wszystkich czujników i przetworników oraz od innych mierzalnych bądź sterowalnych podzespołów samochodu. Gniazdo takie będzie instalowane przede wszystkim dla ułatwienia kontroli ostatecznej u producenta, ale będzie mogło być wykorzystywane i do badań diagnostycznych w czasie eksploatacji pojazdu.

Nie można również wykluczyć, że obserwowany kierunek wprowadzania mikroprocesorów do urządzeń nieprofesjonalnych dotrze również w skali masowej przed rokiem 1980 do samochodu, w którym zastosowanie mikrokomputerów umożliwi bieżącą kontrolę najważniejszych mierzalnych parametrów samochodu, poprawienie komfortu jazdy i jej bezpieczeństwa oraz zmniejszenie kosztów eksploatacji.

Oczywiście, wprowadzenie takiego sterowania podzespołami pojazdu umożliwi całkowicie badania diagnostyczne metodami tradycyjnymi, równocześnie ułatwi instalację centralnego gniazda do diagnostyki skomputeryzowanej.

Reasumując, można spodziewać się, że rozwój diagnostyki pojazdu będzie odbywał się równocześnie w dwu kierunkach: budowy skomputeryzowanych wielostanowiskowych stacji diagnostyki kompleksowej oraz programowalnych testerów diagnostycznych.

● Obecny stan opracowań w kraju

Jak wynika z prac PIMot i rozpoznania prowadzonego w IMM problemu komputeryzacji diagnostyki samochodowej w kraju znajduje się w fazie początkowej zarówno w zakresie koncepcji organizacyjnych jak również konstrukcji sprzętu i opracowania algorytmów diagnozy. W kraju są obecnie produkowane i eksploatowane analogowe zestawy i przyrządy do diagnostyki niektórych podzespółów samochodu.

Pomijając kwestię dokładności zasadniczymi wadami tych przyrządów, ograniczającymi możliwość ich wykorzystania w masowej diagnostyce samochodu są:

- czasochłonność badań,
- znaczny stopień subiektywności,
- konieczność odpowiedniego przygotowania zawodowego diagnosty,
- niewystarczający poziom niezawodności sprzętu

Równocześnie należy odnotować pewne próby automatyzacji stanowisk przeznaczonych do badań konstrukcyjnych i niezawodności niektórych elementów samochodu podejmowane przez wyższe uczelnie na terenie fabryk samochodów. Jednakże wyniki tych prac mogą być tylko w niewielkim zakresie wykorzystane w diagnostyce, ze względu na zupełnie odmienne podejście do badań samochodu i różny zakres i czasokres badań.

Proponowane rozwiązania mają charakter jednostkowy nieuniwersalny, opierają się na sprzęcie i technologii jeszcze obecnie wykorzystywanych, ale w skali lat osiemdziesiątych przestarzałych. Możliwość wykorzystania tych prac w diagnostyce istnieje, głównie w zakresie zastosowania niektórych czujników i przetworników pomiarowych i metod pomiaru wielkości fizycznych specyficznych dla samochodu oraz adaptacji oprzyrządowania mechanicznego stanowisk do badań układów, hamulcowego i jezdnego. Na wspomnianych stanowiskach należałoby prowadzić badania umożliwiające opracowanie algorytmów diagnozy poszczególnych zespołów samochodu. Tworzenie tych algorytmów powinno odbywać się z wyprzedzeniem w stosunku do konstrukcji stanowisk diagnostycznych, aby optymalnie dostosować ją do wymagań diagnozy.

3. Ogólna koncepcja systemu

● Wymagania ogólne

W warunkach krajowych w wyborze koncepcji budowy komputerowych systemów sterowania przeznaczonych do automatyzacji diagnostyki samochodowej należy uwzględnić następujące czynniki:

- stan i perspektywy rozwojowe przemysłów elektronicznego, automatyki i pomiarów, elektromaszynowego i precyzyjnego w aspekcie możliwości zastosowania najnowszych krajowych urządzeń, elementów i zespołów do budowy systemów diagnostycznych,
- obecny poziom techniczny i organizacyjny oraz perspektywy jego polepszenia w przemyśle motoryzacyjnym i na stacjach obsługi,
- możliwość zapewnienia odpowiednich mocy projektowych, produkcyjnych i serwisowych (w aspekcie kadrowym i ekonomicznym) do opracowania, produkcji, uruchomienia i eksploatacji systemów diagnostycznych.

Szczegółowa analiza wymienionych czynników oraz przesłanki techniczno-ekonomiczne prowadzą do wniosku, że krajowy system zautomatyzowanej diagnostyki samochodowej powinien spełniać następujące głównie wymagania:

- drupoziomowość struktury hierarchicznej sprzętu i programowania - w celu zapewnienia możliwości zestawienia systemu z autonomicznymi stanowiskami (testerami) wyspecjalizowanymi do badań określonych podzespółów funkcjonalnych samochodu, przy czym każdy z takich testerów powinien również gwarantować możliwość użytkowania go pojedynczo poza systemem,

- modularność budowy sprzętu i oprogramowania obydwu poziomów - w celu zapewnienia na wyższym poziomie możliwości dołączania dowolnej liczby (w pewnych granicach) testerów różnego przeznaczenia, a na poziomie niższym - składania różnych testerów z pewnego zbioru podstawowych modułów dla ułatwienia produkcji, eksploatacji i modernizacji testerów,
- możliwość etapowego tworzenia systemu od dowolnego poziomu i dowolnego testera tzn. od badań najbardziej "dojrzałych" do automatyzacji badań diagnostycznych,
- dostosowanie się w projekcie systemu do istniejącego i produkowanego standardu modułów, optymalnego z punktu widzenia odpowiednio szerokiego asortymentu, poziomu nowoczesności, perspektyw produkcyjnych i rozwojowych. Umożliwi to skrócenie czasu realizacji systemu oraz redukcję kosztu opracowania, produkcji i eksploatacji systemu.

Wymieniono wyżej wymagania spełnia dwupoziomowy, hierarchiczny system komputerowego sterowania pokazany na rys. 1. System ten umożliwia budowę skomputeryzowanych linii diagnostycznych, złożonych z wielu specjalistycznych stanowisk sterowanych lokalnie w sposób autonomiczny i centralnie koordynowanych za pomocą minikomputera nadrzędnego. Taki system ma bardzo wiele cennych właściwości nadających mu cechy systemu uniwersalnego dla dość szerokiej klasy procesów sterowania:

- łatwość uzyskiwania konfiguracji optymalnej dla danego zadania,
- dużą elastyczność i łatwość rozbudowy i modernizacji,
- odporność systemu na awarie komputera nadrzędnego (możliwość przejścia na sterowanie zdecentralizowane),
- łatwość zwiększenia niezawodności dla określonego fragmentu procesu na wypadek awarii autonomicznego sterownika przez dostawienie drugiego (zdublowanie sygnałów i sterowań),
- ekonomiczność w wypadku stosowania zasady zimnej rezerwy - zamiast rezerwować kosztowny komputer rezerwujemy tylko tańszy blok (np. sterownik z mikroprocesorem, PAO itp.),
- możliwość znacznej redukcji długości kabli, a więc i ich kosztu w następstwie decentralizacji systemu i wykorzystania sterowników autonomicznych jako koncentratorów danych przy źródłach sygnałów.

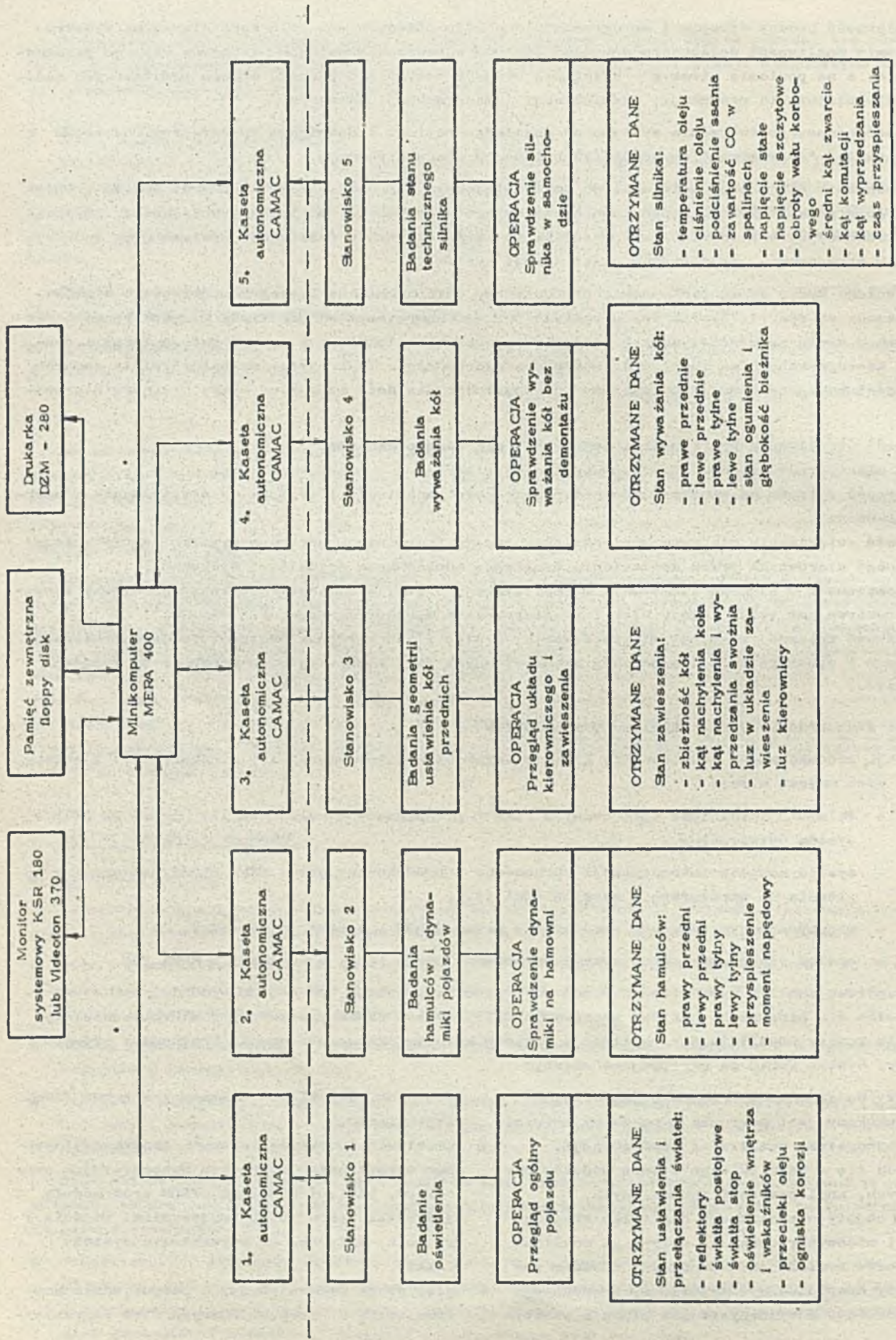
• Wybór standardów modułów współpracy z obiektem

W kraju produkowanych jest obecnie kilka standardów modułów współpracy minikomputera z obiektem. Do nich należą m.in.:

- PI - Polmatic Inteldigit opracowany w PIAP i produkowany w małych seriach przez ZD PIAP - system uniwersalny,
- SMA - system modułów automatyzacji opracowany i produkowany przez MERA ELMAT nakierowany głównie na współpracę z maszyną ODRA 1325,
- CAMAC - międzynarodowy standard produkowany przez POLON - system uniwersalny,
- NUCON 400 - system sterowania obrabiarkami opracowany przez ASEA, a wdrażany w MERA ZSM.

Szczegółowa analiza wymienionych standardów modułów wykazuje, że najodpowiedniejszym standardem modułów dla pierwszego poziomu sterowania byłby system CAMAC wyposażony w mikroprocesorowy sterownik kasety umożliwiający realizację układów autonomicznego sterowania fragmentem procesu diagnozy. System CAMAC ma następujące zalety:

- należy do standardu krajowego POLMATIC,
- produkowany jest seryjnie i ma dobry serwis,
- ma najbogatsze możliwości konfiguracji,
- składa się z największego wyboru modułów cyfrowych, analogowych i pomocniczych,
- jest objęty normalizacją nie tylko krajową ale i międzynarodową, co wpływa na możliwość eksportu systemu oraz umożliwia w razie potrzeby kompletowanie systemu opracowywanego jednostkowo z wykorzystaniem bloku z asorty-
- mentu ok. 800 bloków oferowanych przez firmy zagraniczne,
- umożliwia tworzenie zestawów programowalnych bez zewnętrznego komputera wykorzystując procesory, pamięci ROM, RAM, PROM oraz moduły jednostek sterujących urządzeniami wejścia i wyjścia, należące do wyposażenia systemu CAMAC,
- jest stale modernizowany i bezpośrednio konfrontowany z wyrobami znanych firm zagranicz-



Rys. 1. Minikomputerowy system diagnostyki samochodów osobowych

nych, co gwarantuje stale wysoki poziom nowoczesności,

- prowadzono się prace nad standaryzacją oprogramowania kaset CAMAC.

• Wybór minikomputera systemowego

Decydujący wpływ na wybór typu minikomputera mają w zasadzie zagadnienia programowe. Natomiast wymagania, które zwykle stawia się przed sprzętem komputerowym stosowanym w systemach sterowania dla minikomputera pracującego w systemie hierarchicznym poziomu nadrzędnego można uszereżować w poniższej kolejności:

- wieloprogramowanie
- wielodostępność
- wielopoziomowy układ przerwań
- obszerna PAO z układem ochrony zawartości
- modularność konstrukcji i oprogramowania
- moc obliczeniowa i skuteczność listy rozkazów,
- odpowiedni poziom niezawodności.

Inna waga i kolejność tych wymagań niż dla m.c. w systemach sterowania jednopoziomowego lub sterowania poziomem podrzędnego wynika ze zmienionych proporcji, funkcji sterowania do funkcji przetwarzania czy zarządzania procesem. Wpływa to również na miejsce instalacji tego minikomputera, które w tym wypadku nie musi znajdować się w pobliżu sterowanych obiektów np. agregatów technologicznych, ale może to być miejsce stwarzające dogodniejsze warunki mechaniczne i klimatyczne wpływające z przekazania bezpośredniego sterowania kasetom autonomicznym. Również zagadnienie pracy w czasie rzeczywistym jak i poziom niezawodności nie są tak krytyczne jak w systemach klasycznych.

Modularność konstrukcji minikomputera umożliwia uzyskanie konfiguracji minikomputera optymalnie dostosowanej do zadań występujących w danym systemie, zapowiadając w razie potrzeby zmianę jego struktury (liczbę i rodzaj kanałów i jednostek sterujących wielkości PAO itp.). Równocześnie uzyskuje się istotnie ułatwienie konserwacji minikomputera oraz skrócenie czasu naprawy przez możliwość wymiany uszkodzonego modułu.

Łączna analiza wymagań sprzętowych i programowych prowadzi do wniosku, że obecnie minikomputerem systemowym może być minikomputer MERA 400 lub minikomputer SM-03.

• Wybór sposobu sprzężenia minikomputera systemowego z kasetą autonomiczną CAMAC

W wyborze sprzężenia minikomputera nadrzędnego z kasetą autonomiczną CAMAC kierowano się następującymi względami:

- wykorzystanie w maksymalnym stopniu istniejących kanałów i jednostek sterujących w minikomputerze i odpowiednich bloków pośredniczących w kasecie autonomicznej,
- charakter zastosowanego sprzężenia powinien być na tyle uniwersalny, żeby przejście na inny typ minikomputera nie wymagało żadnych zmian w kasecie autonomicznej.

Zagadnienie szybkości przekazywania informacji nie musi być krytyczne. W wyniku dekompozycji procesu (obiekty) na podprocesy (podobiekty) sterowane lokalnie przez kasety automatyczne w odpowiednich cząstkowych programach sterujących nie zachodzi na ogół potrzeba terminowego przekazywania dużych bloków informacji między minikomputerem systemowym a kasetami. Będą to najczęściej pojedyncze przesłania, na przykład o takim charakterze jak niżej wymieniono.

Kierunek do MERA 400

- prośba o nowy parametr sterowania
- wysłanie żądanej informacji
- żądanie przetworzenia danej według określonego algorytmu
- meldunek alarmowy

Kierunek z MERA 400

- wysłanie nowego parametru sterowania
- wysłanie wyniku przetworzenia
- żądanie określonej informacji
- powiadomienie o awarii

Ponadto najczęściej program sterujący kasetą będzie programem rezydującym w pamięci typu ROM lub PROM, a więc nie będzie potrzeby wymiany programów.

W efekcie przeprowadzonych analiz ustalono, że najwygodniejszym, powszechnie wykorzystywanym obecnie interfejsem byłby szeregowy interfejs synchroniczny występujący w dniu odmianach nieco różniących się (głównie poziomem sygnałów), a mianowicie jako interfejs prądowy dla teletypów np.

ASR33 oraz jako interfejs V-24 wykorzystywany również do łączności na duże odległości za pomocą modemów.

Analiza dokumentacji wykazała, że dopasowanie układów wejścia/wyjścia pracujących różnymi poziomami sygnałów nie przedstawia specjalnych trudności. Opierając się na powyższych przesłankach proponuje się zrealizowanie sprzężenia MERA 400 - CAMAC przez połączenie JS - V-24 pracującej w KZ MERA 400 z blokiem typu 500 w kasecie CAMAC. Zarówno JS - V-24 jak i blok 500 służą do dwukierunkowej bajtowej transmisji szeregowej.

Niezbędne adaptacje polegają na:

- dopasowaniu poziomu sygnałów obydwu jednostek przez instalacje odpowiednich dzielników na pakietach (wykonanych tak, aby powrót do standardowej pracy wymagał jedynie przełączenia odpowiednich zwór na pakiecie),
- dopasowaniu częstotliwości zegarów obydwu jednostek (wymaga tylko ustawienia odpowiednich zworek na pakietach i regulacji potencjometrom); bez dalszych przeróbek maksymalnie możliwa, ze względu na blok 500 częstotliwość zegara wynosi 40 kHz,
- wykonaniu odpowiedniego kabla 4-torowego dla połączenia obydwu jednostek. Rodzaj kabla (tory koncentryczne czy dwuskrotki) jest zależna od odległości między jednostkami i odporności na zakłócenia. Przewiduje się, że dopuszczalna odległość nie powinna być mniejsza od 100 m.

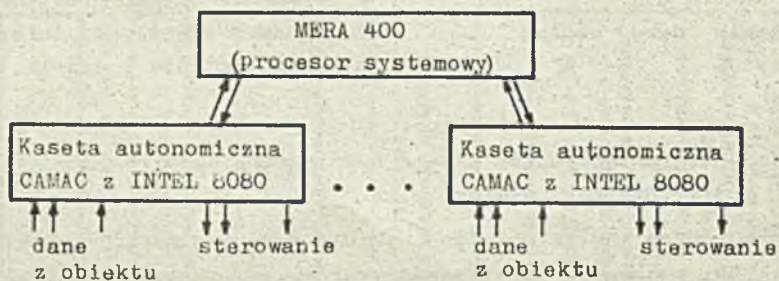
Oprogramowanie takiego sprzężenia po stronie MERA 400 nie jest problemem ze względu na wykorzystanie istniejącego kanału (KZ) i istniejącej JS - V-24. Po stronie kasety CAMAC niezbędny jest program obsługi, najlepiej umieszczony w pamięci ROM lub PROM. Warto podkreślić, że ten typ sprzężenia redukuje znacznie koszt kabli, równocześnie umożliwiając nawet przy transmisji bez wykorzystania modemów uzyskanie zasięgu około 1 km (kablem współosiowym). Ponadto, w wypadku zastosowania sterownika autonomicznego wyposażonego w mikroprocesor INTEL 8080 np. sterownika CMC 8080, możliwe są oprócz wyżej opisanej jeszcze inne realizacje sprzężenia, a mianowicie:

- za pośrednictwem interfejsu teletype sterownika; jest ono identyczne jak poprzednie z punktu widzenia MERA 400 ale znacznie dogodniejsze z punktu widzenia oprogramowania kasety, gdyż umożliwia wyzyskanie dość wysokiego priorytetu jaki zgłasza TTY do mikroprocesora,
- przez interfejs DMA sterownika, co umożliwia równoległe przesłanie bajtowe i transmisję bardzo szybką, ale wymaga budowy specjalnych jednostek sterujących po stronie minikomputera i po stronie kasety oraz całkowicie nowego oprogramowania dla jego obsługi.

Przy wyżej przyjętych założeniach, konfiguracja sprzętowa komputerowego systemu sterowania linią diagnostyczną będzie składać się w dużym stopniu z modułów o charakterze uniwersalnym, produkowanych w kraju.

4. Koncepcja oprogramowania systemu

W proponowanym systemie kompleksowej automatyzacji diagnostyki samochodowej procesor systemowy MERA 400 zarządza pracą pięciu procesorów autonomicznych pracujących równolegle, jest też jednocześnie "centrum" przetwarzania i rejestracji danych. Procesory autonomiczne natomiast, spełniają tu rolę bezpośrednich "sterowników", bądź też tylko rejestratorów zbierających i ewentualnie wstępnie przetwarzających dane z procesu (rys. 2).

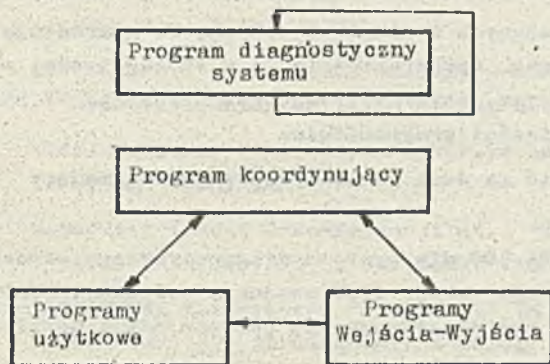


Rys. 2. Schemat systemu

● Oprogramowanie pierwszej warstwy bezpośredniego sterowania

Oprogramowanie pierwszej warstwy (kasyety CAMAC z mikroprocesorowym sterownikiem CMC 8080) powinno składać się z czterech podstawowych modułów (rys. 3):

- programu koordynującego,
- programów wejścia/wyjścia
- programów diagnostycznych systemu;
- programów użytkowych.



Rys. 3. Wzajemne powiązanie modułów systemu warstwy bezpośredniego sterowania

Programy diagnostyczne systemu będą kontrolować poprawność pracy mikroprocesorowego systemu pracującego na poziomie pierwszym. Programy te wykonywane będą w czasie, gdy będzie mała intensywność zgłoszeń do obsługi. Mają one zwykle najniższy priorytet.

Programy użytkowe będą obsługiwały zgłoszenia. Do zgłoszeń wymagających specjalnego oprogramowania należą:

- zgłoszenie od operatora maszyny,
- zgłoszenie pochodzące od operatora procesu (mogą one być związane z aktywizacją punktów pomiarowych),
- zgłoszenie od maszyny zarządzającej (zgłoszenie od maszyny MERA 400),
- grupa zgłoszeń zależnych od sytuacji w sterowanym procesie (zgłoszenia od bloków CAMAC).

Programy dla sterownika CMC 8080 można pisać w języku PL/M. Kompilator tego języka, jak wynika z opisów, obniża efektywność programów tylko w 10%. Poza tym możliwość pisania podprogramów w assemblerze tego procesora zmniejsza w znacznym stopniu ten procent. Kompilator tego języka napisany jest w języku FORTRAN Standard, a więc możliwa jest jego implementacja na maszynie poziomu nadrzędnego (MERA 400).

Oprogramowanie CMC 8080, w tym wypadku nie jest bardzo skomplikowane, dlatego też w małym stopniu zwiększając pracochłonność można cały system napisać w assemblerze INTEL 8080. Assembler ten spełnia wszystkie podstawowe wymagania stawiane zwykle językom, które mają być wykorzystane do sterowania.

Właściwości sprzętowe i programowe CMC 8080:

- bogata lista instrukcji (74 instrukcje)
- rejestry (6 szt. - większa liczba rejestrów ogólnego przeznaczenia zwiększa w znacznym stopniu elastyczność oprogramowania),
- adresowanie - 4 sposoby adresowania:
 - bezpośrednio
 - natychmiastowe
 - rejestrowane przez wskaźnik stosu
 - rejestrowane przez parę rejestrów,
- przerwanie (możliwość organizacji systemu przerwania wielopoziomowo i wielowektorowo),
- programowany zegar czasu rzeczywistego,
- UART dla transmisji szeregowej,
- szybkość działania (najkrótszy czas wykonywania instrukcji 2 ns, najdłuższy 9 ns),
- pakiety użytkowe (możliwość dołączenia dodatkowych pakietów takich jak: pakiet zmiennego przecinka, mnożenie, dzielenie).

● Oprogramowanie procesora systemowego

Minikomputer MERA 400 - będący w omawianym systemie procesorem systemowym - jest uniwersalną 16-bitową maszyną cyfrową o architekturze odpowiadającej wymaganiom stawianym współczesnym systemom cyfrowym. Rozbudowana lista rozkazów zapewnia efektywne działanie na danych o różnorodnej postaci.

Maszyna ta wyposażona jest w zestaw programów testujących i diagnostycznych, co gwarantuje użytkownikowi systemu sprawny serwis i obsługę techniczną. Oprogramowanie, a w szczególności wieloprogramowy i wielodostępny system operacyjny SOM-3 oraz języki wyższego rzędu FORTRAN IV, BASIC, wsparte obszerną biblioteką programów, daje duże możliwości programiście.

Oprogramowanie procesora systemowego można podzielić na dwie grupy: program zarządzający i zbiór programów użytkowych zwanych dalej zadaniami.

Z istniejących już wersji systemów operacyjnych MERA 400 dla systemu diagnostycznego samochodów proponuje się wykorzystać system SOM-3 w wersji CORE. Wersja CORE obejmuje wszystkie konfiguracje pozbawione zewnętrznych pamięci magnetycznych i charakteryzuje się następującymi możliwościami:

- stała liczba zadań ustalona w czasie generacji systemu (tylko zadanie roz/dujące),
- procedura planująca system dzieli czas maszyny między poszczególne zadania, zgodnie z priorytetową kolejką prac systemu,
- zadania mogą być wykonywane na jednym z 256 dopuszczalnych poziomów priorytetu (każdy poziom dopuszcza istnienie własnej kolejki zadań),
- możliwość tworzenia dowolnej konfiguracji bloków pamięci (tzw. podział softwarowy),
- ochrona pamięci w wypadku lokalizacji jednego zadania w bloku (dopuszcza się lokalizowanie idealu zadań w bloku, sprawdzonych pod względem poprawności),
- system zapewnia standardową obsługę przerw (użytkownik ma możliwość dołączania własnej obsługi przerw specjalnych - działającej pod kontrolą systemu),
- ekstrakody wielodostępne (dopuszczalne do wykorzystania we wszystkich zadaniach), możliwość dołączania własnych ekstrakodów (działać one będą pod kontrolą systemu),
- specjalny podsystem wejścia/wyjścia (każdy typ urządzenia ma zindywidualizowaną procedurę obsługi zwaną handlerem), możliwość dołączenia własnych handlerów działających pod kontrolą podsystemu wejścia/wyjścia,
- operacje wejścia/wyjścia mogą wstrzymywać bieg zadania lub wykonywać się współbieżnie z nim (time sharing), dopuszcza się równoległe prowadzenie wielu operacji wejścia/wyjścia (wielodostępność) w jednym zadaniu,
- zadania mogą być sterowane przez operatora (startowanie, zawieszanie, odrzucanie, zmiana priorytetu),
- standardowa obsługa przerw zegarów ch, możliwość dołączania monitora czasu rzeczywistego kontrolującego systemowy zegar, możliwość rozszerzenia monitora czasu rzeczywistego o dowolną liczbę lokalnych zegarów programowych (działających jako timery lub budziki),
- buforowanie wszystkich operacji wejścia/wyjścia (obsługa błędów operacji własna lub pod kontrolą systemu),
- system dopuszcza standardowe i niestandardowe, binarne i znakowe typy organizacji danych.

Wersja CORE systemu SOM-3 spełnia wszystkie wymagania, nałożone na system operacyjny dla procesora systemowego dla stanowiska diagnostycznego samochodów. Zatem po dołączeniu handlera kasety CAMAC MERA 400 implementacja tego systemu dla tych potrzeb nie wymagałaby żadnych dodatkowych zmian.

Literatura

- [1] Podstawy sterowania 1974 t. 4 z. 3
- [2] NINDEISEN W.: Wielopoziomowe układy sterowania i koordynacji. Warszawa: Politechnika Warszawska 1976

- [3] GOŚCIŃSKI A.: Oprogramowanie maszyn cyfrowych do sterowania procesów przemysłowych. Kraków: AGH 1975
- [4] SZMYD J.: Raport z zakresu programu rozwoju komputerowych systemów automatyzacji badań i pomiarów diagnostycznych. Warszawa: oprac. wewn. IMM 1976
- [5] MAJEWSKI Z.: Opracowanie danych wyjściowych do badań diagnostycznych samochodów osobowych. Warszawa: PINot 1975.
- [6] MAJEWSKI Z.: Opracowanie analizy w zakresie stosowanych metod diagnostycznych ze szczególnym uwzględnieniem metod elektronicznych. Warszawa: PINot 1974
- [7] WIŚNIEWSKI Z.: Perspektywy wykorzystania zestawów CAMAC do sterowania procesami. Warszawa: oprac. wewn. IMM 1975
- [8] Materiały sekcji 2 konferencji nt.: "Elektronizacja systemów automatyki przemysłowej". Katowice 9-10.IX.1976 r. Warszawa: PIE 1976
- [9] WIŚNIEWSKI Z.: Wystawa "Fizyka 75" (Moskwa). Niektóre aspekty sprzętowe i systemowe. Warszawa: IMM 1975 sprawozdanie z deleg. zagr.
- [10] HILDEBRANDT K., LEHL E.H.: Zastosowanie komputera do sterowania procesami przy stanowisku kontroli silników. Warszawa: IMM 1974, tłumaczenie - oprac. wewn.
- [11] WIŚNIEWSKI Z., FILIPIŃSKI J.: Architektura i możliwości techniczne systemu MERA 400 z punktu widzenia sterowania procesami w szczególności szybkozmiennymi. Warszawa: IMM 1974 opracowanie wewn.
- [12] JANKOWSKI T., WIŚNIEWSKI Z.: Projekt założeń na moduł sprzęgający minikomputer MERA 400 z systemem CAMAC. Warszawa: IMM 1975 opr. wewn.
- [13] MERA 400. Opis użytkowy. Tom I. Warszawa: 1976

mgr inż. Tadeusz CIEŁSTOWSKI
mgr inż. Grzegorz SZYMSIAK
Instytut Maszyn Matematycznych

ASTER - SYSTEM TERMINALOWY NA MINIKOMPUTERZE SERII MERA 300

Wstęp

W Pracowni Minikomputerowych Systemów Terminalowych Zakładu Systemów Wyszukiwania i Teleprzetwarzania Informacji IMM opracowano terminal ASTER-M300-A (ASynchroniczny TERminal - MERA-300 - wersja A). Prace nad terminalem podjęto z myślą o zastosowaniu w konwersacyjnym systemie wyszukiwania informacji KWINTET współpracującym z systemem administracji bazą danych - SAD. Obydwa te systemy są tworzone w pozostałych pracowniach Zakładu SWITI. Nie był to jedyny powód podjęcia prac nad terminalem.

Narastające w kraju zainteresowanie teleprzetwarzaniem, wywołane m.in. niedaleką już perspektywą wyposażenia komputerów R-32 w programowany multipleksor transmisji danych, powoduje wzmożone zapotrzebowanie na terminale. Jednocześnie stwierdzono, że wykorzystywane przez wielu użytkowników minikomputery serii MERA 300 mogą być programowo przystosowane do pracy jako terminale, po niezbyt kosztownym rozszerzeniu konfiguracji sprzętowej. Istnieją również wśród niektórych użytkowników potrzeby powiązania lokalnego przetwarzania danych (np. z wydziału produkcyjnego) z centralnym przetwarzaniem informacji (np. dla celów zarządzania całym przedsiębiorstwem). Powyższe przesłanki były bodźcem do opracowania programu dostosowującego dowolny minikomputer serii MERA 300 do zdalnej współpracy z komputerem R-32 lub komputerem IBM systemu 360/370. Wybór typu terminala i sposobu współpracy z komputerem centralnym był uzależniony od możliwości oprogramowania komunikacyjnego komputerów JS EMC oraz ograniczeń programowych i sprzętowych minikomputerów serii MERA 300.

W artykule omówiono budowę i zasadę działania terminala ASTER-M300-A, a także jego program sterujący. Na zakończenie poruszono również możliwości budowy innych typów końcowych stacji zdalnego przetwarzania danych, oparte na rozwiązaniach stosowanych w programie ASTER.

Dane ogólne terminala ASTER-M300-A

Terminal ten powstał przez przystosowanie sprzętowe i programowe minikomputera MERA 300 do pracy asynchronicznej w systemach zdalnego przetwarzania danych JS EMC i IBM. Przystosowanie sprzętowe polegało na dołączeniu do konfiguracji minikomputera jednostki sterowania transmisją asynchroniczną oraz modemu. Przystosowania programowe dokonano przez opracowanie i zainstalowanie w minikomputerze programu sterującego o nazwie ASTER.

ASTER-M300-A ma takie same funkcje i algorytm wymiany informacji z maszyną główną, jak terminal IBM 2740 model 2 w wersji z buforowanym odbiorem i kontrolą poprawności bloków danych (patrz: "IBM 2740 Communication Terminal Models 1 and 2 Component Description" - GA24-3403-4). Ma on bufor nadawczo-odbiorczy o standardowej pojemności 248 znaków i współpracuje z maszyną główną metodą transmisji asynchronicznej, naprzemiennej (półdupleks) na łączach stałych (niekomutowanych). Czytanie informacji z bufora nie niszczy jego zawartości.

Standardowa szybkość transmisji wynosi 600 b/s.

Program ASTER umożliwia dokonanie zmiany parametrów standardowych terminala, a mianowicie:

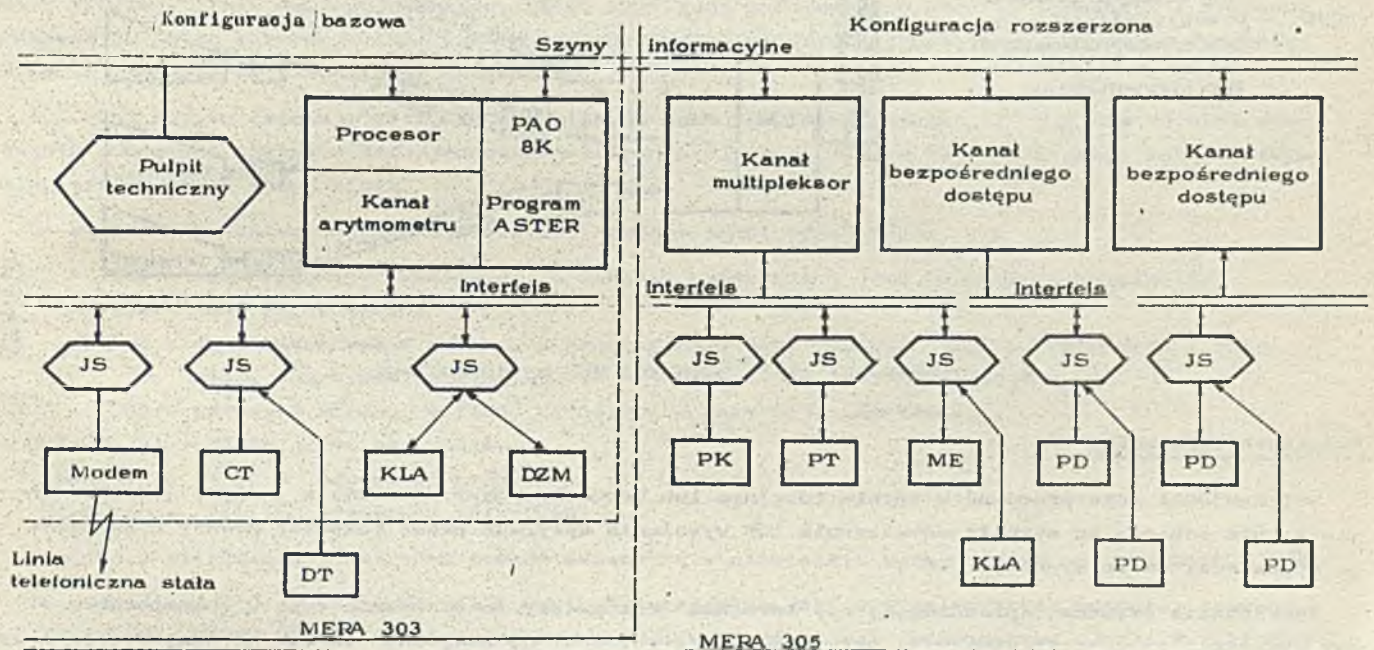
- zmianę szybkości transmisji,
- cechowanie terminala jako nadrzędnego w grupie lub na linii,
- zmianę okresów odliczanych czasów (np. ustawięcia odbioru i nadawania),
- zmianę pojemności bufora,
- zmianę usytuowania bufora w PAO,
- zmianę długości transmitowanych bloków informacji,
- zmianę indywidualnego lub grupowego adresu terminala.

Do pracy minikomputera MERA 300 jako stacji końcowej ASTER-M300-A wymagana jest następująca minimalna (bazowa) konfiguracja sprzętowa:

- jednostka centralna z pamięcią operacyjną (8 K bajtów),
- drukarka DZM 180 z klawiaturą,
- czytnik taśmy dziurkowanej,
- zespół jednostek sterujących urządzeniami zewnętrznymi,
- jednostka sterująca transmisją (MOM-DS 11 lub 12),
- modem 600/1200 Bd z interfejsem wg zaleceń V24 CCITT.

Jako urządzenie wprowadzania-wyprowadzania danych zastosowana jest drukarka DZM 180 z klawiaturą.

Konfiguracje minikomputera MERA 300, w których może pracować program ASTER pokazane są na rysunku 1.



Objaśnienia:

- JS - jednostka sterująca
- CT - czytnik taśmy papierowej
- DT - dziurkarka taśmy papierowej
- ME - monitor ekranowy
- KLA - klawiatura alfanumeryczna
- DZM - drukarka znakowo-mozaikowa
- PK, PT, PD - pamięć kasetowa, taśmowa, dyskowa

Rys. 1. Konfiguracje minikomputera MERA 300

ASTER-M300-A jest przewidziany do pracy w sieciach terminalowych, w których maszyną główną jest komputer jednolitego systemu elektronicznych maszyn cyfrowych (JS EMC) lub komputer systemu IBM 360/370. Maszyna główna musi być wyposażona w jednostkę sterowania transmisją danych, którą może być multipleksor transmisji danych (np. typu IBM-2701, IBM-2702 lub IBM-2703) lub procesor komunikacyjny (np. typu IBM-3704 lub IBM-3705).

Terminal może być dołączony również do zdalnego multipleksora (np. typu IBM 2712), który z kolei musi być, przez kanał łączności, dołączony do przymaszynowej jednostki sterowania transmisją danych. Może on pracować z multiplekseorem (procesorem komunikacyjnym) w układzie dwupunktowym lub w układzie wielopunktowym, gdy kilka lub kilkanaście terminali przyłączonych jest do tej samej linii.

Przykład schematu sieci połączeń terminali z maszyną główną pokazany jest na rys. 2.

komunikacyjnej i wywołaniu jej przez komputer. Dane z linii mogą być odebrane do bufora, gdy terminal będzie zadresowany przez komputer i znajduje się w stanie gotowości w trybie komunikacyjnym.

Bez błędnie odebrany blok danych jest po zakończeniu transmisji automatycznie wydrukowany na papierze. Jeżeli blok będzie odebrany z błędami, wydruk zawartości bufora następuje na żądanie operatora.

Ręczne sterowanie terminalem dokonywane jest za pomocą kluczy na pulpicie technicznym minikomputera oraz kluczy systemowych i klawiszy specjalnych na klawiaturze drukarki DZM-180.

Do sygnalizacji stanu pracy terminala użyte są lampki systemowe drukarki DZM-180.

Zasady sterowania terminalem przez komputer główny

Transmisja od komputera do terminala lub odwrotnie jest zawsze inicjowana przez komputer. Komputer wysyła w linię sekwencje znaków, które zawierają informacje o adresie terminala i kierunku transmisji. Są to sekwencje adresowania (addressing) i wywoływania (polling), które poprzedzają transmisję tekstu do i od stacji końcowej.

Ponieważ często istnieje potrzeba komunikacji typu rozgłoszeniowego, stosuje się również adresowanie grupowe i ogólne w celu jednoczesnej transmisji tego samego tekstu do grupy lub do wszystkich terminali na danej linii.

Do sterowania transmisją stosuje się następujące znaki specjalne:

- Ⓒ - EOT - znak kończący transmisję, a zarazem ustawiający stan początkowy terminala;
- Ⓓ - EOB - znak końca bloku;
- Ⓓ - EOA - znak zakończenia adresu, a jednocześnie znak początku tekstu komunikatu i odpowiedź pozytywna na wywołanie;
- Ⓔ - SOA - początek adresu, używany wyłącznie w sekwencji adresowania.

Ponadto do sterowania używa się znaków:

- "-" (minus) - Ⓔ - odpowiedź negatywna
- "." (kropka) - Ⓕ - odpowiedź pozytywna.

Procedury sterowania terminalem przedstawione są w niewielkim uproszczeniu na rysunkach 3 i 4.

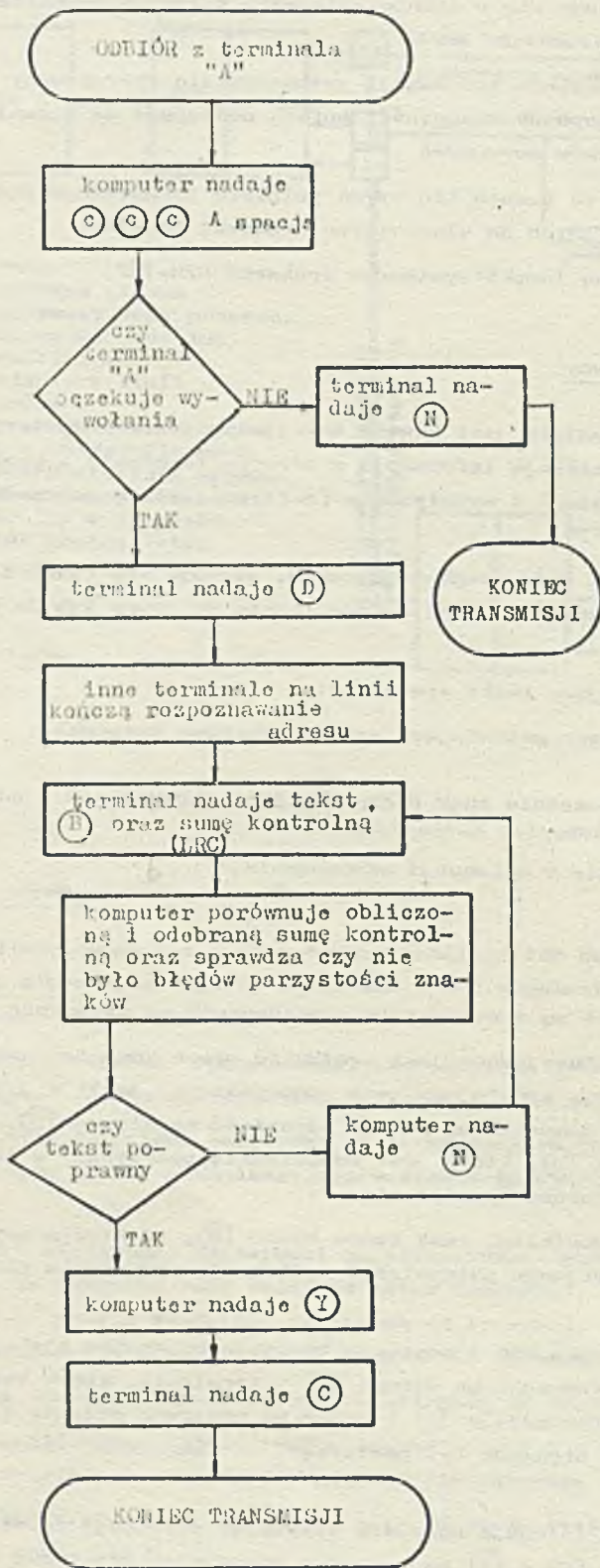
Transmisja danych od terminala do komputera inicjowana jest wysłaniem przez komputer sekwencji wywołania. Odpowiedź stacji końcowej na sekwencję wywołującą jest jednoznakowa. Jeśli z jakiegokolwiek względów nie jest ona gotowa do nadawania komunikatu, nadaje odpowiedź negatywną Ⓔ. Jeżeli terminal oczekiwał na wywołanie, to nadaje znak Ⓓ, który jest odpowiedzią pozytywną, a zarazem sygnalizuje innym terminalom na linii, że zakończono wywoływanie.

Wywołany terminal nadaje następnie tekst komunikatu, znak końca bloku Ⓓ, po którym następuje nadanie obliczonego przez siebie znaku kontrolnego parzystości wzłużnej, tzw. znaku LRC (Longitudinal Redundancy Check).

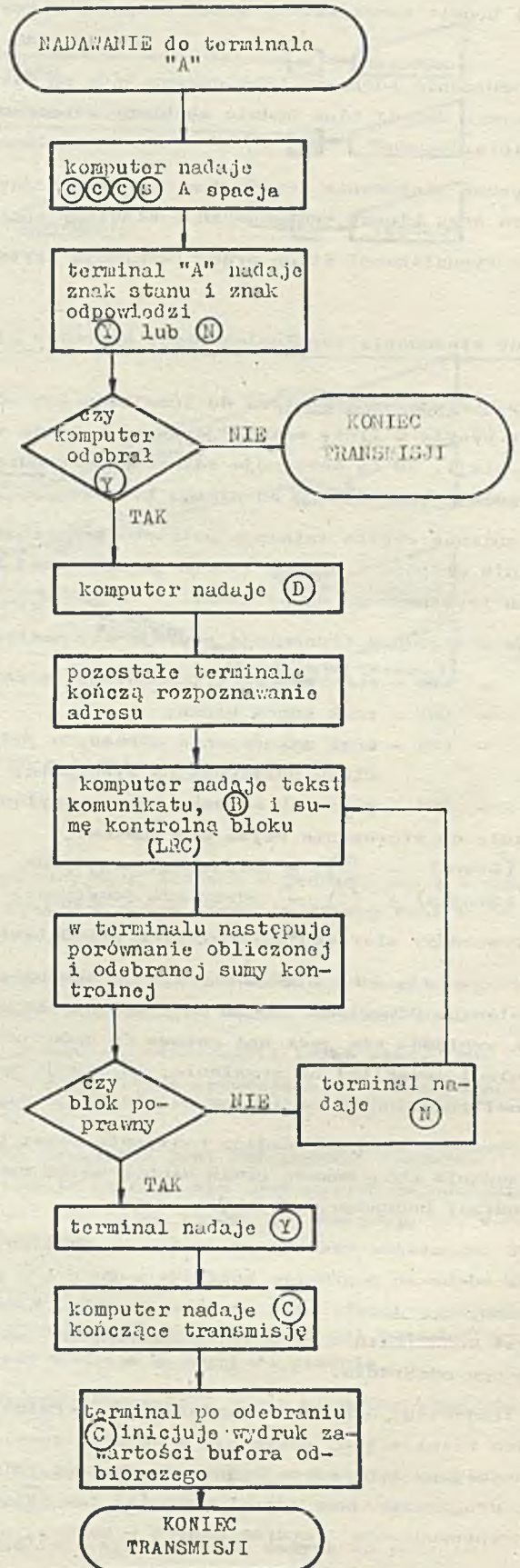
W komputerze centralnym następuje analiza znaku LRC i kontrola parzystości znaków bloku. Jeśli blok odebrano poprawnie komputer nadaje Ⓕ i oczekuje na sygnał Ⓒ z terminala, który kończy transmisję. Jeżeli blok ma przekłamanie, komputer nadaje Ⓔ i czeka na powtórne nadanie tekstu. Tekst komunikatu wraz z Ⓓ i znakiem LRC może niekiedy być powtarzany wielokrotnie, aż do prawidłowego odebrania.

Transmisje danych od komputera do terminala inicjuje komputer wysyłając sekwencję adresowania, która różni się od sekwencji wywołania znakiem SOA - Ⓔ na początku sekwencji. Odpowiedź stacji końcowej na tę sekwencję jest dwuznakowa. Pierwszy znak, tzw. znak stanu, sygnalizuje stan terminala, drugi znak jest odpowiedzią Ⓕ lub Ⓔ. Jest możliwe rozpoznanie stanów terminala takich, jak wprowadzanie i wydruk danych z bufora, oczekiwanie na wywołanie, brak papieru w drukarce itp.

Jeśli terminal dał odpowiedź pozytywną, następuje do niego transmisja danych. Komputer nadaje znak Ⓓ, który spełnia analogiczną funkcję, jak przy wywołaniu. Następnie nadany jest tekst ko-



Rys. 3. Transmisja terminal - komputer



Rys. 4. Transmisja komputer - terminal

komunikatu, znak (H) i znak LRC. Przy odbiorze stacja końcowa bada parzystość każdego przychodzącego znaku, a następnie poprawność sumy kontrolnej (LRC). Jeśli blok był odebrany poprawnie, wtedy terminal nadaje odpowiedź (Y). Po odebraniu z komputera znaku kończącego transmisję nastąpi wydrukowanie z bufora odebranego poprawnie komunikatu.

W wypadku wykrycia błędu terminal nadaje (N). Komputer powtarza wtedy nadawanie tekstu ze znakiem (H) i znakiem LRC. W ten sposób komputer będzie powtarzał komunikat, aż do otrzymania pozytywnej odpowiedzi. Może on przerwać powtarzanie nadając znak (C) kończący transmisję. Wydruk komunikatu na stacji końcowej wówczas nie nastąpi automatycznie, lecz na żądanie operatora. Możliwa jest jeszcze taka sytuacja, że komunikat jest zbyt długi i nie mieści się w buforze terminala. Stacja końcowa przerywa wtedy transmisję nadając znak (C) po zapełnieniu bufora.

Możliwe jest przesyłanie komunikatu z terminala do komputera, a następnie w kierunku odwrotnym po jednym zainicjowaniu transmisji. Dzieje się tak, gdy po odebraniu komunikatu z terminala komputer ma do nadania komunikat do tej samej stacji końcowej. Wtedy zamiast odpowiedzi pozytywnej (Y) komputer nadaje (D), a następnie tekst komunikatu, znak (H) i znak LRC. Dalsza współpraca terminala z komputerem przebiega tak, jak po adresowaniu.

W wypadku adresowania grupowego lub ogólnego konwersację z komputerem prowadzi wyłącznie terminal nadrzędny w grupie lub nadrzędny na danej linii. Pozostałe terminale tylko odbierają to, co nada komputer i "nasłuchują" odpowiedzi terminala nadrzędnego. Sekwencja adresowania grupowego jest analogiczna do sekwencji adresowania indywidualnego, tylko w miejscu adresu indywidualnego terminala znajduje się adres grupowy. Adresem ogólnym jest znak "/", (znak kreski ukośnej - slash).

Program sterujący ASTER

W obecnej wersji ASTER jest wykonany jako program samodzielny. Zajmuje on w pamięci operacyjnej PAO minikomputera MERA 300 obszar około 3250 bajtów. W obszarze tym mieści się również bufor komunikacyjny o pojemności 248 znaków (bajtów). Na początku obszaru PAO, w którym umieszczony jest program ASTER, wydzielone są 3 strony pamięci; na nich mieszczą się wszystkie rejestry, liczniki i wskaźniki programowe. Te wydzielone strony zwane są stronami sterującymi programu ASTER.

Rejestry składają się z 1 lub 2 bajtów i służą do przechowywania stałych parametrów programu takich, jak:

- adres początku bufora komunikacyjnego,
- pojemność bufora,
- długość komunikatu (przesyłanego bloku informacji),
- indywidualny i grupowy adres terminala itp.

Ponadto w niektórych rejestrach przechowywane są chwilowo takie informacje, jak:

- adres pierwszej wolnej pozycji bufora komunikacyjnego,
- znak odebrany z linii lub z klawiatury,
- znak pobrany z bufora do nadania w linię lub do wydruku,
- stan jednostki sterującej, która odrzuciła rozkaz "czytaj" lub "pisz",
- stan kluczy systemowych,
- aktualny stan lampek sygnalizacyjnych: itp.

Liczniki zbudowane są na 1, 2 lub 3 bajtach. Jeden z tych liczników, tzw. licznik czasu, służy do realizacji stanu aktywnego oczekiwania na przerwanie oraz do odliczania czasu ustawiania modemu na odbiór lub na nadawanie, przerwy w transmisji znaków (w obu kierunkach). Ponadto licznik ten taktuje zapalenie i gaszenie tych lampek, które migotaniem sygnalizują stan pracy terminala (w ten sposób sygnalizowana jest np. nieoperatywność urządzenia wejścia/wyjścia lub kanału transmisji).

Inne liczniki służą do zliczania liczby znaków umieszczanych w buforze, ilości znaków wydrukowanych w jednym wierszu itp.

Wskaźniki programowe są 1-bajtowe i na ogół przyjmują wartość 0 lub 1. Tylko niektóre wskaźniki przyjmują wartość 0, 1 lub 2.

Rolę wskaźnika programowego można porównać do roli zwrotnicy kolejowej. Stan (ustawienie) zwrotnicy decyduje, na który tor będzie skierowany pociąg, czyli określa trasę przebiegu pociągu. Podobnie stan (wartość) wskaźnika programowego określa, która procedura jest w danym momencie inicjowana, tzn. jaką "trasą" realizowane jest sterowanie programowe terminala. Stan wszystkich wskaźników programowych określa jednoznacznie stan, tryb i fazę pracy urządzenia w danej chwili.

Pod względem budowy ASTER składa się z modułów:

- sterowania, • komunikacyjnego, • wprowadzania i wyprowadzania danych.

Oprócz modułów program zawiera również następujące podprogramy:

- umieszczanie znaku w buforze, • konwersja znaku z kodu wewnętrznego na kod komunikacyjny,
- pobieranie znaku z bufora, • konwersja znaku z kodu komunikacyjnego na kod wewnętrzny.
- wydruk znaku,

Podprogramy konwersji kodu są wykorzystywane tylko w module komunikacyjnym, a pozostałe podprogramy są używane we wszystkich modułach.

Konieczność stosowania konwersji kodów wynika stąd, że minikomputer MERA 300 i jego urządzenia zewnętrzne (klawiatura, drukarka) pracują w kodzie ISO-7, zaś transmisja musi się odbywać w kodzie komunikacyjnym PTTC/EBCD.

Moduły programu składają się z segmentów. Procedury zawarte w tych segmentach realizują procesy programowo odpowiadające stanom pracy terminala takim, jak: gotowość, wprowadzanie danych do bufora, wydruk zawartości bufora, odbiór znaków z linii, nadawanie znaków w linię itd. Budowę i działanie programu sterującego ASTER wyjaśniają schematy logiczne modułów przedstawione na rysunkach 5 - 7.

Moduł sterowania zawiera cztery segmenty programowe. Segment 1 realizuje:

- ustawienie na stronie zerowej PAO adresu etykiety segmentu rozpoznania przerwania wejścia/wyjścia i maski przerwania;
- zainicjowanie wydruku nazwy programu, jeśli klucz systemowy nr 1 jest włączony;
- ustawienie początkowych wartości rejestrów i wskaźników programowych na stronach sterujących, czyli tzw. "zerowanie" terminala;
- ustawienie trybu pracy terminala w zależności od stanu klucza systemowego nr 0;
- ustawienie stanu gotowości terminala do odbioru sygnałów z linii;
- ustawienie sygnalizacji trybu i stanu pracy terminala i przekazanie sterowania do segmentu 2.

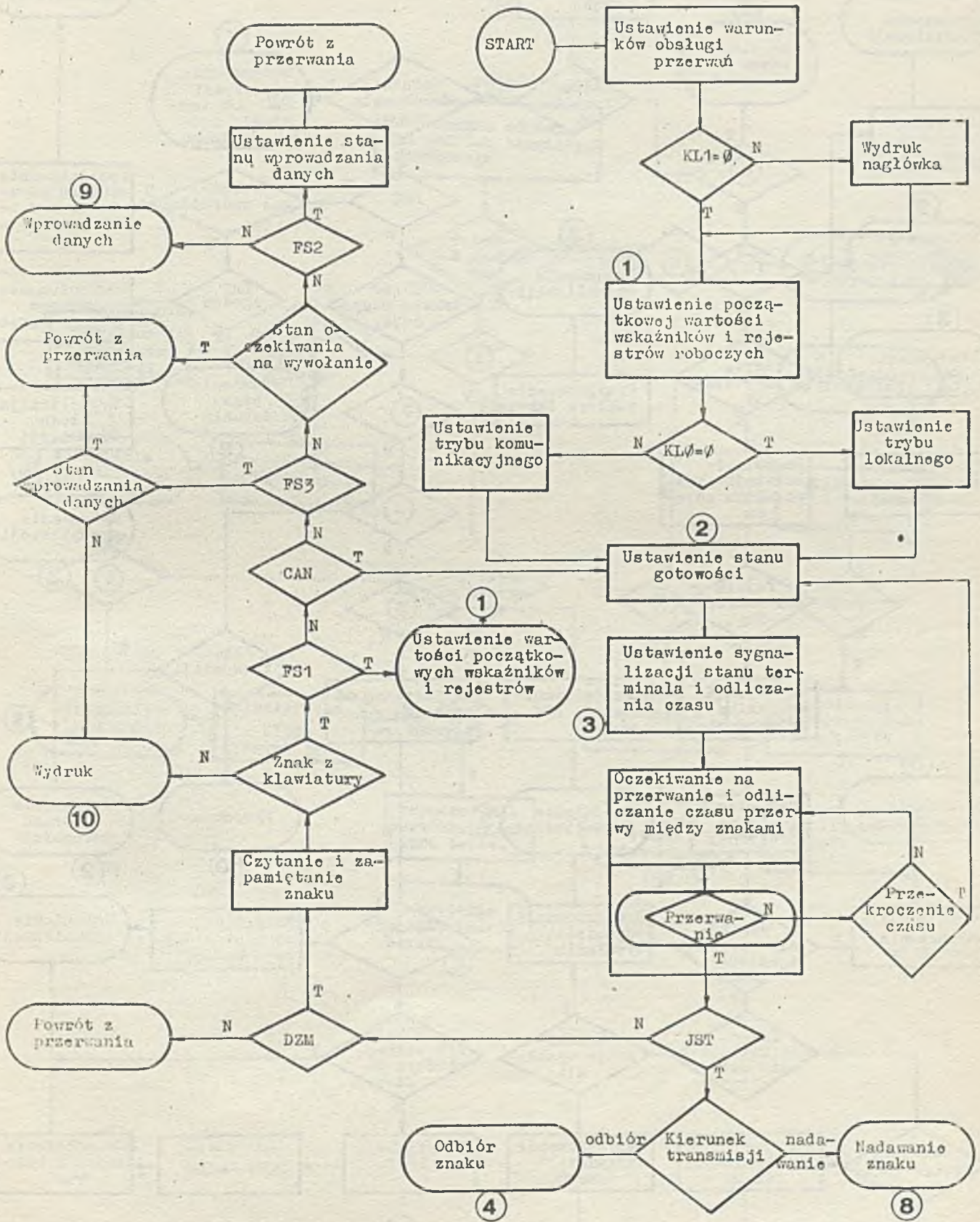
W segmencie 2 realizowane jest aktywne oczekiwanie na przerwanie wejścia/wyjścia i odliczanie czasu.

Z chwilą przyjęcia przerwania sterowania przejmuje segment 3, który rozpoznaje przyczynę przerwania i w zależności od wyniku tego rozpoznania oraz w zależności od fazy pracy inicjuje obsługę przerwania przez inny segment programowy.

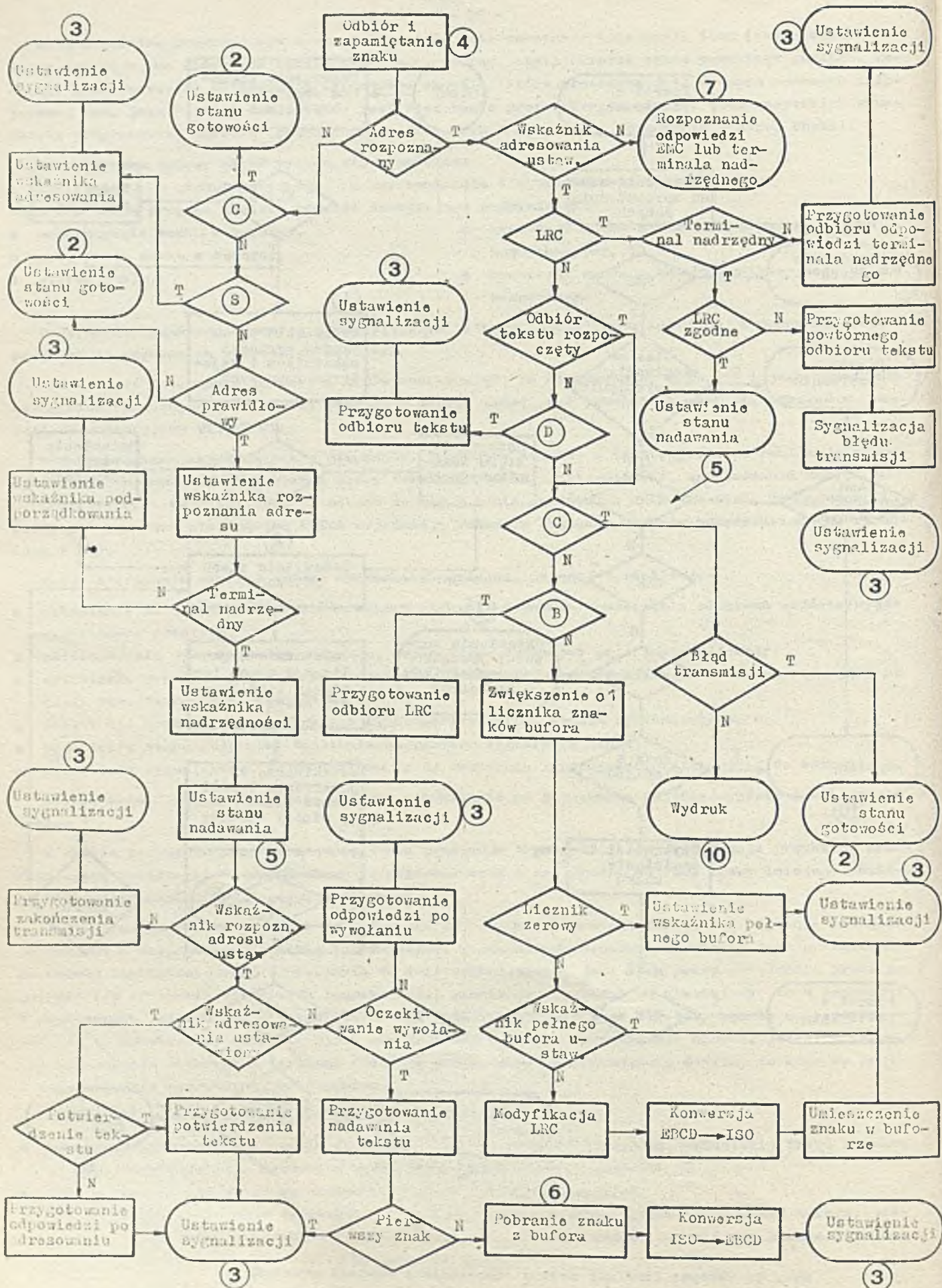
Jeśli przerwanie nie pochodzi od jednostki sterującej transmisją (JST), to działanie przejmuje segment 4 obsługujący drukarkę i klawiaturę. W przeciwnym wypadku sterowanie jest przekazane do modułu komunikacyjnego. Odpowiednio do tego sygnalizowany jest stan pracy urządzenia przez zapalenie lub zgaszenie właściwych lampek. Jeśli przerwanie pochodzi od klawiatury, to w segmencie 4 realizowany jest odbiór i rozpoznanie tego znaku, przy czym znak tekstowy będzie umieszczony w buforze tylko wówczas, jeśli terminal znajduje się w stanie wprowadzania danych. Jeżeli przerwanie spowodowała drukarka i terminal znajduje się w stanie wyprowadzania danych, to kolejny znak z bufora będzie wydrukowany na papierze.

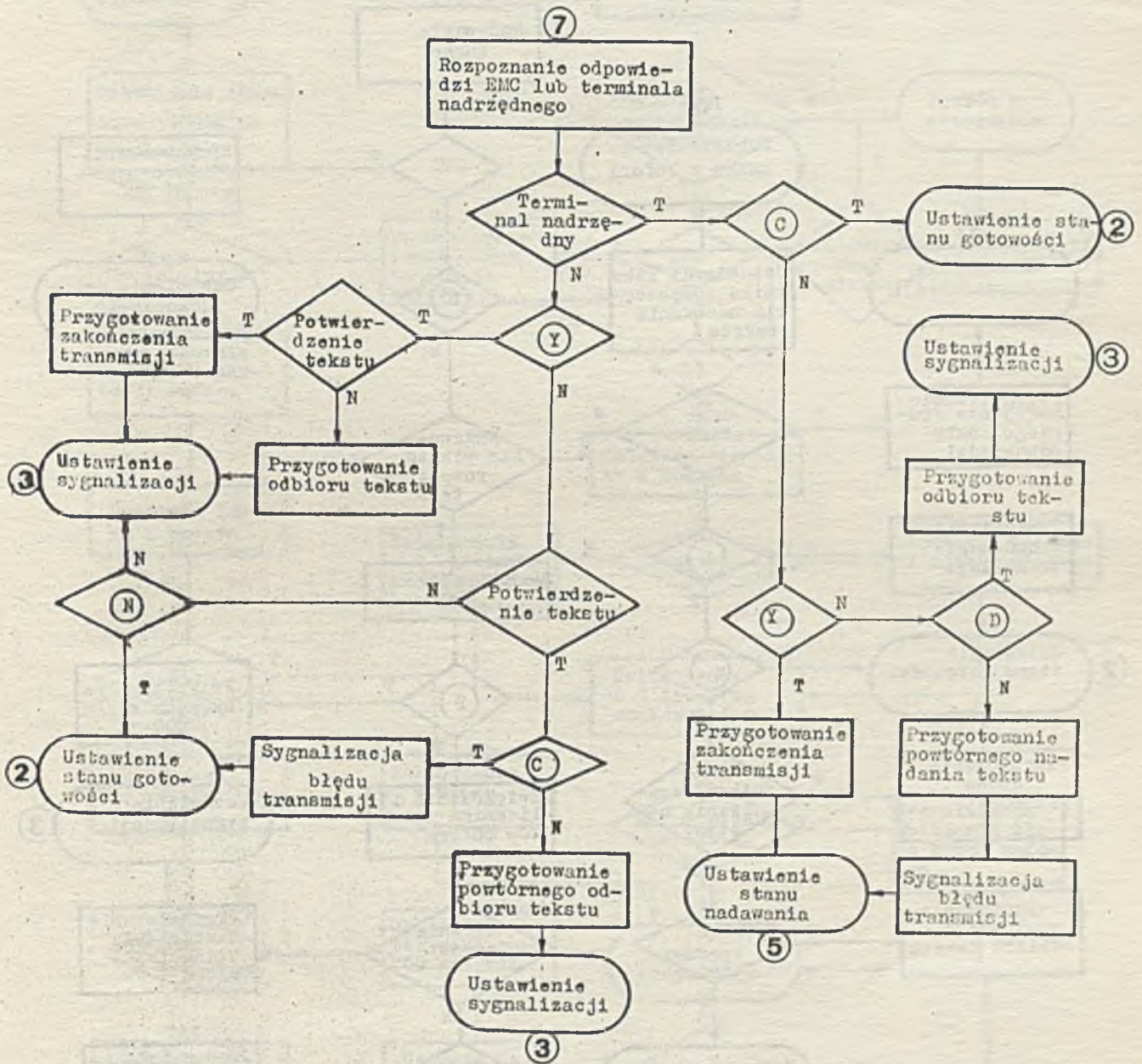
Znaki funkcyjne rozpoznane w segmencie 4 powodują:

- FS1 - przekazanie sterowania do segmentu 1, "zerowanie" terminala, ustawienie trybu i stanu terminala itd. zgodnie z wyżej opisanym działaniem segmentu 1;
- FS2 - ustawienie stanu wprowadzania danych i powrót do segmentu 2;
- FS3 - ustawienie stanu wyprowadzania danych i zainicjowanie wydruku zawartości bufora, jeśli terminal nie znajduje się w stanie wprowadzania danych, gdyż w tym stanie FS3 nie działa, tzn. powoduje tylko powrót z przerwania;

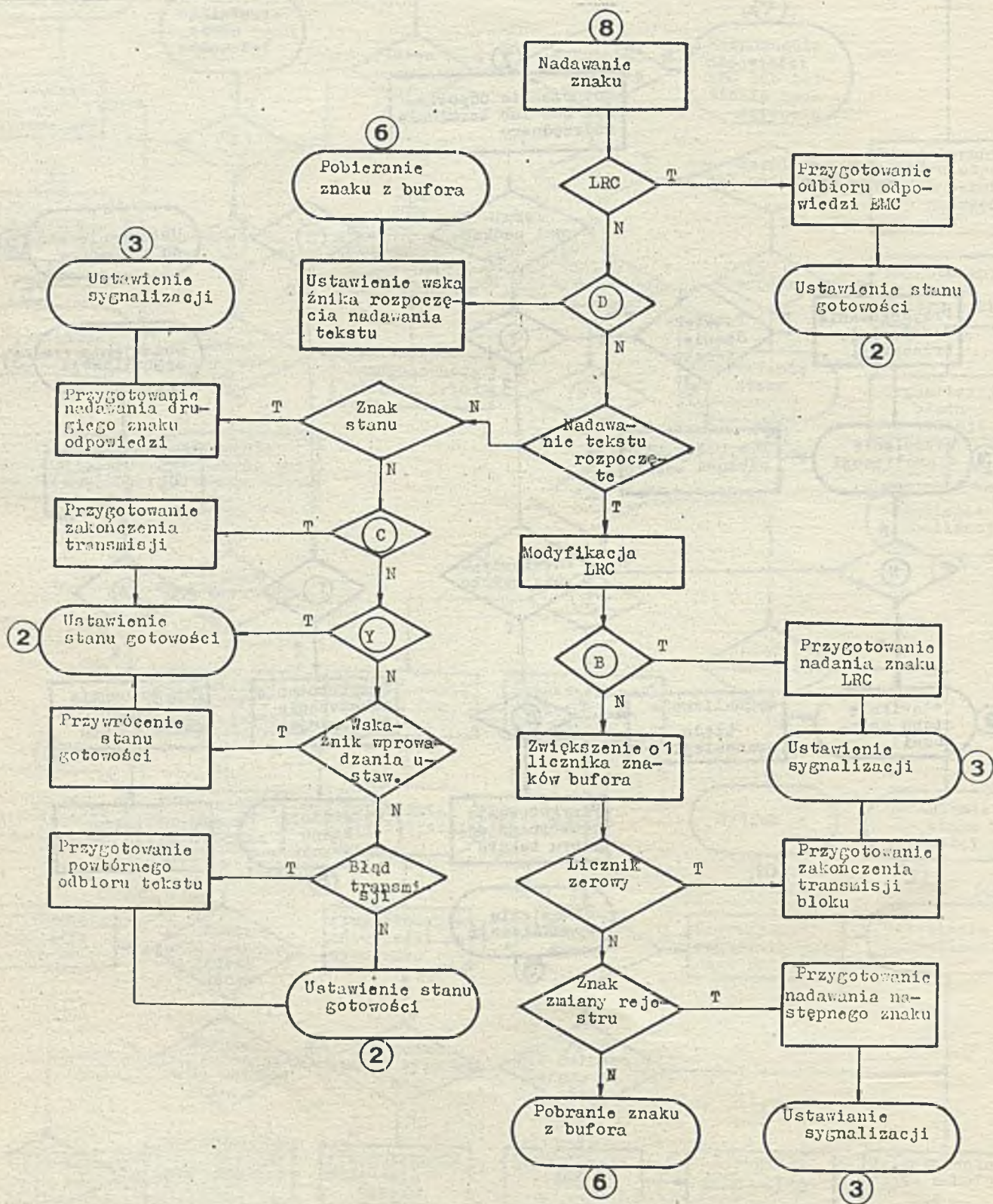


Rys. 5. Schemat logiczny modułu sterowania i obsługi przerw

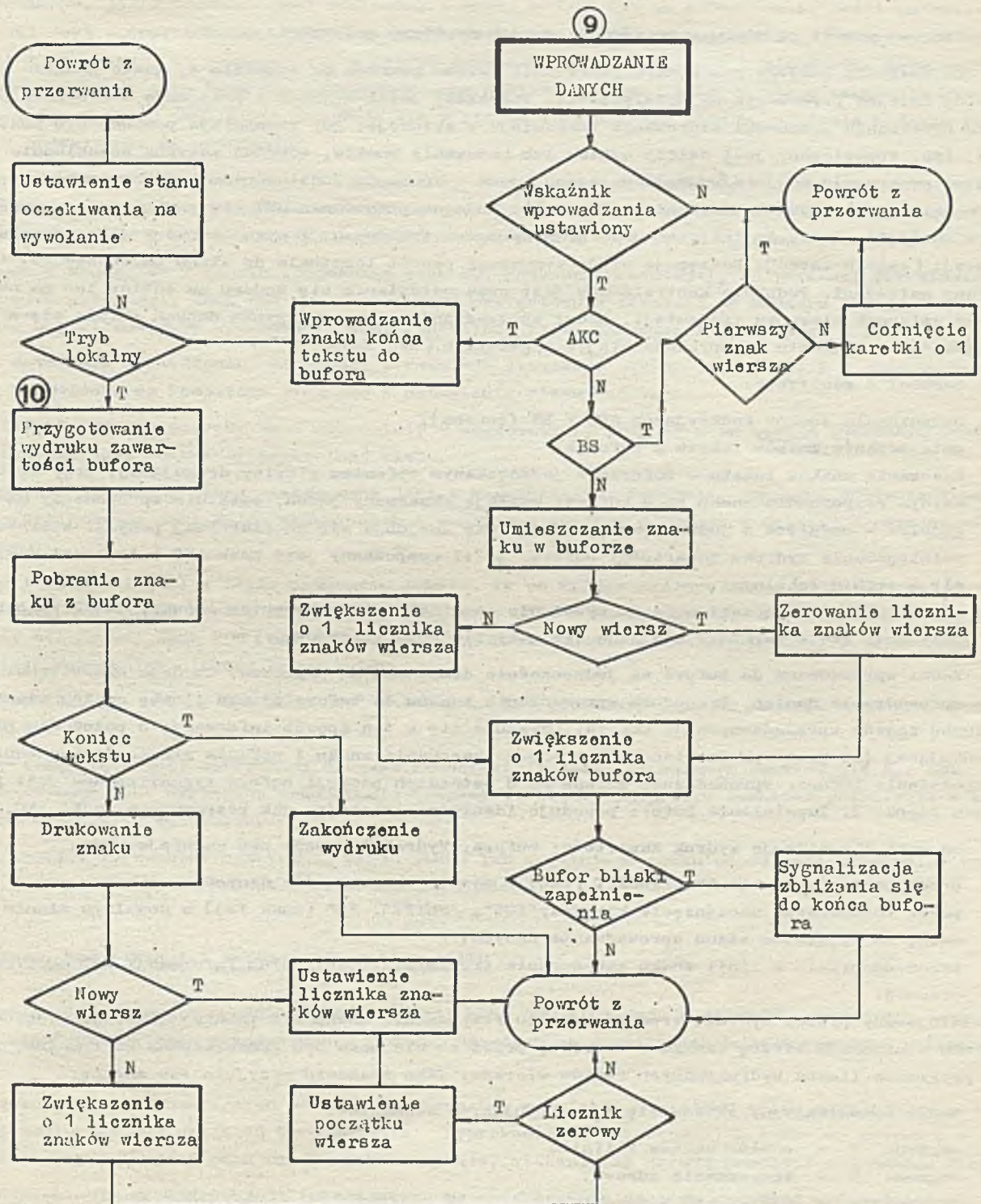




Rys. 6a. Schemat logiczny modułu komunikacji - część II



Rys. 6b. Schemat logiczny modułu komunikacji - część III



- CAN - powrót ze stanu wprowadzania danych do stanu gotowości.

Po obsłudze każdego przerwania sterowanie zawsze powraca do segmentu 2, gdzie program oczekuje na kolejne przerwanie wejścia/wyjścia. Jednakże, jeśli segment 2 przejmie sterowanie po obsłudze przerwania jednostki sterowania transmisją w sytuacji, gdy transmisja powinna być kontynuowana, tzn. spodziewany jest dalszy odbiór lub nadawanie znaków, wówczas aktywne oczekiwanie na kolejne przerwanie wejścia/wyjścia połączone jest z kontrolą (odliczaniem) odstępu czasu między poprzednim, a następnym przerwaniem JST. Jeśli kolejne przerwanie JST nie pojawia się w dopuszczalnym odstępie czasu, to fakt ten jest przez program traktowany i sygnalizowany jako przerwa w transmisji (awaria łącza). Następuje wtedy wymuszony powrót terminala do stanu początkowego, t.j. do stanu gotowości. Podobnie kontrolowany jest czas ustawiania się modemu na odbiór lub na nadawanie przy zmianach kierunku transmisji. Moduł wprowadzania - wyprowadzania danych składa się z dwóch segmentów: segmentu 5 (wprowadzania) i segmentu 6 (wydruku).

Segment 5 realizuje:

- rozpoznanie znaków funkcyjnych AKC i BS (cofacz);
- umieszczenie znaków tekstu w buforze;
- kasowanie znaków tekstu w buforze z jednoczesnym cofaniem głowicy drukującej, przy czym po każdym rozpoznaniu znaku BS w buforze zostaje skasowany jeden, ostatnio wprowadzony znak, zaś głowica - cofnięta o jedną pozycję, jeśli nie znajduje się na pierwszej pozycji wiersza;
- zainicjowanie wydruku zawartości bufora, jeśli rozpoznany jest znak AKC i terminal znajduje się w trybie lokalnym;
- ustawienie stanu oczekiwania na wywołanie terminala przez komputer główny, jeśli rozpoznany jest znak AKC i terminal znajduje się w trybie komunikacyjnym.

Znaki wprowadzane do bufora są jednocześnie drukowane na papierze, co daje operatorowi kontrolę wprowadzania danych. Procedura wprowadzania znaków do bufora zlicza liczbę znaków wiersza i liczbę znaków wprowadzonych do bufora. Uzyskuje się w ten sposób informację o położeniu głowicy drukującej (co jest wykorzystane w procedurze kasowania znaku i cofania głowicy) oraz kontrolę zapełnienia bufora. Wprowadzanie znaków na 8 ostatnich pozycji bufora sygnalizowane jest pulsowaniem lampki 2. Zapełnienie bufora powoduje identyczną reakcję, jak rozpoznanie znaku AKC.

Segment 6 realizuje wydruk zawartości bufora. Wydruk ten może być zainicjowany:

- przez naciśnięcie klawisza AKCEPT (AKC) w stanie wprowadzania danych;
- przez równoczesne naciśnięcie klawiszy "FUN", "SHIFT", "3" (znak FS3) w dowolnym stanie terminala, za wyjątkiem stanu wprowadzania danych;
- przez odebranie z linii znaku zakończenia transmisji, jeśli blok informacji był odebrany poprawnie.

Zainicjowany proces wydruku trwa do momentu rozpoznania znaku ETX (koniec tekstu). Procedura wydruku kontroluje liczbę znaków w wierszu, przez co nie może być przekroczona maksymalna, zadana w programie liczba wydrukowanych znaków wiersza. Jako standard przyjęto 159 znaków.

Moduł komunikacyjny składa się z następujących segmentów:

- segment 7 - odbiór znaków z linii;
- segment 8 - rozpoznanie adresu;
- segment 9 - ustawienie nadawania;
- segment 10 - nadawanie znaków w linii;
- segment 11 - rozpoznanie odpowiedzi terminala nadrzędnego;
- segment 12 - rozpoznanie odpowiedzi komputera głównego.

Segment 7 realizuje odbiór sygnałów linii oraz analizuje, czy przyjęty znak jest znakiem sterującym, znakiem tekstu, czy znakiem odpowiedzi. Dalsze działanie programu uzależnione jest od wyników tej analizy.

Jeśli terminal jest w fazie odbioru sekwencji adresowania lub wywołania, to działanie przejmie segment 8, w którym dokonywane jest rozpoznanie adresu.

Po rozpoznaniu swojego adresu indywidualnego w sekwencji adresowania lub wywołania, sterowanie przechodzi do segmentu 9. Tak samo przebiega działanie programu po rozpoznaniu adresu grupowego

lub ogólnego, jeśli terminal jest nadrzędny w danej grupie lub na danej linii. Jeśli natomiast terminal jest podporządkowany, to po rozpoznaniu adresu grupowego lub ogólnego, sterowanie przechodzi z segmentu 7 do segmentu 11, w którym dokonywane jest rozpoznanie odpowiedzi terminala nadrzędnego, przy czym terminal podporządkowany dostosowuje swój stan do stanu terminala nadrzędnego, o ile w danej chwili nie znajduje się w stanie wprowadzania danych, w stanie wydruku lub w stanie oczekiwania na wywołanie.

Po nadaniu tekstu z terminala do komputera, sterowanie z segmentu 7, po odbiorze znaku z linii, przechodzi do segmentu 12, gdzie dokonywane jest rozpoznanie odpowiedzi komputera głównego. Odpowiedź ta decyduje o dalszym działaniu terminala zgodnie z algorytmem opisanym wcześniej.

Jeśli w segmencie 7 będzie rozpoznany znak EOT (koniec transmisji), to terminal przestawia się w stan gotowości. Znak EOA (koniec adresu) przygotowuje terminal do odbioru tekstu. Znak EOB (koniec bloku) powoduje sprawdzenie sumy kontrolnej bloku informacji odebranego z linii.

Po dokonaniu sprawdzenia wierności transmisji działanie przechodzi do segmentu 9 w celu przygotowania odpowiedzi do komputera głównego i ustawienia stanu nadawania.

Odbiór tekstu dokonywany jest całkowicie w segmencie 7, przy czym po każdym odebraniu znaku modyfikowany jest znak sumy kontrolnej bloku.

Nadawanie znaków w linię realizowane jest w segmencie 10. Podobnie, jak przy odbiorze znaku z linii, analizowany jest każdy nadany znak i wynik tej analizy decyduje o dalszej drodze działania programu.

Jeśli terminal jest w fazie nadawania tekstu, to po każdym nadanym znaku modyfikowany jest licznik sumy kontrolnej bloku, a następnie pobierany jest z bufora kolejny znak tekstu, o ile ostatnio nie był nadany znak EOB (koniec bloku) lub licznik znaków bloku nie wskazuje, że została osiągnięta maksymalna długość komunikatu.

W wypadku, gdy nadany był znak EOB, z licznika LRC pobierany jest do nadania znak sumy kontrolnej bloku.

Gdy osiągnięta jest maksymalna długość komunikatu i ostatnim nadanym znakiem nie był EOB, to przygotowywane jest wymuszone zakończenie transmisji bloku, przy czym terminal pozostanie w stanie oczekiwania na wywołanie.

Po nadaniu znaku sumy kontrolnej bloku (LRC) lub ostatniego znaku odpowiedzi terminala, zmienia się kierunek transmisji z nadawania na odbiór.

Możliwości modyfikacji i zastosowań systemu ASTER

Konfiguracje sprzętowe minikomputerów serii MERA 300 stwarzają możliwość opracowania innych typów urządzeń zdalnego przetwarzania danych, opartych na rozwiązaniach zastosowanych w programie ASTER.

W zależności od konkretnych potrzeb użytkowników modyfikacje systemu ASTER mogą polegać na rozszerzeniu zestawu urządzeń wprowadzania - wyprowadzania informacji, a także na zmianach w zakresie innych funkcji i cech użytkowych terminala, włącznie ze zmianą procedur transmisji danych.

W odniesieniu do modyfikacji polegających na rozszerzeniu zestawu urządzeń wprowadzania i wyprowadzania informacji, w grę wchodzi właściwie wszystkie urządzenia zewnętrzne, które mogą pracować w rozszerzonej konfiguracji minikomputera serii MERA 300, jak to pokazano na rys. 1. Można by więc rozważać kilka wariantów asynchronicznych stacji końcowych, mających ten sam algorytm wymiany informacji z maszyną główną, co terminal ASTER-M300-A, lecz różniących się od niego urządzeniami wprowadzania i wyprowadzania danych. Najbardziej jednak celowo wydają się następujące modyfikacje:

- terminal z pamięcią dyskową,
- terminal z urządzeniami taśmy papierowej,
- terminal z monitorem ekranowym.

W każdym z tych wariantów może być także dołączona pamięć kasetowa zamiast czytnika i dziurkarki taśmy papierowej, albo jako dodatkowe źródło i odbiornik danych.

Inne modyfikacje systemu ASTER, które różniłyby się od stacji ASTER-M300-A pod względem funkcji i cech użytkowych, mogą dotyczyć Jego przystosowania:

- tylko do pracy w układzie dwupunktowym (uproszczony program sterujący);
- do pracy w systemie wsadowym;
- tylko do pracy dialogowej (z buforem o pojemności kilku znaków);
- do działania na łączach komutowanych;
- do współpracy z innymi terminalami itd.

Wydaje się, że spośród tego typu modyfikacji najbardziej celowy jest terminal konwersacyjny o właściwościach i procedurach sterowania transmisją takich, jakie ma terminal IBM 2741, ponieważ jest on obsługiwany w wielu istniejących systemach (np. CRJE, VM370).

Realizacja zmodyfikowanych wariantów systemu ASTER uzależniona jest od konkretnych potrzeb użytkowników.

Możliwości zastosowania systemu ASTER są różnorodne, zwłaszcza tam, gdzie już są stosowane minikomputery serii MERA 300, a równocześnie istnieje lub jest przewidywany komputer np. typu IBM S/370 lub R-32 z systemem zdalnego przetwarzania danych. Ograniczymy się do podania przykładów wskazujących możliwości i celowość zastosowania różnych wariantów terminali ASTER w obiektowych systemach komputerowej automatyzacji przemysłu maszynowego.

● Przykład 1

W dużych zakładach produkcyjnych, których wyroby finalne składają się z wielu skomplikowanych i różnorodnych w sposobie wytwarzania elementów i zespołów, działu ośrodek elektronicznego przetwarzania danych wyposażony między innymi w komputer IBM S/370 z rozbudowaną siecią terminali. Ośrodek obliczeniowy, na podstawie założonych na tym komputerze baz danych, prowadzi centralne przetwarzanie danych dla potrzeb zarządzania zakładami. Jednocześnie niektóre podsystemy centralnego systemu przetwarzania danych takie, jak ewidencja i kontrola realizacji zamówień na części zamienne, ewidencja, planowanie i kontrola realizacji zamówień na specjalne wykonania wyrobów, są prowadzone na minikomputerach MERA 305 wyposażonych w jedną lub dwie jednostki pamięci dyskowych, na których założone są lokalne zbiory danych dla tych podsystemów. Zarówno zakładanie zbiorów, jak i ich aktualizacja odbywa się na podstawie danych dostarczanych z centralnego systemu w postaci wydruków. Dane te są następnie ręcznie wprowadzane do minikomputerów, co jest procesem żmudnym i długotrwałym, a także jest źródłem wielu pomyłek.

Wyniki dobowego przetwarzania są dostarczane z tych podsystemów do centralnego systemu lub do właściwych użytkowników również w postaci wydruków. Dodać należy, że dział, w którym użytkowane są podsystemy lokalne, znajdują się w znacznej odległości od ośrodka obliczeniowego. Aby usprawnić proces dostarczania danych z centralnego systemu do podsystemów lokalnych i przesyłania wyników w odwrotną stronę, postanowiono zastosować system terminalowy ASTER w wersji dyskowej, wyposażając minikomputery MERA 305 w potrzebne jednostki sterujące transmisją asynchroniczną (adapтеры linii) i łącząc je za pośrednictwem modemów i linii wewnętrznej sieci telefonicznej z centralnym komputerem. Program sterujący terminala ASTER jest tu specjalnie przystosowany do potrzeb użytkownika. Przystosowanie to polega między innymi na możliwości transmitowania w obie strony dużych masywów danych, na automatycznym rozpakowywaniu rekordów przy pobieraniu ich z lokalnego zbioru danych i spakowywaniu rekordów przy przesyłaniu z bufora terminala na dysk, drukowaniu specjalnych komunikatów do operatora terminala, informujących go o przebiegu transmisji. Oszacowano, że zastosowanie terminalowego systemu ASTER skraca wielokrotnie czas wymiany informacji między systemem centralnym a podsystemem, a ponadto unika się pomyłek, jakie mają miejsce przy ręcznym wprowadzaniu danych do minikomputerów lub w trakcie ręcznego przenoszenia danych z wydruku na maszynowy nośnik informacji (karty dziurkowane), celem wprowadzenia wyników lokalnego przetwarzania danych do systemu centralnego.

● Przykład II

W fabryce wytwarzającej urządzenia elektroniczne, na wydziale obróbki mechanicznej pracują obrabiarki sterowane numerycznie za pomocą papierowych taśm dziurkowanych. W tym samym budynku, w innym wydziale tej samej fabryki montowane są na płytkach drukowanych układy elektroniczne, które następnie są sprawdzane i uruchamiane za pomocą testera sterowanego minikomputerem MERA 303. Do montażu układów elektronicznych używane są automaty i półautomaty również sterowane za pomocą dziurkowanych taśm papierowych. Programy testowania i kontroli tych układów i ich uruchamiania wprowadzane są do minikomputera także z taśm dziurkowanych, przy czym minikomputer jest wykorzystywany do celów testowania i uruchamiania pakietów elektronicznych tylko w ciągu jednej zmiany.

Taśmy dziurkowane do sterowania wszystkimi, wyżej wymienionymi procesami produkcyjnymi są wytwarzane centralnie w ośrodku elektronicznego przetwarzania danych, znacznie oddalonym od omawianych wydziałów produkcyjnych i dostarczane do nich okresowo za pomocą konwencjonalnych środków transportu.

Zuznaczyć trzeba, że zarówno na wydziale obróbki mechanicznej, jak i na wydziale montażu potrzebna jest częsta wymiana taśm ze względu na zmiany cyklu produkcyjnego oraz z powodu naturalnego zużywania się taśm. Oddalenie wydziałów produkcyjnych od ośrodka przetwarzania i trudności komunikacyjne powodują, że dystrybucja taśm jest niewystarczająca. Zdarzają się z tego powodu przestoje maszyn, co wpływa ujemnie na rytmiczność produkcji. Ośrodek EPD jest wyposażony w komputer IBM S/360 z możliwością zdalnego dostępu do niego przez multipleksor IBM 2703. Ponadto ośrodek EPD ma w niedługim czasie otrzymać komputer R-32, w którym także przewidywany jest system zdalnego przetwarzania danych.

Aby więc usprawnić dostarczanie taśm sterujących dla omawianych wydziałów produkcyjnych zdecydowano się zastosować system terminalowy ASTER, dostosowany do współpracy z czytnikiem i dziurkarką taśmy papierowej. System ten dodatkowo jest przystosowany do reprodukcji taśm w trybie pracy lokalnej. W ten sposób w ciągu drugiej zmiany, przy użyciu terminala ASTER, można przygotować taśmy sterujące do produkcji na dzień następny, zaś w sytuacjach awaryjnych możliwa jest komunikacja z komputerem centralnym w dowolnym momencie. Poprawia to zdecydowanie rytmiczność produkcji i wykorzystanie maszyn.

Literatura

- [1] IBM Systems Reference Library: IBM 2740/2741 Communication Terminal - Operators Guide. GA27-3001.
- [2] IBM Systems Reference Library: IBM 2740 Communication Terminal model 1 and 2 - Component description. GA 24-3403.
- [3] Momik 8b - zasady działania, cz. I: T-OF-605-D T-OF-607-D. Warszawa: MERA-INM 1974.
- [4] Dokumentacja techniczno-użytkowa terminala asynchronicznego ASTER-M300-A. Warszawa: MERA-INM 1977, opr. wewnętrzne
- [5] MARTIN J.: Wstęp do transmisji danych. Warszawa: WNT 1976.

mgr inż. Jerzy MYSIOR
Instytut Maszyn Matematycznych

SYSTEMY OPERACYJNE W PROBLEMOWO ZORIENTOWANYCH SYSTEMACH MINIKOMPUTEROWYCH

Przez problemowo zorientowany system minikomputerowy rozumiemy konfigurację sprzętową zawierającą małą maszynę cyfrową (minikomputer) z zestawem wyspecjalizowanych urządzeń zewnętrznych, wyposażoną w wyspecjalizowane oprogramowanie i przeznaczoną do rozwiązywania konkretnego problemu obliczeniowego wynikającego z potrzeb obiektu współdziałającego z systemem. Centralne miejsce w oprogramowaniu zajmuje system operacyjny.

Systemy operacyjne istnieją we wszystkich systemach komputerowych i zasady według jakich są one konstruowane w znacznej mierze nie zależą od klasy zastosowań (specjalizacji) systemu komputerowego. Powno jednak właściwości obiektu oraz problemu obliczeniowego rzutują na sprzętowe i programowe rozwiązania systemu komputerowego, w tym również na budowę systemu operacyjnego. Mówiąc o systemach operacyjnych w problemowo zorientowanych systemach minikomputerowych nie można więc pominąć zarówno tych cech, które charakteryzują wszystkie systemy operacyjne jak i tych, które wynikają ze specyficznych zastosowań systemów minikomputerowych.

1. Podział zasobów podstawowym zadaniem systemu operacyjnego

Systemem operacyjnym maszyny cyfrowej zwykle się nazywać taki zestaw środków (przeważnie automatycznych tzn. programów i niekiedy ludzkich tzn. operatorów), który umożliwia podział zasobów sprzętowych danej maszyny cyfrowej (czyli procesorów uniwersalnych, kanałów przesyłania, które mogą być traktowane jako wyspecjalizowane procesory, pamięci operacyjnych i pomocniczych oraz urządzeń zewnętrznych) między niezależnych użytkowników (czyli współdziałających z daną maszyną cyfrową ludzi, lub współdziałające maszyny - w tym również maszyny cyfrowe) według zadanych reguł, umożliwiając im współdziałanie w osiągnięciu zamierzonego celu.

Ten cel bywa różny w różnych zastosowaniach maszyn cyfrowych. Może być nim wykonanie planu produkcji w jakimś zakładzie przemysłowym (w obiektowych systemach komputerowego sterowania), uzyskanie szybkich i wiarygodnych odpowiedzi na żądanie zarezerwowania miejsca w samolocie (w systemach rezerwacji), uzyskanie wysokiej przepustowości (liczba zadań wykonanych w jednostce czasu) systemu liczącego (w systemach uniwersalnych) itp.

Podana wyżej definicja systemu operacyjnego eksponuje pojęcie podziału zasobów. Podział zasobów bywa niekiedy warunkiem koniecznym osiągnięcia zamierzonego celu, prawie zawsze natomiast pozwala osiągać ten cel taniej. Łatwo zauważymy, że np. w systemie rezerwacji miejsc użytkownicy muszą korzystać ze wspólnej bazy danych, która aktualizuje się w następstwie realizacji ich żądań. O dostęp do tej bazy, zainstalowanej np. w pamięci dyskowej, użytkownicy będą się ubiegać niezależnie. Jeśli w trakcie realizacji żądania użytkownika A pojawi się żądanie użytkownika B, to jego realizacja musi być wstrzymana do chwili, kiedy będzie przygotowana odpowiedź dla A oraz zawartość bazy zostanie zaktualizowana np. będzie odnotowane, że miejsce w samolocie jest zajęte. Użytkownik B, który uzyska teraz dostęp do bazy będzie, podobnie jak A, dysponował całą bazą, jednak w innym, niż A, okresie czasu. Można więc powiedzieć, że jedna rzeczywista pamięć dyskowa jest podzielona na wiele wirtualnych pamięci dyskowych, o tej samej co rzeczywista pojemności, istniejących jednak w nie pokrywających się okresach czasu.

Rozważmy inną sytuację. Jeśli w maszynie cyfrowej wykonywany jest program użytkownika A i program ten zażądał wprowadzenia porcji danych np. z czytnika kart, to w trakcie przesyłania danych do pamięci przez kanał przesyłania procesor maszyny pozostaje bezczynny. Przyjmijmy, że w pamięci jest wystarczająco dużo miejsca, aby pomieścić program użytkownika B. W tym czasie, kiedy na rzecz programu użytkownika A działa kanał przesyłania, procesor mógłby wykonywać program użytkownika B i np. wrócić do wykonywania programu A, gdy przesyłanie będzie zakończone. W rezultacie

takich przełączeń, bez żadnych inwestycji sprzętowych, program użytkownika A wykonałby się w niewiele tylko dłuższym czasie niż gdyby sam rezydował w pamięci (dochodzi czas przełączania procesora). Natomiast praktycznie za darmo, w tym samym czasie byłby wykonywany (być może do końca) program B, który w przeciwnym razie musiałby oczekiwać na rozpoczęcie wykonywania do chwili zakończenia programu A. W ten sposób każdy z programów może być wykonany w średnio krótszym czasie licząc od chwili zgłoszenia do wykonania, niż gdyby czekał w kolejce na całkowite zwolnienie maszyny. Dzieląc jeden procesor fizyczny na pewną liczbę wirtualnych procesorów, istniejących w rozłącznych okrasach czasu, system operacyjny pozwala uzyskać zarówno lepsze wykorzystanie sprzętu (minimalizację okresów bezczynności procesora), jak i wzrost przepustowości systemu (zwiększenie liczby programów wykonanych w ustalonym okresie czasu - np. w ciągu doby). Uzyskujemy więc niższy koszt usług obliczeniowych bez pogarszania ich jakości.

Przeprowadzone wyżej bardzo uproszczone rozumowanie pokazuje jakiego rodzaju problemy rozwiązuje system operacyjny.

2. System operacyjny a oprogramowanie użytkowe

Niekiedy można się spotkać ze stwierdzeniem, że w systemach stałoprogramowych, których użytkownikami są nie ludzie lecz np. obrabiarki, nie istnieje system operacyjny. Oczywiście nie można nikomu zabronić wygłaszania takich poglądów. Jeśli ktoś nie wyróżnia w oprogramowaniu systemu operacyjnego to widocznie nie odczuwa tej potrzeby. Wszystko zależy bowiem od punktu widzenia. Wyodrębnianie z większej całości i nazywanie jakiegoś pojęcia ma sens jedynie wtedy, gdy pomaga tę całość lepiej zrozumieć. Jeśli np. ładujemy do pamięci maszyny cyfrowej jej pełne oprogramowanie z taśmy papierowej, to może nam wystarczyć świadomość, że taśma ma długość 100 mb a czas jej wczytywania nie przekracza 10 sekund i wcale nas nie interesuje, czy na tej taśmie zakodowano system operacyjny, czy też nie. Jeśli analizujemy zachowanie się systemu komputerowego na podstawie obserwacji wprowadzanych do systemu danych i uzyskiwanych wyników, nie musimy nazywać składowych oprogramowania ani nawet wyróżniać w systemie części sprzętowej i programowej. Jeśli jednak chcemy zrozumieć, jak działa oprogramowanie lub jeśli chcemy oprogramowanie zaprojektować, musimy wykonać jego dekompozycję na takie elementy, które wygodniej jest analizować oraz projektować oddzielnie.

Szczególnie ważna jest możliwość wydzielenia z oprogramowania tych części, które można projektować bez uwzględniania specyficznych cech maszyny cyfrowej, a których działanie da się sprowadzić do przekształcania zbioru danych w zbiór wyników. Części te składają się na to, co zwykle nazywamy oprogramowaniem użytkowym. Do programów z tej grupy zaliczyć również można kompilatory języków programowania, gdyż funkcjonalnie są one również programami przetwarzającymi dane w wyniki. Programy w systemach czasowo uwarunkowanych również mogą być rozpatrywane w tych kategoriach, jeśli zależności czasowe czyli opóźnienia lub chwile rozpoczynania jakichś akcji, potraktować jako elementy zbiorów danych dostarczanych tym programom. Podział na części przez zrównoleżenie wykonawstwa pozwala szybciej uzyskać programy o wyższej jakości, gdyż łatwiej jest dobrze zaprojektować prostszy, wyraźnie określony program niż program skomplikowany.

Systemy operacyjne w systemach zmiennoprogramowych, których użytkownikami są ludzie, manifestują swoją obecność w gotowym systemie, wymagając znajomości zasad opisywania programów przekazywanych do wykonywania (języki typu Job Control Language) itp. Systemy operacyjne w systemach stałoprogramowych są, po zrealizowaniu systemu, niewidoczne. Niemniej wydzielenie systemu operacyjnego również w tych systemach wynika z tych samych przyczyn, dla których wydziela się oprogramowanie użytkowe. System operacyjny jest tym fragmentem oprogramowania, który może być konstruowany według zasad wspólnych dla wszystkich systemów operacyjnych, w znacznym stopniu niezależnie od typu maszyny cyfrowej i charakterystyki zbioru programów użytkowych. Oczywiście typ maszyny cyfrowej i rodzaj zastosowań mają wyraźny wpływ na konstrukcję systemu operacyjnego, ale racjonalniej jest iść od podobieństw do różnic niż odwrotnie.

3. System operacyjny symulatorem maszyny wirtualnej obliczenia

Do podziału oprogramowania na oprogramowanie użytkowe i system operacyjny można dojść również przeprowadzając inne rozumowanie.

Jeśli chcemy zastosować maszynę cyfrową do rozwiązywania jakiegoś problemu wynikającego z potrzeb jakiegoś zbioru użytkowników (rozumianych tak jak wyżej) powinniśmy sformułować ten problem i określić sposób jego rozwiązywania. Sposób rozwiązywania problemu można wyrazić w formie obliczenia, które rozpatrywać można jako uporządkowany zbiór operacji wykonywanych na skończonym zbiorze danych, przekształcających ten zbiór w inny skończony zbiór wyników.

Realizacja obliczenia w maszynie cyfrowej sprowadza się do umieszczenia programu tego obliczenia w pamięci maszyny i przydzielenia temu programowi niezbędnej liczby procesorów. Program obliczenia jest sformalizowanym zapisem tego obliczenia w języku programowania. Operacje obliczenia wyrażane są za pośrednictwem instrukcji (zdań) tego języka.

Układając program obliczenia z reguły nie interesujemy się technicznymi detalami konkretnej maszyny cyfrowej. Nie troszczymy się o to w jaki sposób strumienie danych i wyników programu będą przenoszone przez konkretne urządzenia wejścia-wyjścia, w jaki sposób program będzie rozmieszczony w pamięci i wreszcie skąd w maszynie znajdzie się niezbędny zestaw procesorów.

Zakładamy, że taka maszyna istnieje i że jest nią abstrakcyjna, wirtualna maszyna obliczenia. Maszyna ta dysponując zbiorem wirtualnych procesorów i wirtualną pamięcią kontaktuje się z otoczeniem przez wirtualne urządzenia wejścia-wyjścia będące producentami danych i konsumentami wyników programu. Maszyny wirtualne niezmiernie rzadko są realizowane sprzętowo, przeważnie są one symulowane na maszynach rzeczywistych. Poszukując odpowiadającej nam maszyny wirtualnej w istocie poszukujemy zrealizowanego jej symulatora.

Warto sobie uświadomić, że obszar poszukiwań ogranicza podjęta uprzednio decyzja: wybór języka programowania. Wybór języka niskiego poziomu (assemblera) praktycznie oznacza również wybór konkretnej implementacji symulatora maszyny wirtualnej. Jeśli wybraliśmy język wysokiego poziomu, to możemy odnaleźć wiele implementacji potrzebnej nam maszyny wirtualnej. W wielu wypadkach bowiem naszą maszynę wirtualną można odwzorować w innej maszynie wirtualnej, dokonując formalnego przekształcenia programu naszego obliczenia na równoważny mu funkcjonalnie program innej maszyny. Można to wykonać za pomocą kompilatora przyjętego języka programowania, który albo istnieje albo musi być napisany.

Niekiedy jednak wśród dostępnych symulatorów nie znajdziemy takiego, który odpowiadałby naszym potrzebom. Musimy wtedy ten symulator skonstruować.

Zauważmy, że skoro maszyna wirtualna ma mieć niezbędną liczbę wirtualnych procesorów, wirtualne pamięci itp. to utworzyć je możemy z fizycznych procesorów i pamięci dokonując ich podziału wg jakichś reguł. Wyżej pokazaliśmy, że podziałem sprzętowych zasobów maszyny cyfrowej zajmie się system operacyjny tej maszyny. Symulator maszyny wirtualnej na maszynie rzeczywistej będzie więc systemem operacyjnym tej rzeczywistej maszyny cyfrowej działającym na rzecz programu obliczenia czyli oprogramowania użytkowego.

Przystępując do konstruowania symulatora maszyny wirtualnej musimy zacząć od zaprojektowania tej maszyny. Projektując maszynę wirtualną określamy jej język. Język maszyny wirtualnej jest prawie zawsze językiem niskiego poziomu z tej przyczyny, że z reguły obejmuje instrukcje rzeczywistej maszyny (arytmetyczne, logiczne), których po prostu nie trzeba wtedy symulować. Przekształcanie programu obliczenia, jeśli napisano go w języku wysokiego poziomu, jest zadaniem kompilatora, który, być może, będzie miał z tym sporo roboty. Nie ma to jednak wpływu na jakość realizacji obliczenia, gdyż kompilacja przebiega w innym czasie i często w innym miejscu niż realizowane jest obliczenie. Gdyby natomiast system operacyjny musiał interpretacyjnie wykonywać wszystkie instrukcje programu obliczenia, miałoby to bezpośredni wpływ na czas tej realizacji.

Wybór rzeczywistego systemu komputerowego, a ściślej jego konfiguracji, ograniczany jest przez konieczność uwzględnienia urządzeń wejścia/wyjścia dostosowanych do fizycznych właściwości współdziałającego obiektu (inne urządzenia - gdy z systemu korzystają ludzie, inne - gdy maszyny) oraz

przez możliwość realizacji maszyny wirtualnej o wymaganych charakterystykach czasowych (czas realizacji) i objętościowych (pojemności pamięci dla programów i danych).

Warto tu podkreślić, że o ile przy określaniu urządzeń zewnętrznych swoboda wyboru jest ograniczona przez wymagania obiektu, o tyle przy pozostałych składowych systemu komputerowego (procesory uniwersalne, kanały przesyłania oraz pamięci operacyjne i pomocnicze) ograniczenia nie są tak ostre - można znaleźć wiele funkcjonalnie podobnych zestawów sprzętowych.

W zależności od wymagań pamięciowych i czasowych obliczenia - maszynę wirtualną z nią związaną będziemy realizować albo na dużej maszynie cyfrowej albo na minikomputerze. Jest to wskazówka jedynie orientacyjna. Właściwie racjonalnym kryterium doboru sprzętu jest, aby wybrana maszyna cyfrowa nie była zbyt duża w stosunku do potrzeb, albo inaczej, żeby to była najtańsza z dostępnych konfiguracji sprzętowa, na której da się zrealizować maszynę wirtualną niezbędną do realizacji obliczenia.

4. Wpływ specjalizacji systemu komputerowego na system operacyjny

W zależności od rozległości problemu, na rzecz rozwiązywania którego ma działać system komputerowy (co ma wpływ na rozmiar obliczenia i w konsekwencji na "rozmiary" systemu sprzętowego), możemy mówić o systemach o szerokiej i o wąskiej specjalizacji.

Systemami o najszerszej specjalizacji są tzw. systemy uniwersalne, przeznaczone do rozwiązywania dowolnych zadań formułowanych przez użytkowników, dopuszczające różne formy współdziałania z tymi użytkownikami (wsadowo i wielodostępnie). Przykładem takiego systemu może być działający na maszynach IBM 370 system VM, który pozwala jednocześnie działającym użytkownikom korzystać niezależnie albo z konwersacyjnego systemu CMS, albo z systemów wsadowych takich jak OS czy DOS.

Systemów o wąskiej specjalizacji należy szukać wśród systemów sterujących pojedynczymi maszynami lub liniami produkcyjnymi, wśród maszyn fakturujących, inteligentnych urządzeń końcowych itp. Przykładem takiego systemu może być SKSC-1 zrealizowany na maszynach MERA 300. Systemy takie, z reguły realizowane na minikomputerach, możemy nazwać problemowo zorientowanymi systemami minikomputerowymi.

Dla systemów o wąskiej specjalizacji, problemowo zorientowanych, charakterystyczne jest, że są to z reguły systemy stałoprogramowe, tzn. z niezmiennymi programami obliczenia. Systemy uniwersalne z kolei, to przeważnie systemy zmiennoprogramowe, tj. takie, w których fragmenty programu obliczenia zmieniają się w czasie.

Charakterystyki maszyn wirtualnych w systemach stało- i zmiennoprogramowych oraz warunki ich sprzętowej realizacji zależą od pewnych cech programów obliczeń.

Program obliczenia musi wyrażać intencję współbieżnego wykonywania operacji tego obliczenia. Współbieżność, czyli nieustalona kolejność (w tym możliwa jednoczesna) wykonywania operacji, przede wszystkim wynika z faktu, że obiekt, na rzecz którego obliczenie jest realizowane, często jest producentem i konsumentem wielu niezależnych strumieni danych i wyników (użytkownicy niezależnie prowadzący konwersację z maszyną, przyrządy pomiarowe wmontowane w linię produkcyjną itp.) a więc odpowiednio stanowiący składowe obliczenia konsumentów i producentów tych strumieni muszą mieć możliwość niezależnego działania. Z drugiej strony, współbieżność pozwala łagodzić wymagania w stosunku do sprzętu, (głównie do szybkości działania procesora), na którym realizowana jest maszyna wirtualna obliczenia.

Nie wchodząc w szczegóły, które obszernie omawia literatura z zakresu teorii programowania równoległego, wspomnę tu jedynie, że obliczenie współbieżne, dla ułatwienia analizy jego właściwości oraz łatwiejszego zaprogramowania, przekształcane jest na zbiór współdziałających obliczeń sekwencyjnych (procesów), przy czym to współdziałanie wyraża się przez synchronizację procesów czyli ograniczanie współbieżności przy korzystaniu ze wspólnych zmiennych. Ta synchronizacja sprowadza się do zakazu jednoczesnego wykonywania wskazanych operacji należących do różnych procesów (reguła wzajemnego wykluczania) lub do narzucania kolejności (reguła komunikacji).

Operacje synchronizujące, rozgałęziające procesy (po jednej operacji - kilka), zbierające pro-

cesy (po kilku operacjach - jedna) znajdują swoje odzwierciedlenie w konstrukcjach językowych w tzw. językach programowania równoległego i w konsekwencji w instrukcjach maszyny wirtualnej obliczenia. Inną grupę instrukcji maszyny wirtualnej stanowią instrukcje, najogólniej mówiąc, wejścia-wyjścia, czyli instrukcje dotyczące wirtualnych urządzeń zewnętrznych. Ponadto mogą występować instrukcje pozwalające uwzględniać w obliczeniu reakcję na zdarzenia zewnętrzne np. działanie w zaplanowanych chwilach czasowych, wykonywanie zaplanowanych akcji, gdy zaistnieje jakieś zdarzenie w obiekcie. Ten zbiór instrukcji rozszerzany bywa, o czym wspomniano wyżej, o arytmetyczne, logiczne itp. instrukcje rzeczywistej maszyny cyfrowej.

Jak już powiedziano wcześniej, symulator maszyny wirtualnej na maszynie rzeczywistej (system operacyjny) jest zbiorem programów maszyny rzeczywistej, przekształcających tę maszynę na maszynę wirtualną obliczenia. Podstawowym wykorzystywanym mechanizmem jest system przerwań. Źródłem przerw są urządzenia zewnętrzne, zegar oraz programy procesów obliczenia, głównie przez ekstrakody, stanowiące formę wyrażenia instrukcji różnych od realizowanych bezpośrednio przez sprzęt.

Podstawowym zadaniem systemu operacyjnego jest realizacja abstrakcji przyjętej przy dekompozycji obliczenia współbieżnego na zbiór procesów, tj. dostarczenia każdemu procesowi wirtualnego procesora. Te wirtualne procesory powstają w wyniku przełączania fizycznego procesora lub rzadziej większej liczby procesorów, z jednego ciągu instrukcji na inny, zgodnie z przyjętą strategią. Inne realizowane abstrakcje dotyczą pamięci (wirtualizacja pamięci czyli powiększanie obszaru pamięci operacyjnej przez wykorzystanie pamięci pomocniczej) oraz urządzeń zewnętrznych (tworzenie z rzeczywistych urządzeń zewnętrznych urządzeń wygodniejszych, transmitujących dane według formatów wymaganych przez procesy obliczenia).

Na zadania te, poza oczywistą powinnością realizacji ekstrakodów, nakładane są dodatkowe ograniczenia. Niektóre z nich mają związek ze specjalizacją systemu, inne od tej specjalizacji nie zależą.

W systemach o wąskiej specjalizacji ograniczenia podstawowe dotyczą z reguły czasu reakcji na zdarzenia zachodzące w obiekcie (opóźnienie przekraczające zadany okres czasu lub zgubienie sygnału mogą mieć nieodwracalne konsekwencje), oraz odporności systemu na zakłócenia wprowadzane przez obiekt i sam system komputerowy (system powinien niwelować skutki mniejszych zakłóceń a przy poważnych - nie podejmować decyzji szkodliwych dla obiektu). Ograniczenia te mają wpływ na wybór sprzętu, gdyż muszą istnieć odpowiednie mechanizmy - jak np. wielopoziomowy system przerw, układowa kontrola i sygnalizacja uszkodzeń sprzętu oraz na konstrukcje programowe (rozbudowany aparat kontroli).

W systemach o szerokiej specjalizacji, a zwłaszcza zmiennoprogramowych, przeznaczonych do wykonywania dowolnych zadań, dąży się w pierwszej kolejności do najlepszego wykorzystania sprzętu. W konsekwencji - niekiedy nawet kosztem wydłużenia czasu reakcji dla pojedynczego użytkownika (czas obrotu zadania) dąży się do wzrostu przepustowości, tj. powiększenia liczby jednocześnie użytkowników systemu, zwiększenia liczby wykonywanych programów w jednostce czasu. Większą przepustowość uzyskuje się przez odbieranie procesora sprzętowego tym procesom, które oczekują na zakończenie wprowadzenia czy wyprowadzenia danych i przydzielanie go innym, które w danej chwili nie korzystają z wejść/wyjść oraz przez grupowanie w pamięciach pomocniczych danych i wyników należących do różnych procesów tak, aby stosowne urządzenia zewnętrzne były stale obciążone. Oczywiście jest, że im większa jest liczba procesów w systemie - a więc zadań użytkowników - tym lepsze można osiągnąć wykorzystanie sprzętu.

W systemach zmiennoprogramowych źródłem poważnych ograniczeń jest fakt, że zadania dostarczone przez użytkowników mogą działać niezgodnie z intencjami ich autorów. System musi umieć bronić się przed wynikającym z błędów w jednym programie - niebezpieczeństwem zakłócenia działań innych programów, zniszczenia przechowywanych danych i zakłóceń pracy samego systemu operacyjnego. Dąży się więc do stworzenia takich sytuacji, aby błędy miały jedynie lokalne konsekwencje zaś próba przekroczenia przyznanych użytkownikowi uprawnień była wykrywana zanim spowoduje szkodliwe skutki w otoczeniu błędnego procesu.

System musi również dawać sobie radę z zamierzonym lub przypadkowym przywłaszczeniem sobie przez jeden proces procesora lub urządzenia zewnętrznego na nieograniczony czas (np. pętla bez

końca w programie lub szczególnie trudna do wykrycia blokada - deadlock, tj. błąd synchronizacji oddzielnie poprawnych procesów wyrażający się w oczekaniu wszystkich procesów na spełnienie warunku, który nie może być spełniony).

Systemy stałoprogramowe z założenia nie zawierają programów błędnych. Błędy, które mimo wszystko pozostają (testowanie nie może dać gwarancji, że wszystkie błędy wykryto), traktowane są na równi z błędami w implementacji systemu operacyjnego i można je przyrównać do wad konstrukcyjnych sprzętu. Wykrycie i usunięcie takich błędów ułatwić może czytelna dokumentacja oprogramowania - najlepsza, gdy całe oprogramowanie wyrażone jest w języku programowania wysokiego poziomu. Nie ma to jednak bezpośredniego wpływu na konstrukcję maszyny wirtualnej obliczenia ani na symulator tej maszyny tzn. na system operacyjny.

Charakterystyczny dla systemów zmiennoprogramowych brak zaufania w stosunku do użytkownika-programisty, w każdym dobrze skonstruowanym systemie rozciągać się powinien na operatorów systemu. System musi umieć wykrywać oraz ignorować każdą błędną akcję podjętą przez operatorów przy użyciu legalnych środków, tzn. bez użycia młotka czy kolby do lutownia.

Specjalizacja systemu, której konsekwencje, jak pokazaliśmy, rozciągają się zarówno na sprzętowo jak i programowo elementy systemu, prowadzić powinna do sytuacji, w których dany system jest dobrze dostosowany do potrzeb określonego użytkownika, tj. obiektu współdziałającego.

Zasady tej nie da się odnieść do systemów uniwersalnych. Jeśli chce się być jednakowo dobrym dla wszystkich, to w praktyce często jest się jednakowo złym dla każdego. Rozwiązania uniwersalne prowadzą częstokroć, jak to ma miejsce np. w systemach zrealizowanych na maszynach IBM 360/370 do sytuacji, w której poprawę jakości usług np. skrócenie czasu reakcji systemu - czasu obrotu zadania, uzyskać można jedynie przez niewspółmiernie z efektami duże nakłady finansowe na rozbudowę zestawu sprzętowego np. powiększanie pojemności pamięci operacyjnych oraz pomocniczych. W ten sposób system o najszerszej specjalizacji staje się systemem o specjalizacji najwęższej, i de facto realizując politykę producenta sprzętu, który chce sprzedawać tego sprzętu jak najwięcej.

Oczywiście sytuacja taka prowadzi do postawienia na głowie wszelkich prób racjonalnego podejścia do projektowania oprogramowania. Widać bowiem wyraźnie, że im mniej optymalnie zaimplementuje się maszynę wirtualną, im więcej miejsca i czasu pochłaniają wchodzące w skład symulatora maszyny wirtualnej (systemu operacyjnego) programy, tym lepiej (dla producenta sprzętu!).

5. Wytwarzanie systemów operacyjnych dla problemowo zorientowanych systemów minikomputerowych

Jeśli poważnie potraktujemy dążenie do minimalizacji kosztów realizacji systemu jako całości, musimy preferować takie rozwiązania, w których suma kosztów sprzętu i oprogramowania jest minimalna.

Warto w tym miejscu zwrócić uwagę na fakt, że koszt oprogramowania obejmuje zarówno jego zaprojektowanie, implementację na konkretnym systemie sprzętowym jak i późniejszą eksploatację, w trakcie której nie rzadko zachodzi potrzeba wprowadzania zmian do oprogramowania. Koszt ten ponadto powinien być rozpatrywany w kontekście przewidywanych efektów zastosowania pełnego systemu, a dokładniej - rozpatrzenia strat wynikających z opóźnienia przekazania systemu do użytkowej eksploatacji. Może się okazać, że droższy system, uzyskany wcześniej, jest w efekcie bardziej opłacalny niż tańszy, uzyskany później. Oczywiście koszty i czas uzyskania oprogramowania systemu i naczej będą się kształtować w specjalistycznym przedsiębiorstwie dysponującym dobrze przygotowaną kadrą profesjonalnych projektantów i programistów oraz narzędziami sprzętowo-programowymi, a inaczej, gdy potencjalny użytkownik systemu sam podejmie próbę jego zbudowania.

Nając na uwagę obniżkę kosztów oprogramowania rozważmy dwie drogi wiodące do celu: standaryzację i automatyzację wytwarzania.

Standaryzacja prowadząca do uzyskiwania rozwiązań typowych, które mogą być powielane w wielkiej liczbie egzemplarzy, pozwala zmniejszyć koszt, nawet bardzo wysoki, wyprodukowania wzorcowego egzemplarza, rozkładając go na wszystkie sprzedane kopie; koszt powielenia można praktycznie pominać. Rozwiązanie takie jest wystarczająco dobro dla typowych zastosowań, a praktycznie stosu-

Je się je do systemów zmiennoprogramowych o szerokiej specjalizacji, również w wersji minikomputerowej.

Bardziej sensowne dla minikomputerów zastosowania w systemach problemowo zorientowanych, poza być może inteligentnymi urządzeniami końcowymi, raczej nie pozwalają pokładać zbyt wielkich nadziei w standaryzacji. Charakterystyczna dla minikomputerów jest z jednej strony wielkoseryjna produkcja sprzętu, z drugiej wielka różnorodność zastosowań w systemach o wąskiej specjalizacji. Stosowana metoda "dopasowywania" oprogramowania do potrzeb tzn. zawężanie orientacji problemowej - praktycznie sprowadza się do zmian w oprogramowaniu użytkowym, czyli konstruowaniu nowego lub modyfikacji istniejącego programu obliczenia bez naruszania systemu operacyjnego - co bywa ekonomicznie uzasadnione, ale prowadzi częstokroć do konieczności "dopasowywania" obiektu do systemu (przysłowiowy problem nosa i tabakiory).

Najbardziej odpowiednio dla zastosowań nietypowych byłoby oprogramowanie "szyte na miarę". Jednak wciąż wysokie koszty produkcji oprogramowania, a zwłaszcza systemów operacyjnych, nie pozwalają na szerokie stosowanie tego podejścia, co niewątpliwie hamuje rozwój zastosowań techniki komputerowej. Obniżki kosztów poszukiwać należy w automatyzacji, a ściślej w komputerowo wspomaganym procesie produkcji oprogramowania. Wykorzystanie maszyn cyfrowych do produkcji programów jest historycznie jednym z pierwszych nienumerycznych zastosowań techniki komputerowej. Stąd rozwijanie technik kompilacji programów, próby wykorzystywania maszyn cyfrowych do dowodzenia poprawności programów itp. Być może jednak fakt, że ta dziedzina zastosowań uprawiana jest od dawna zaciążył na tendencji do traktowania każdej maszyny cyfrowej jako narzędzia do produkcji programów, niezależnie od szczególnej dziedziny zastosowań, na rzecz której ma działać. W szczególności maszynę cyfrową traktuje się jako narzędzie, które usprawnia produkcję programów dla niej samej.

Podejście to prowadzi albo do rozszerzenia problemu, który ma rozwiązywać dany system komputerowy o problem produkcji programów, co w konsekwencji niweluje korzyści, jakie można uzyskać z wąskiej specjalizacji albo do implementowania na tym samym sprzęcie dwu różnych systemów oprogramowania, działających co prawda nie w tym samym czasie, ale wymagających stosowania sprzętu bardziej rozbudowanego i droższego, niż gdyby każdy z tych systemów realizowano na niezależnych maszynach.

System komputerowo-wspomaganego wytwarzania oprogramowania dla problemowo zorientowanych systemów minikomputerowych powinien być implementowany na maszynie dużej, dobrze oprogramowanej. System minikomputerowy uzyskiwałby przez tę dużą maszynę oprogramowanie dopasowane do przewidywanych zastosowań minikomputera, nie zawierające żadnych zbędnych z punktu widzenia rozwiązywanego problemu elementów, nie powodujące więc żadnych dodatkowych wymagań sprzętowych. Znaczna obniżka kosztu produkcji oprogramowania pozwoliłaby poważnie myśleć o opłacalnym stosowaniu minikomputerów do rozwiązywania różnorodnych nietypowych problemów. Rozwój zastosowań minikomputerów jest więc uwarunkowany rozwojem metod i narzędzi produkcji ich oprogramowania, w tym również systemów operacyjnych.

z cyklu: Programowanie obrabiarek

dr inż. Stanisław DONKOWICZ-SITTAUER
mgr inż. Halina GUTOWSKA
Instytut Maszyn Matematycznych

S Y S T E M A P T *

1. Informacje wstępne

System APT (Automatically Programmed Tools) przeznaczony jest do automatycznej generacji taśm sterujących pracą obrabiarek sterowanych numerycznie (OSN).

System ma własny język o tej samej nazwie (język APT). Programista-technolog za pomocą instrukcji języka APT opisuje proces obróbki detalu, a więc jego kształt oraz sposób wykonania, otrzymując tzw. program obróbki części. Zadaniem systemu APT jest przekształcenie poszczególnych instrukcji języka APT na kody sterujące funkcjami obrabiarki oraz położeniem narzędzia skrawającego względem obrabianego przedmiotu. Działanie systemu opisano w punkcie 2 niniejszego artykułu.

Tworzenie systemu APT rozpoczęto jeszcze w końcu lat pięćdziesiątych na zamówienie powietrznych sił zbrojnych w Massachusetts Institute of Technology (USA). Prototyp systemu stworzono w MIT w 1955 r., a już w 1961 r. powstał długofalowy program rozwoju APT. Program ten stale jest aktualizowany. Prowadzone są prace nad rozwojem systemu APT, związane z potrzebami człowieka, dotyczącymi jak najefektywniejszego sterowania obrabiarkami. W pracach tych bierze aktywny udział około 123 uczestników (Instytucji) z różnych krajów świata (m.in. z Anglii, Francji, Szwecji, RFN, Japonii), ale koordynuje i w pełni odpowiada za całość programu IIT Research Institute (Illinois).

Rozwojem APT interesuje się nie tylko przemysł lotniczy (dla którego początkowo system był tworzony) i nie tylko przemysły pokrowne, jak przemysł stoczniowy, czy też motoryzacyjny, ale również producenci obrabiarek i komputerów.

Pierwsi uważają, że produkując obrabiarki powinni od razu przewidzieć, w jaki sposób będą one sterowane oraz w jaki sposób będą generowane taśmy do tego sterowania.

Producenci komputerów natomiast zdają sobie sprawę, iż w ramach oprogramowania standardowego (lub handlowego) produkowanego sprzętu komputerowego musi znajdować się odpowiedni system do automatycznego generowania taśm sterujących obrabiarkami.

Tak różnorodnej współpracy system APT zawdzięcza stosunkowo dużą uniwersalność zarówno pod względem merytorycznym, głównie w zakresie możliwości odwzorowania kształtów, jak i pod względem możliwości implementacji na komputerach różnego typu.

APT jest systemem uniwersalnym - stwierdzenie to oznacza, że nie jest związany ani z konkretnym typem obrabiarek, ani komputerów. Jednak ze względu na swoją złożoność i rozmiary, wymaga stosowania komputerów o pamięci operacyjnej minimum 256 kB oraz wyposażonych w pamięci pomocnicze o pojemności kilku MB. Tak więc na ogół APT jest implementowany na dużych komputerach, jak IBM-3/30, IBM-S/370, CDC, UNIVAC, IRIS i im podobnych. Dzięki temu jest on łatwo osiągalny, a co za tym idzie stosunkowo szeroko stosowany. Dodatkowo na popularność APT wpływa fakt, iż ma duże możliwości odwzorowania złożonej geometrii kształtu przedmiotu i sterowania ruchem narzędzia.

Jak wiadomo, najszerzej stosowane są trzy rodzaje układów sterowania obrabiarek: punktowe, odcinkowe i kształtowe, różniące się między sobą sposobem przesuwania narzędzia między kolejnymi punktami.

★

Artykuł niniejszy powstał w wyniku prac realizowanych w IMI w Zakładzie DC i opiera się głównie na materiale zawartym w podręczniku programowania [3], [4] oraz własnych doświadczeniach. Pozostałe pozycje wymienione w spisie literatury służyły do wyrobienia sobie ogólnego poglądu na omawianą problematykę i dlatego nie powoływano się w treści artykułu na konkretne pozycje.

W układzie sterowania punktowego narzędzie doprowadzone jest do określonego programu punktu, bez wyznaczania drogi przejścia narzędzia. Jest on zwykle stosowany w wiertarkach.

W układzie sterowania odcinkowego droga przejścia narzędzia między zadanymi punktami jest ściśle określona i zgodna z kierunkiem osi obrabiarki. Jest on zwykle stosowany dla tokarek.

Układ sterowania kształtowego jest najbardziej rozbudowany i obejmuje wszystkie możliwości sterowania punktowego i odcinkowego, a ponadto możliwe jest sterowanie przejściem narzędzia wzdłuż linii prostych nachylonych pod zadanym kątem, po okręgu lub po innej matematycznie określonej krzywej. Układ ten stosowany jest zwykle we frezarkach.

Wyposażenie systemu APT w możliwość sterowania kształtowego spowodowało, iż jest on szczególnie przydatny do przygotowywania programów obróbki przedmiotów o przestrzennie złożonych kształtach dla obrabiarek mających ruchy sterowane numerycznie wzdłuż 3 do 5 osi.

Zrealizowanie postulatów sterowania kształtowego wymaga znacznej rozbudowy systemu w stosunku do systemów umożliwiających sterowanie punktowe, bądź też odcinkowe. Tak więc APT jest systemem o dużych wymaganiach sprzętowych (wielkość pamięci operacyjnej, pamięć pomocnicza), w związku z czym jest systemem kosztownym w eksploatacji. Jest to również spowodowane stosunkowo długimi czasami obliczeń, jakie zazwyczaj system musi wykonać. Ponadto sam język APT stał się bardzo bogaty, ale wymaga większych umiejętności od programisty, jak też może być źródłem popełniania błędów, przez tegoż programistę. Warto jednak zdawać sobie sprawę, że z punktu widzenia odwzorowania operacji technologicznych, system APT ustępuje innym (np. EXAPT-owi).

Zasadniczo system APT jest przeznaczony do eksploatacji w trybie wsadowym, powstała jednak również wersja konwersacyjna tego systemu pod nazwą CP-CMS/APT (Conversational Programming - Cambridge Monitor System) zbudowana przez IBM Los Angeles Science Centre i zaimplementowana na komputerze S/360 model 67.

System APT miał wiele kolejnych wersji, z których pierwszą, naprawdę użyteczną był APT III. Był on napisany w języku FORTRAN II dla mc IBM 709-7090 w 1961 r. Obecnie stosowana wersja - APT IV, o zmienionej strukturze w stosunku do poprzednich, jest napisana w języku FORTRAN IV. Ponadto będziemy się zajmować jedynie tą ostatnią wersją systemu APT.

Na zakończenie tych uwag warto zaznaczyć, iż napisanie systemu APT w języku FORTRAN pociągnęło za sobą różnorodne konsekwencje - miało wpływ zarówno na strukturę systemu i sposób jego działania, jak i na sam język APT, prowadząc do zastosowania różnego rodzaju fortranopodobnych rozrządzeń i konwencji.

2. Działanie systemu

Ostatecznym celem systemu APT jest wygenerowanie taśmy sterującej dla konkretnej OSN opisującej obróbkę konkretnej części. Forma części, zadawana w sposób ścisły instrukcjami języka APT, jest przekształcana przez system na współrzędno poszczególnych punktów pośrednich, tak aby narzędzie skrawające, poruszając się od punktu do punktu po liniach prostych, mogło obrobić daną część zgodnie z wymaganym kształtem i z zadaną dokładnością. Tak więc sformułowane w języku APT instrukcje programu obróbki części zawierają informacje opisujące:

- geometrię (kształt) części, jaką chcemy otrzymać w wyniku procesu obróbki,
- dokładność odwzorowania części (czyli żądany przy interpolacji zakres tolerancji),
- sposób obróbki (a więc m.in. kolejność operacji i jej parametry technologiczne).

W trakcie przetwarzania programu części, z formy zapisanej w języku APT do formy kodu na taśmie sterującej obróbką, można wyróżnić pewne etapy w pracy systemu, będące równocześnie jego podstawowymi funkcjami:

● Translacja językowa

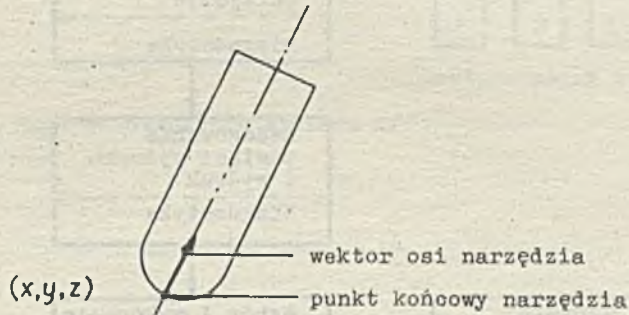
W fazie translacji językowej złożone instrukcje języka APT są redukowane do postaci prostego kłucucha elementarnych komend, zgodnie z regułami gramatycznymi i znaczeniem słów ze słownika APT. Elementarne komendy są kolejno interpretowane, tzn. sterują one innymi funkcjami systemu.

● Ujednolicenie opisu geometrii części

Programista definiuje kształt geometryczny części za pomocą zbioru krzywych i powierzchni takich, jak proste, płaszczyzny, okręgi, elipsoidy itp. Warto zwrócić uwagę, że język APT jest pod tym względem bardzo bogato wyposażony i elastyczny. Pozwala mianowicie na definiowanie tych samych tworów geometrycznych wieloma różnymi sposobami. Tak więc programista ma stosunkowo dużo swobody i może definiować kształt opisywanego detalu w najwygodniejszy dla siebie sposób. Natomiast system APT sam ujednolica te opisy, a mianowicie: dokonuje transformacji układu współrzędnych, wykonuje konieczne analizy opisanych konstrukcji geometrycznych i obliczenia niezbędne do przedstawienia tych opisów w przyjętej, prostej postaci (tzw. postaci kanonicznej). Na przykład: jeżeli programista zdefiniował punkt, odwołując się do definicji dwóch prostych, na przecięciu których leży ten punkt, to system zastępuje ten opis obliczonymi przez siebie współrzędnymi tego punktu.

● Obliczanie pozycji narzędzia

Najbardziej złożoną i pracochłonną funkcją systemu APT jest obliczanie kolejnych pozycji narzędzia na podstawie instrukcji języka APT, zawartych w programie obróbki części i opisujących ruch narzędzia. Pozycje narzędzia są podawane za pomocą współrzędnych x, y, z odnoszących się zawsze do tego samego wybranego punktu leżącego na powierzchni narzędzia (tzw. punktu końcowego narzędzia) oraz cosinusa kąta kierunkowego wektora osi narzędzia (rys. 1). Kolejne pozycje narzędzia definiują tzw. drogą narzędzia.



Rys. 1. Przykład określenia pozycji narzędzia

● Praca programu postprocesora

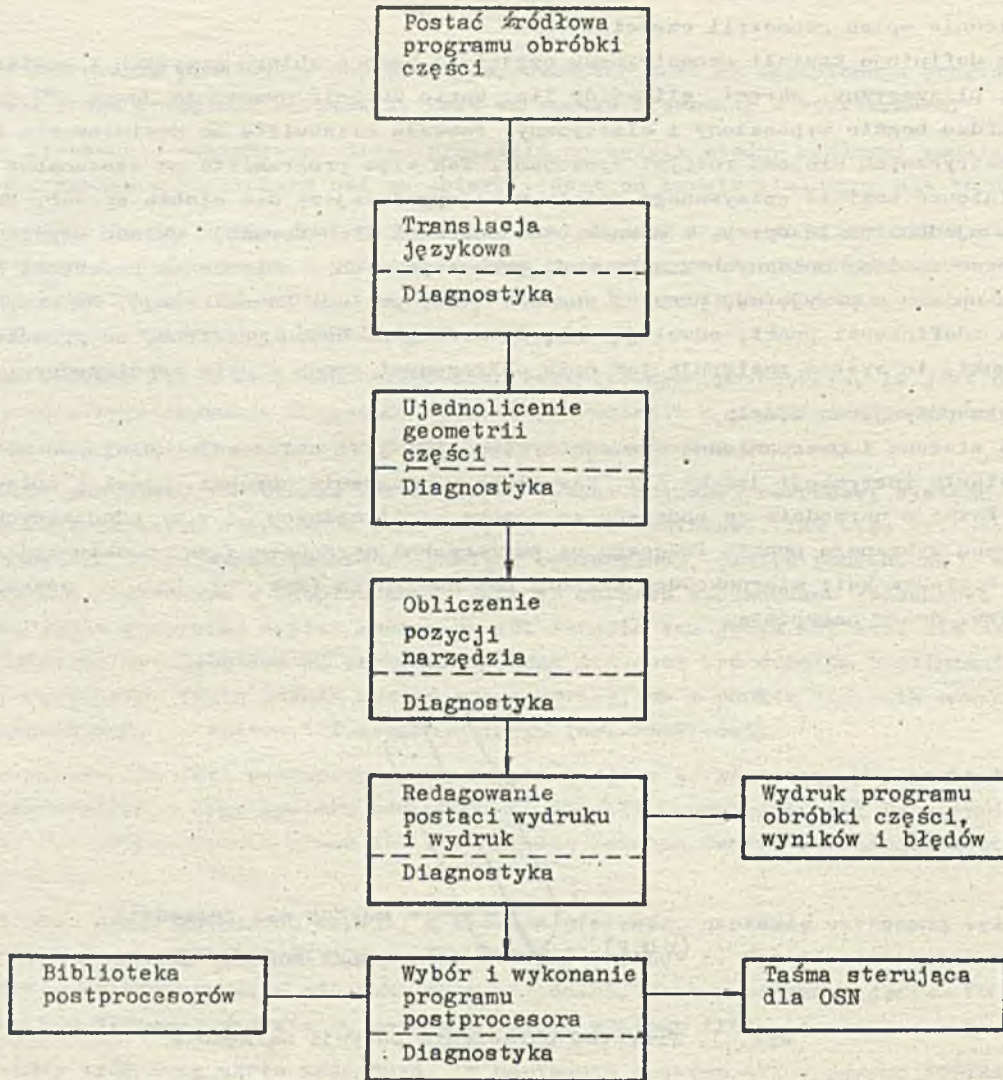
Układy sterowania obrabiarek różnią się między sobą, np. wymaganym kodem znaków na taśmie perforowanej, czy też kodami funkcji obrabiarki. Między uniwersalnym systemem APT a konkretną obrabiarką o pewnych szczególnych wymaganiach, pośredniczy postprocesor. Dokonuje on odpowiedniej zmiany kodu, interpretuje i nadaje znaczenie parametrom technologicznym zdefiniowanym w programie części takim, jak określenie szybkości posuwu, włączenie dopływu chłodziwa itp., a także sprawdza, czy nie zostały przekroczone ograniczenia dynamiczne danej obrabiarki. Wobec powyższego chcąc efektywnie korzystać z uniwersalnego systemu APT, użytkownik musi uzupełnić go pewną liczbą programów postprocesorów, odpowiadających konkretnym obrabiarkom, dla jakich system ma generować taśmy sterujące (rys. 2).

Wyżej wymienione funkcje są równocześnie kolejnymi fazami pracy systemu. Natomiast w odmienny sposób system realizuje funkcję

● Diagnostyczną, czyli wykrywanie błędów

W celu wykrywania błędów wszystkie wymienione wyżej fazy pracy systemu mają wbudowane procedury diagnostyczne. Tak więc wykrywane są błędy syntaktyczne, geometryczne, przesunięcia narzędzia w niedozwolony lub niemożliwy do wykonania sposób itp. Standardowe procedury diagnostyczne działają tak, że w wypadku napotkania określonego błędu drukowany jest jego numer wraz z komunikatem opisującym rodzaj błędu.

Innym sposobem sprawdzania poprawności programu może być, zaplanowane przez programistę piszącego dany program obróbki części, bądź wydrukowanie zawartości tzw. zbioru CLDATA (Cutler Location DATA) - czyli współrzędnych kolejnych pozycji narzędzia, bądź wykreślenie drogi narzędzia na autokreślarec. Kontrolę taką należy wykonać jeszcze przed wygenerowaniem taśmy sterującej pracą ob-



Rys. 2. Schemat działania systemu APT

rubiarki. Kolejny sposób kontroli jest w zasadzie pozasystemowy, gdyż polega on na próbnej pracy obrabiarki sterowanej wygenerowaną taśmą, realizującej obróbkę części w materiale zastępczym, który w wypadku nieprawidłowo zaplanowanego procesu obróbczego nie spowodowałby zniszczenia narzędzia lub uszkodzenia całej maszyny.

3. Struktura systemu APT

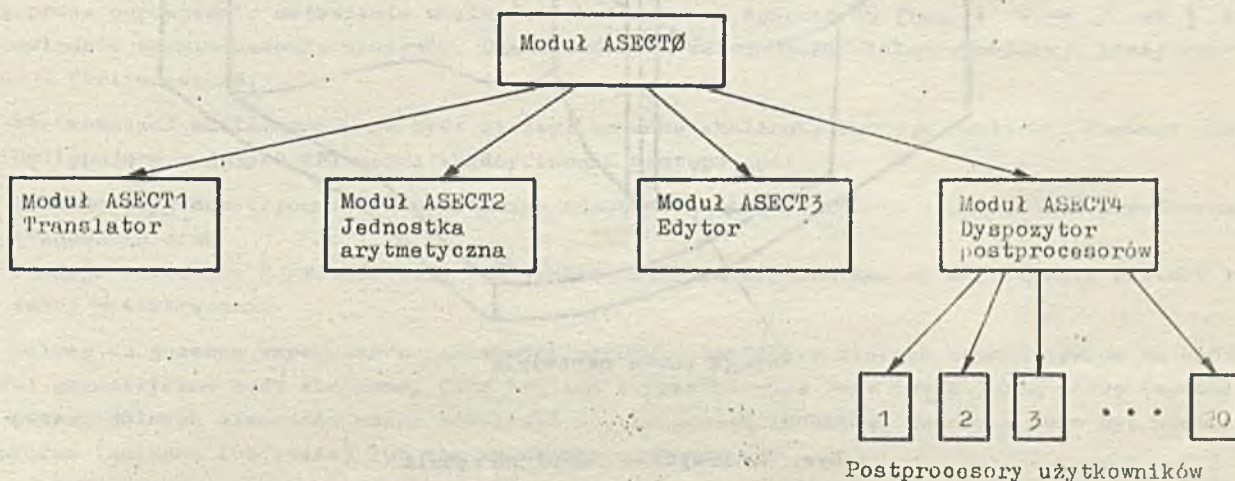
Realizacja programowa systemu APT będzie omówiona na przykładzie implementacji tego systemu na mc IBM S/360.

System APT składa się z 25 modułów, przy czym 20 modułów przeznaczonych jest na umieszczenie 20 postprocesorów, napisanych przez użytkownika. Pozostałe 5 modułów pełni w systemie następujące funkcje:

- ASECT 0 - Jest modułem sterującym, komunikuje się z systemem operacyjnym oraz wszystkimi pozostałymi modułami systemu APT, które nie mogą się między sobą bezpośrednio komunikować; wykonuje on również wszystkie operacje WE/WY,
- ASECT 1 - Jest modułem translatora, który dokoduje instrukcje programu obróbki części,
- ASECT 2 - Jest modułem jednostki arytmetycznej, która oblicza kolejne pozycje narzędzia skrawającego,

- ASECT 3 - jest modulem edytora, który wykonuje wszystkie funkcje związane z redagowaniem, kopiowaniem drogi narzędzia, wydrukami i krośloniem drogi narzędzia na autokreślarec,
- ASECT 4 - jest modulem dyspozytora postprocesorów, który wybiera żądany przez program obróbki części postprocesor i przekazuje sterowanie do modułu zawierającego ten postprocesor.

Ogólny schemat struktury systemu APT przedstawiony jest na rys. 3.



Rys. 3. Struktura systemu APT

4. Charakterystyka języka APT

U w a g i o g ó l n e

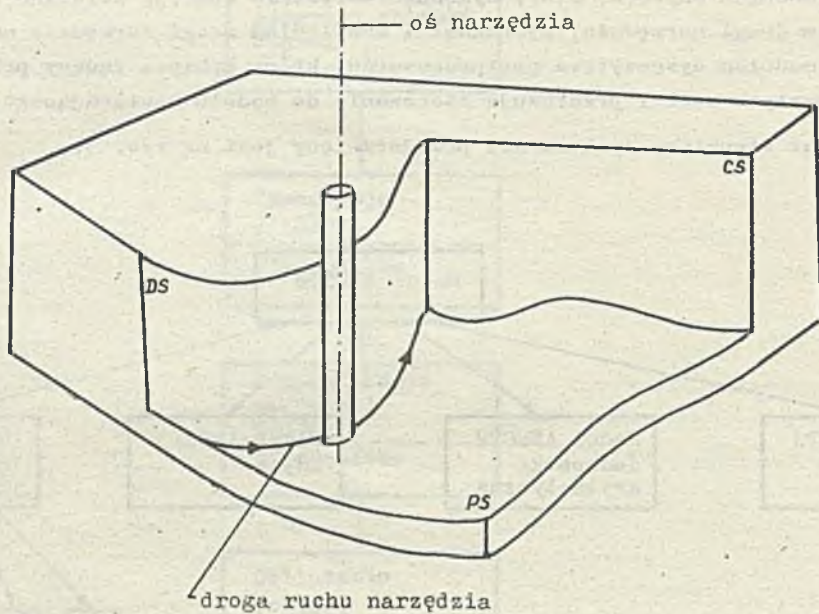
Program obróbki części składa się ze zbioru instrukcji pisanych wg reguł problemowo-zorientowanego języka APT. Zwykle około 70% programu stanowią instrukcje definiujące kształt geometryczny części. Pozostała część instrukcji opisuje ruch narzędzia skrawającego, funkcje obrabiarki oraz steruje pracą samego systemu.

W systemie APT przyjęto zasadę, że opisując poruszając się względem siebie narzędzie i przedmiot obrabiany, zawsze traktuje się przedmiot jako nieruchomy, a porusza się narzędzie skrawające, bez względu na to, jak w rzeczywistości odbywa się proces obróbczy. Założenie to było konieczne dla nadania jednolitej postaci opisu wszystkim prawdopodobnym procesom obróbczym. Dopasowanie takiego opisu do konkretnych cech stosowanej obrabiarki realizuje dopiero postprocesor.

Kształt, jaki ma przyjęć obrabiana część, jak uprzednio wspomiano, zdefiniowany jest zbiorem powierzchni, zaś każde przesunięcie narzędzia wymaga określenia trzech powierzchni (PS, DS, CS) przecinających się w jednym punkcie, względem których odbywa się dany ruch roboczy. W czasie ruchu roboczego narzędzie musi cały czas dotykać (rys. 4) do powierzchni, zwanej powierzchnią przedmiotu - PS (part surface). Równocześnie powinno ono poruszać się po pewnej zadanej drodze. W tym celu programista definiuje tzw. powierzchnię prowadzącą - DS (drive surface), do której ma być równoległa oś poruszającego się narzędzia. Położenie osi narzędzia względem powierzchni DS może być różne i jest definiowane za pomocą tzw. modyfikatorów. Szczegółowo sprawa ta jest omówiona w punkcie "Instrukcje ruchu narzędzia". Elementarny ruch narzędzia, zdefiniowany przy pomocy jednej instrukcji, kończy się, gdy narzędzie osiągnie tzw. powierzchnię ograniczającą - CS (check surface). Przykład takiego ruchu narzędzia przedstawiony został na rys. 4.

Powierzchnią obrabianą może być zarówno PS, jak i DS - zależy to od rodzaju zastosowanego narzędzia oraz typu obrabiarki, najczęściej jednak jest to PS.

Dla każdego elementarnego ruchu roboczego w zasadzie definiuje się inną powierzchnię prowadzącą i powierzchnię ograniczającą, natomiast powierzchnia przedmiotu pozostaje najczęściej bez zuba-



Rys. 4. Przykład ruchu narzędzia

ny. Taki sposób daje bardzo duże możliwości kierowania ruchem narzędzia i jest szczególnie przydatny do sterowania kształtowego.

Instrukcje języka APT tworzone są z elementów:

● Słowa główne i pomocnicze języka APT, tworzące słownik języka APT. W języku APT zdefiniowano zbiór słów o ściśle określonym znaczeniu. Wyróżniamy wśród nich:

słowa główne, które określają typy powierzchni geometrycznych

(np. POINT, CIRCLE), funkcje arytmetyczne lub geometryczne (np. SINF, ABSF, DISTF), rodzaj ruchu (np. GO, GORGT), funkcje obrabiarki (np. FEDRAT, END) lub też sposób wykonania operacji

oraz słowa pomocnicze, używane jako modyfikatory (np. TO, PAST).

● Symbole

Symbolami (nazwami) są słowa utworzone przez programistę, które identyfikują zdefiniowany przez niego obiekt, co umożliwia późniejsze odwoływanie się do tego obiektu w programie. Symbol może się składać z co najwyżej 6 znaków alfanumerycznych, z których co najmniej jeden musi być literą oraz musi być różny od dowolnego słowa ze słownika APT. Symbol powinien być zdefiniowany, zanim wystąpi pierwsze odwołanie do niego.

● Liczby - przedstawione jako całkowite lub rzeczywiste (w postaci dziesiętnej bądź w postaci ułamkowej).

● Znaki specjalne . / + - , \$ % & * **

są używane do rozdzielania poszczególnych elementów w instrukcjach oraz do określenia operacji arytmetycznych.

● Etykiety-instrukcji

Etykieta umożliwia odwoływanie się do takiej instrukcji w innych instrukcjach tegoż programu. Etykiety budowane są z ciągu co najwyżej 6 znaków alfanumerycznych, z których wszystkie mogą być cyframi. Umieszcza się je z lewej strony instrukcji i oddziela od niej nawiasem zamykającym. Pojedyncza instrukcja może składać się z co najwyżej 600 wyżej wymienionych elementów.

Ze względu na funkcje pełnione w programie obróbki części, instrukcje języka APT można podzielić na:

- Instrukcje arytmetyczne,
- Instrukcje sterujące pracą komputera,
- Instrukcje definiujące (definicje),
- Instrukcje postprocesora.
- Instrukcje ruchu narzędzia,

Instrukcje arytmetyczne

Służą do nadania symbolom wartości liczbowych wynikających z wykonania odpowiednich operacji arytmetycznych zdefiniowanych w postaci wyrażenia arytmetycznego. Wyrażenie arytmetyczne tworzy się przez odpowiednie ustawienie wielkości skalarnych i operatorów (tzn. + - * / **) oraz odpowiednie rozmieszczenie nawiasów. Dla określenia kolejności działań obowiązują klasyczne konwencje Fortranowskie.

Wielkościami skalarnymi mogą być: liczby, zmienne skalarne, funkcje skalarne. Funkcje skalarno występujące w języku APT można sklasyfikować następująco:

- funkcje trygonometryczne i arytmetyczne odpowiadające odpowiednim standardowym procedurom Fortranowskim oraz
- funkcje wektorowe i geometryczne tzn. takie, których argumentami są odpowiednio wektory lub twory geometryczne.

Należy tu jeszcze wspomnieć o możliwości używania tablic, w których zapamiętywane są bądź wielkości geometryczne bądź skalarne. Cała tablica reprezentowana jest przez jedną nazwę (symbol), a do poszczególnych elementów można odwoływać się za pomocą indeksów. Indeksom może być wielkość skalarna (zmienna lub stała) lub też wyrażenie arytmetyczne.

Definicje

Język APT zawiera bardzo bogaty zestaw instrukcji umożliwiających definiowanie w różny sposób różnorodnych tworów i kształtów geometrycznych, takich jak:

- powierzchnie analityczne - proste, płaszczyzny, walce i inne powierzchnie, które można opisać za pomocą równań,
- powierzchnie uzyskiwane doświadczalnie opisane są za pomocą zbioru punktów, które stanowią podstawę do interpolacji (np. TABCYL),
- zbiory punktów. Jako zbiory punktów można definiować punkty leżące na prostej, na łuku okręgu, na przecięciu się dwóch rodzin prostych równoległych oraz stanowiące dowolną kombinację punktów i zbiorów punktów.

Należy zwrócić uwagę, iż język APT umożliwia definiowanie tych samych tworów geometrycznych różnymi sposobami. Dzięki temu, jak już wspomniano, język jest bardzo elastyczny i daje programiście wiele ułatwień, co jest jednak okupione koniecznością przeprowadzania przez system standaryzacji postaci definicji.

Definicja geometryczna ma następującą postać:

nazwa symboliczna = rodzaj elementu zgodnie ze słownikiem APT	/	parametry opisujące
--	---	---------------------

Przykłady instrukcji definiujących języka APT wraz z ich możliwościami przedstawione są w tabeli.

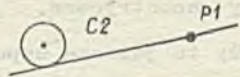
Oprócz elementów wymienionych w tabeli, w APT można również definiować także powierzchnie analityczne, jak stożki (CONE), kule (SPHERE), krzywo drugiego stopnia (GCONIC), krzywo czwartego stopnia (LCONIC) oraz powierzchnie stopnia drugiego (QUADRIC).

Można też definiować krzywe, które powstają przez interpolację w uporządkowanym zbiorze punktów - przesunięcie proste wzdłuż takiej krzywej określa powierzchnię zwaną, wspomnianym już, tabelarycznym walcem (TABCYL).

Można też definiować powierzchnie prostokątne (instrukcja RLDSRF), które powstają przez połączenie liniami prostymi odpowiednich punktów leżących na dwóch zadanych krzywych przestrzennych, co jest szczególnie użyteczne w konstrukcjach lotniczych.

Przykłady definicji geometrycznych języka APT

Lp.	Nazwa definiowanego tworzu geometrycznego	Nazwa w języku APT	Liczba sposobów definiowania	Przykłady instrukcji definiujących	Omówienie przykładu definicji (znaczenie)
1	Punkt	POINT	12	P1=POINT/10,-6,18 P2=POINT/INTOF,L3,L5	Punkt o nazwie P1 jest punktem o współrzędnych 10,-6,18 Punkt P2 leży na przecięciu prostej L3 i L5 (L3 i L5 są nazwami wcześniej zdefiniowanych prostych)
2	Prosta	LINE	16	L1=LINE/P1,P2 L2=LINE/P1,LEFT,TANTO,C2	Prosta L1 przechodzi przez 2 punkty P1 i P2 Prosta L2 przechodzi przez punkt P1 i jest styczna po lewej stronie (patrząc od punktu P1) do okręgu C2
3	Okrąg	CIRCLE	10	C1=CIRCLE/CENTER,P1,RADIUS,15	Okrąg C1 ma środek w punkcie P1 i promień 15
4	Wektor	VECTOR	10	WEKT=VECTOR/N1,CROSS,V2	Wektor WEKT jest iloczynem wektorowym wektorów N1 i V2
5	Płaszczyzna	PLANE	7	PL1=PLANE/P1,P2,P3	Płaszczyzna PL1 przechodzi przez punkty P1,P2,P3
6	Walec	CYLINDR	2	WAL=CYLINDR/5,5,5,0,0,1,15	Walec WAL o promieniu 15, wektorze kierunkowym osi o składowych 0,0,1. Punkt 5,5,5 leży na osi walca



Bardzo przydatne są również definicje macierzy (MATRIX), które umożliwiają określanie przekształceń układu współrzędnych (np. przez obrót, przesunięcie itp.). Po odpowiednim zdefiniowaniu macierzy podanie jej w instrukcji REFSYS powoduje odpowiednie przekształcenie układu współrzędnych dla następujących po niej definicji geometrycznych.

Oprócz kształtu geometrycznego programista może również definiować kształt narzędzia skrawającego - za pomocą instrukcji CUTTER, a także pasmo tolerancji, w którym są interpolowane uprzednio zdefiniowane krzywe (instrukcje INTOL, OUTTOL, TOLER). Należy pamiętać, że im węższe określi się pasmo tolerancji, tym więcej odcinków prostoliniowych interpolujących zadaną krzywą system będzie zmuszony wygenerować.

Instrukcje ruchu narzędzia

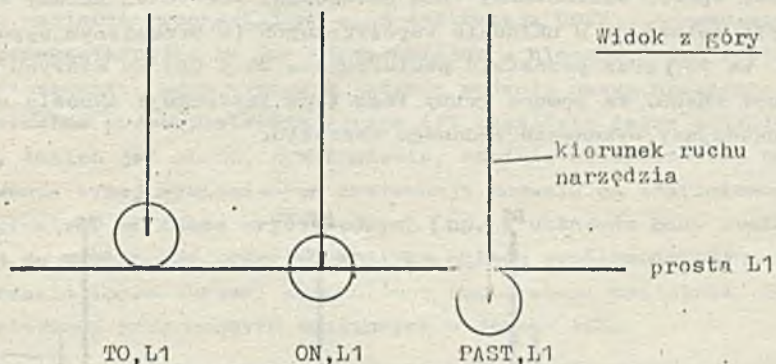
Przed pierwszą instrukcją ruchu narzędzia w programie obróbki części należy umieścić instrukcję, która definiuje punkt, z którego należy rozpocząć ten ruch. Ma ona postać:

FROM/określenie punktu

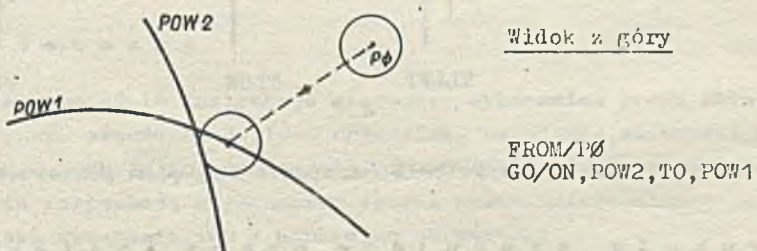
Następnie narzędzie jest ustawiane w pozycji roboczej za pomocą instrukcji GO. W skład tej instrukcji wchodzi m.in. modyfikatory: TO, ON i PAST, określające położenie narzędzia względem jednej lub kilku powierzchni. Znaczenie tych modyfikatorów podaje rys. 5, a sposób ich interpretacji rys. 6.

Po instrukcji GO następuje ciąg instrukcji określających ruchy robocze narzędzia. Zakłada się przy tym, że podczas ruchu roboczego narzędzie:

- styka się z powierzchnią przedmiotu (PS),
- porusza się wzdłuż powierzchni prowadzącej (DS),
- jest zatrzymywane przez powierzchnię ograniczającą (CS).

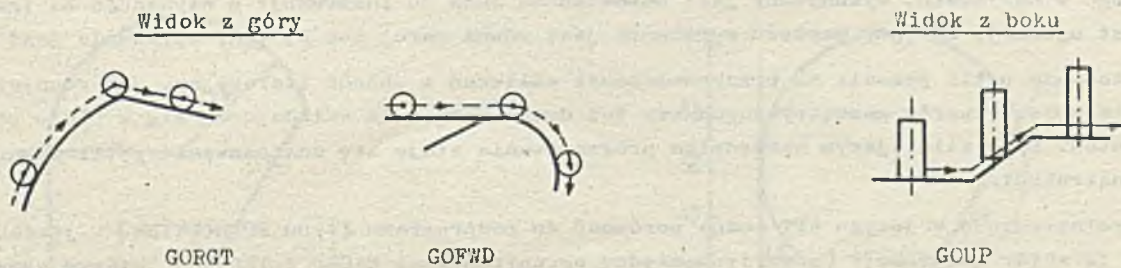


Rys. 5. Znaczenie modyfikatorów TO, ON i PAST



Rys. 6. Przykład ustawienia narzędzia do ruchu roboczego

Natomiast chcąc określić kierunek ruchu narzędzia stosuje się takie słowa kluczowe języka APT, jak: GOLFT (w lewo), GORGT (w prawo), GOFWD (do przodu), GOBACK (do tyłu), GOUP (do góry) i GOWDN (do dołu) - patrz rys.7.; a w celu określenia położenia narzędzia względem powierzchni prowadzącej - używa się modyfikatorów: TLRGT - oś narzędzia leży na prawo od powierzchni prowadzącej, TLLFT - oś narzędzia leży na lewo od powierzchni prowadzącej, TLRON - oś narzędzia leży na powierzchni prowadzącej. Stronę powierzchni prowadzącej określa się patrząc zgodnie z kierunkiem ruchu narzędzia. Odpowiedni przykład przedstawia rys.8.

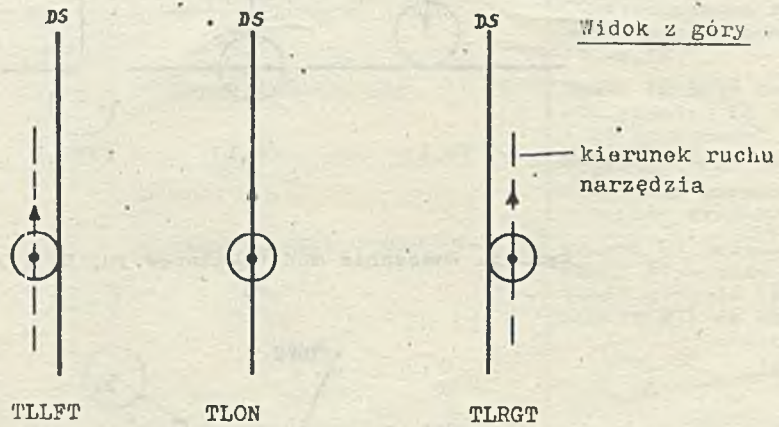


Rys. 7. Sposoby określania kierunku ruchu narzędzia

W instrukcji ruchu podaje się więc:

- ustawienie narzędzia względem powierzchni prowadzącej,
- kierunek ruchu narzędzia,
- powierzchnię prowadzącą dla danego ruchu narzędzia,
- powierzchnię ograniczającą dla danego ruchu narzędzia (wraz z odpowiednim modyfikatorem TO, ON i PAST),

definiując w ten sposób elementarny ruch narzędzia. Wcześniej należy zdefiniować powierzchnię przedmiotu czyli opisać ją w układzie współrzędnych (w przeciwnym wypadku systemu APT uzna płaszczyznę $z = 0$ za PS) oraz pozostałe powierzchnie DS i CS, do których odwołujemy się przez ich nazwy instrukcji ruchu. Za pomocą grupy tego typu instrukcji określa się ruch narzędzia (drogę narzędzia), powodujący wykonanie zadanego kształtu.



Rys. 8. Ustawienie narzędzia względem powierzchni prowadzącej

Instrukcje sterujące pracą systemu

W systemie APT można napisać stosunkowo niewielki, ze względu na ilość instrukcji, program, opisujący dosyć skomplikowaną część. Wielokrotne wykonanie tego samego zestawu czynności, bez konieczności powtarzania zbioru instrukcji opisujących tego rodzaju działanie, osiąga się w programie przez tworzenie pętli, makroinstrukcji, lub przez powtarzanie fragmentu uprzednio zdefiniowanej drogi narzędzia w innym, dowolnym miejscu tego samego programu obróbki części (instrukcje TRACUT i COPY).

Pętlę tworzy się za pomocą instrukcji skoku warunkowego:

IF (wyrażenie) E1,E2,E3

gdzie umieszczone w nawiasach wyrażenie, może być zmienną skalarną lub dowolnym wyrażeniem arytmetycznym o określonej, w danym momencie, wartości, zaś E1,E2,E3 są etykietami instrukcji, do których należy przejść podczas wykonywania się programu. Działanie tej instrukcji jest analogiczne jak instrukcji arytmetycznej IF w języku FORTRAN, a mianowicie, w zależności od znaku wyrażenia, zawartego w nawiasach, wykonywany jest odpowiednio skok do instrukcji o etykiecie E1 (gdy wyrażenie jest ujemne), E2 (gdy wartość wyrażenia jest równa zero) lub E3 (gdy wyrażenie jest dodatnie).

Stosowanie pętli pozwala na przeprowadzenie obliczeń w sposób iteracyjny, jak również na definiowanie zbioru tworów geometrycznych czy też drogi narzędzia składającej się z wielu podobnych fragmentów. Dużo silniejszym narzędziem programowania staje się zastosowanie pętli wewnątrz makroinstrukcji.

Makroinstrukcję w języku APT można porównać do podprogramu (typu SUBROUTINE) w języku FORTRAN - jest to zbiór instrukcji (zawartych między ogranicznikami MACRO i TERMAC), którym nadano pewną nazwę. Nagłówek makroinstrukcji:

nazwa makroinstrukcji = MACRO/lista parametrów formalnych

zawiera jej nazwę oraz nazwy parametrów formalnych, oddzielonych od siebie przecinkami. Zdefiniowaną w taki sposób makroinstrukcję można następnie wywołać instrukcją CALL w dowolnym miejscu programu obróbki części, nadając parametrom makroinstrukcji konkretne wartości.

Zo względu na szczególnie ukierunkowanie języka APT niezwykle użyteczne wydają się być instrukcje, pozwalające na powtórzenie i ewentualnie przekształcenie już zdefiniowanego fragmentu drogi narzędzia. Do tego celu służy instrukcja TRACUT - powodująca przekształcenie fragmentu drogi na-

rzędzia zgodnie z określoną macierzą przekształceń oraz instrukcja COPY - powodująca skopiowanie fragmentu drogi narzędzia przekształcając go lub pozostawiając w niezminionej postaci. Wymienione instrukcje są proste w stosowaniu, gdyż wymagają jedynie podania nazwy uprzednio zdefiniowanej macierzy opisującej potrzebne przekształcenie (język APT umożliwia łatwo definiowanie macierzy prostych przekształceń, takich jak obrót, przesunięcie, odbicie zwierciadlane oraz składanie tych przekształceń). Stosowanie wyżej wymienionych instrukcji pozwala na zdefiniowanie drogi narzędzia w wygodnym dla programisty układzie współrzędnych (np. w układzie bazy wymiarowej), a następnie przekształcenie jej do wymaganego przez obrabiarkę układu współrzędnych; można je też użyć do kilkakrotnego powtórzenia opisu obróbki elementów o takim samym kształcie. Opisane konstrukcje są tylko przykładami ułatwień programowych dostępnych w języku APT.

Pełne wykorzystanie możliwości systemu znacznie podnosi efektywność programowania, w sensie zmniejszania pracochłonności i możliwości popełnienia błędu, wymaga jednak od programisty dogłębnej znajomości języka APT, co przy ogromnej liczbie instrukcji tego języka (nieporównywalnie większej od języka FORTRAN czy ALGOL) bo ok. 150 instrukcji, jest trudne i może być zrealizowane dopiero po dłuższym okresie programowania w tym języku.

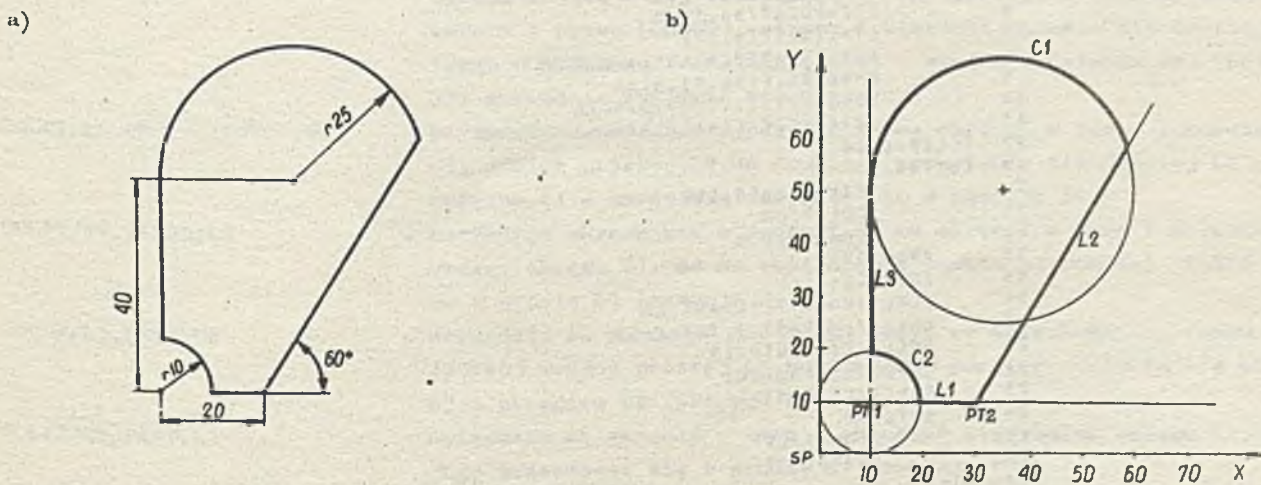
Instrukcje postprocesora

W programie obróbki części należy umieścić instrukcje sterujące wykonaniem przez obrabiarkę poszczególnych funkcji takich, jak np. włączenie dopływu chłodziwa, ustalenie szybkości posuwu, czy też szybkości obrotów wrzeciona. Funkcje te pełnią rolę pomocniczą w stosunku do samej obróbki. Nie wszystkie układy sterowania rozpoznają w jednakowy sposób słowa uniwersalnego języka, dlatego też zadaniem postprocesora jest związanie ich z konkretną obrabiarką.

W języku APT jest zarezerwowana duża grupa słów oraz instrukcji z parametrami, które interpretuje dopiero program postprocesora. Należy je stosować tylko w powiązaniu z konkretnym postprocesorem. Niektórym słowom, takim jak SPINDL, COOLNT, FEDRAT przypisuje się zwykle takie samo znaczenie (punkt 5) - aczkolwiek nie jest to obowiązujące.

5. Przykład programu w APT

W celu przybliżenia czytelnikowi problemów programowania w języku APT, w tym punkcie przedstawimy przykład programu obróbki części oraz pokrótce opisujemy metodę postępowania prowadzącą do otrzymania takiego programu. Wszelkie rozważania są przedstawione na podstawie przygotowania programu obróbki części pokazanej na rys. 9. Wybrana część jest płaskim detałem o kształcie dającym się opisać za pomocą prostych i okręgów.



Rys. 9. Część do obróbki

- a) rysunek techniczny części
- b) sposób odwzorowania kształtu części w programie obróbki

Nieskomplikowany kształt przeznaczanego do obróbki detalu wymaga bardzo krótkiego programu obróbki części, co pozwala na szczegółową analizę przebiegu prac.

Podstawowym materiałem do napisania programu jest plan obróbki części. W planie tym ustala się:

- kolejność operacji technologicznych wykonywanych w czasie obróbki,
- narzędzia do wykonania poszczególnych operacji,
- parametry obróbki (np. szybkość posuwu i inne).

Do planu obróbki dołącza się rysunek konstrukcyjny części.

Programista, przystępując do pisania programu obróbki części, powinien przede wszystkim ustalić układ współrzędnych, w jakim będzie definiował geometrię części w programie i nanieść go na rysunek części. Ponadto musi on zdefiniować kształt geometryczny części, wykorzystując do tego celu dostępne w języku APT twory geometryczne, opisane w punkcie 4 niniejszego artykułu. Dla zdefiniowania poszczególnych tworów opisujących kształt części, nieraz konieczne jest definiowanie pewnych elementów pomocniczych, nie wynikających bezpośrednio z kształtu części. I tak punkty PT1 oraz PT2 w przedstawionym programie potrzebne są jako konstrukcje pomocnicze do zdefiniowania prostych L1 oraz L2.

Zdefiniowane w programie krzywe bądź powierzchnie, powinny się ze sobą łączyć (przecinać się lub być styczne) w taki sposób, aby odpowiednie ich fragmenty łącznie opisywały zamknięty kontur części.

W następnej fazie tworzenia programu obróbki części należy opisać ruch narzędzia wzdłuż konturu, wykorzystując do tego celu instrukcje ruchu scharakteryzowane w punkcie 4 oraz zdefiniować potrzebne do tego ruchu odpowiednie powierzchnie: przedmiotu (PS), prowadzącą (DS) i ograniczającą (CS). W końcowej fazie programista musi zdefiniować: kształt narzędzi do obróbki części, parametry obróbki (zgodnie z planem obróbki) oraz wstawić instrukcje pomocnicze takie, jak instrukcje początku i końca programu, zdefiniowanie nazwy postprocesora, który będzie przetwarzać wyniki z danego programu obróbki części, instrukcje powodujące otrzymanie odpowiednich wydruków itp.

Poniżej przedstawiono wydruki tekstu programu, powodującego obróbkę części z rys. 9; który powstał w omówiony wyżej sposób, a następnie kolejno omówiono znaczenie poszczególnych instrukcji w programie.

```
1 PARTAO PRZYKŁAD PROGRAMU CZĘŚCI W APT
2 CLPRNT
3 NOPOST
4 SP=POINT/0,0,0
5 PT1=POINT/10,10,0
6 PT2=POINT/30,10,0
7 L1=LINE/PT1,PT2
8 L2=LINE/PT2,ATANGL,60,XAXIS
9 C1=CIRCLE/35,50,0,25
10 L3=LINE/10,10,0,10,50,0
11 C2=CIRCLE/CENTER,PT1,RADIUS,10
12 TOLER/C.20
13 CUTTER/2
14 SPINDL/2000,CLW
15 COOLNT/ON
16 FEDRAT/20.0
17 FROM/SP
18 GC/TO,L1
19 TLRGT,GORGT/L1,PAST,L2
20 GOLFT/L2,PAST,2,INTOF,C1
21 GOLFT/C1,TANTO,L3
22 GCFWD/L3,PAST,C2
23 GOLFT/C2,PAST,L1
24 GUTL/SP
25 COOLNT/OFF
26 SPINDL/OFF
27 FINI
```

PARTNO	PRZYKŁAD PROGRAMU CZĘŚCI W APT - identyfikuje część i nadaje tytuł wydrukowi programu oraz taśmie sterującej
CLPRNT	- powoduje wydruk zawartości zbioru CLDATA (rys.11), stanowiącego informacje wejściowe dla programu postprocesora. Zbiór CLDATA powstaje w czasie wykonywania się programu obróbki części a ściśle - tworzony jest przez moduł ASECT2 (punkt 3 niniejszego artykułu),
NOPOST	- wskazuje, że nie będzie wołany program postprocesora,
SP=POINT/0,0,0	- definiuje punkt o współrzędnych (0,0,0) i nadaje mu nazwę SP,
PT1=POINT/10,10,0	- definiuje punkt PT1 o współrzędnych (10,10,0)
PT2=POINT/30,10,0	- definiuje punkt PT2 o współrzędnych (30,10,0)
L1=LINE/PT1,PT2	- definiuje prostą L1 jako przechodzącą przez punkty PT1, PT2
L2=LINE/PT2,ATANGL,60,XAXIS	- definiuje prostą L1 nachyloną pod kąt 60° do osi OX i przechodzącą przez punkt PT2
C1=CIRCLE/35,50,0,25	- definiuje okrąg o środku w punkcie (35,50,0) i promieniu 25
L3=LINE/10,10,0,10,50,0	- definiuje prostą L3 jako przechodzącą przez punkty (10,10,0) i (10,50,0)
C2=CIRCLE/CENTER,PT1,RADIUS,10	- definiuje okrąg C2 o środku w punkcie PT1 i promieniu 10
TOLER/0.20	- definiuje zakres tolerancji dla interpolacji krzywej jako równy 0,20
CUTTER/2	- definiuje kształt narzędzia skrawającego dla następnych instrukcji ruchu jako walec o średnicy 2 i standardowej wysokości 5 (przyjętej przez system APT)
SPINDL/2000,CLW	- ustawia szybkość obrotów wrzeciona w obrabiarce na 2000 obr/min, przy czym kierunek obrotu jest zgodny z kierunkiem ruchu wskazówek zegara
COOLNT/ON	- włącza dopływ chłodziwa w obrabiarce
FEDRAT/20.0	- ustawia szybkość posuwu na 20 cali/min
FROM/SP	- określa, że narzędzie ma rozpocząć ruch z punktu początkowego SP (pozycja P1 na rys. 10)
GO/TO,L1	- narzędzie będzie przesunięte po najkrótszej drodze od punktu początkowego (SP) aż do uzyskania styczności położenia TO z prostą L1, a więc znajdzie się w pozycji P2 (rys.10)
TLRGT,GORGT/L1,PAST,L2	- określa następujące parametry: narzędzie skrawające będzie przesuwać się po prawej stronie (TLRGT) części wzdłuż prostej L1 (która jest powierzchnią prowadzącą dla tego ruchu) aż do momentu uzyskania położenia PAST względem prostej L2 (która jest powierzchnią ograniczającą w tej instrukcji); przy prostej L1 należy dokonać zwrotu w prawo (GORGT); strony i kierunki określa się zawsze patrząc w kierunku ruchu narzędzia; w wyniku działania tej instrukcji narzędzie znajdzie się w pozycji P3
GOLFT/L2,PAST,2,INTOF,C1	- narzędzie skrawające z pozycji P3 ma skrócić w lewo i poruszać się wzdłuż prostej L2 do drugiego przecięcia się prostej L2 z okręgiem C1 - narzędzie znajdzie się w pozycji P4
GOLFT/C1,TANTO,L3	- narzędzie skrawające z pozycji P4 ma skrócić w lewo i poruszać się wzdłuż okręgu C1, aż do osiągnięcia punktu styczności TANTO okręgu z prostą L3 (pozycja P5)
GOFWD/L3,PAST,C2	- narzędzie ma poruszać się w tym samym co dotychczas kierunku (GOFWD) wzdłuż prostej L3,poza (PAST) punkt przecięcia się prostej L3 z okręgiem C2 (pozycja P6)
GOLFT/C2,PAST,L1	- narzędzie ma skrócić w lewo i poruszać się wzdłuż okręgu C2, poza jego przecięcie się z prostą L1 (pozycja P7)
GOTO/SP	- narzędzie przesunie się po najkrótszej drodze do punktu SP (pozycja P1)
COOLNT/OFF	- wyłącza dopływ chłodziwa w obrabiarce
SPINDL/OFF	- powoduje wyłączenie obrotów wrzeciona
FINI	- oznacza koniec programu obróbki części.

```

--SECTION 3--
ARTIC PRZYKŁADOWY PROGRAM OBROBKI CZĘŚCI W APT
OUTFIL/ 0.200000 0.200000 0.200000 0.200000
INTOL/ 0.0 0.0 0.0 0.0
CUTTER/ 2.000000

SPINDL/ 2000.0000. 2LW
COCLNT/ ON
FEEDRAT/ 20.0000

*** FROM/ SP ( 0) 0.0 0.0 0.0 0.0
*** SO---/ L1 ( 0) 0.0 0.000000 0.0
*** GO---/ L1 ( 0) 30.5773303 0.000000 0.0
*** GO---/ L2 ( 0) 59.6129473 58.9447192 0.0

SURFACE DATA - CIRCLE ( DS ) ** C1 ( 0) ** IMPL. TO **
CANON FORM IS 35.0000000. 50.0000000. 0.0 0.0 0.0000000.
*** GO---/ C1 ( 0) 58.3890430 61.8064873 0.0
55.0388359 64.8783283 0.0
50.609755 71.0470689 0.0
43.3608591 74.0729440 0.0
39.8151134 75.8020172 0.0
33.4663275 74.1538927 0.0
27.8611730 73.0919326 0.0
21.8818195 72.6793593 0.0
17.0083933 69.0459782 0.0
13.1038112 64.3873874 0.0
10.3776603 58.9543502 0.0
8.8768843 51.0993170 0.0
9.0000000 50.0000000 0.0

*** GO---/ L3 ( 0) 9.0000000 18.9442719 0.0

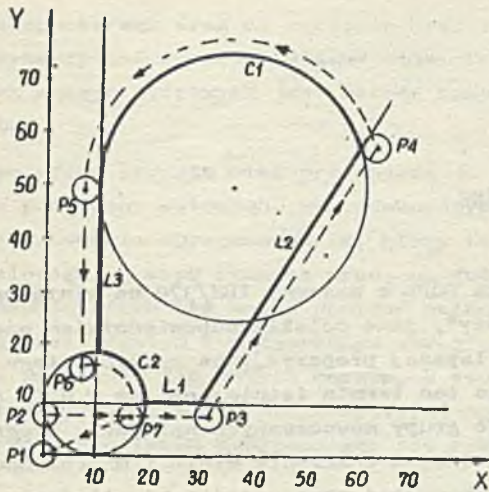
SURFACE DATA - CIRCLE ( DS ) ** C2 ( 0) ** IMPL. TO **
CANON FORM IS 10.0000000. 10.0000000. 0.0 0.0 0.0000000.
*** GO---/ C2 ( 0) 10.0113497 18.9791938 0.0
12.7248410 18.1930190 0.0
14.3639010 16.3639610 0.0
18.1930190 13.7248410 0.0
10.9791958 10.6113497 0.0
18.9442719 9.0000000 0.0

*** GO---/ SP ( 0) 0.0 0.0 0.0 0.0

SPINDL/ OFF
COCLNT/ OFF
END
- FINI **
    
```

```

RECNO 1 ISN 1
RECNO 2 ISN 2
RECNO 3 ISN 3
RECNO 4 ISN 4
RECNO 5 ISN 5
RECNO 6 ISN 6
RECNO 7 ISN 7
RECNO 8 ISN 8
RECNO 9 ISN 9
RECNO 10 ISN 10
RECNO 11 ISN 11
RECNO 12 ISN 12
RECNO 13 ISN 13
RECNO 14 ISN 14
RECNO 15 ISN 15
RECNO 16 ISN 16
RECNO 17 ISN 17
RECNO 18 ISN 18
RECNO 19 ISN 19
RECNO 20 ISN 20
RECNO 21 ISN 21
RECNO 22 ISN 22
RECNO 23 ISN 23
RECNO 24 ISN 24
RECNO 25 ISN 25
RECNO 26 ISN 26
RECNO 27 ISN 27
RECNO 28 ISN 28
    
```



Rys. 10. Droga narzędzia skrawającego podczas obróbki części

Instrukcje o numerach 4 - 13 są, ogólnie biorąc, instrukcjami definiującymi; przy czym 4 - 11 określają kształt części, instrukcje o numerach 14, 15, 16, 25 i 26 - są instrukcjami postprocesora, zaś instrukcje o numerach 17 - 24 określają ruch narzędzia.

Wstawienie między instrukcje postprocesora całego zespołu instrukcji ruchu narzędzia było poddyktowane koniecznością wykonania przez układ sterowania obrabiarką wszystkich czynności opisanych instrukcjami od 14 do 26 właśnie w tej kolejności.

Na rys. 10 przedstawiono drogę narzędzia skrawającego podczas obróbki części, stanowiącą ilustrację do powyższego omówienia kolejnych instrukcji programu, zaś na rys. 11 pokazano wydruk zawartości zbioru CLDATA będący efektem wykonania instrukcji CLPRNT.

6. Uwagi końcowe

Język APT jest szeroko stosowany w USA, w mniejszym stopniu w Europie. Był on zaimplementowany na wielu dużych komputerach CDC, UNIVAC, IRIS, a nawet stał się de facto językiem standardowym takich firm, jak IBM. O jego popularności świadczy również fakt powstania bardzo dużej grupy języków wzorujących się na APT, charakteryzujących się jednak mniejszym stopniem złożoności. Nazywa się je APT-podobnymi. Zaliczamy do nich języki: ADAPT, IFAPT, NEL, MINIAPT, EXAPT i wiele innych.

Zo względu na istnienie bardzo dużej grupy języków do programowania OSN, podjęto próbę ich standaryzacji. Komitet TC97/SC9 International Standard Organization (ISO), zajmujący się tym problemem ustalił, że język standardowy powstanie na podstawie APT oraz języków APT-podobnych: IFAPT, EXAPT i NEL 2CL. A więc język APT z niewielkimi zmianami stanie się formalnie standardem ogólnosięwiatowym tak, jak już jest językiem standardowym w USA.

Planowany jest również dalszy rozwój APT idący w kierunku jak najlepszego sterowania OSN. Efektywność pracy programisty może być znacznie zwiększona przy bezpośredniej i graficznej komunikacji z systemem. Są więc planowane odpowiednie wersje APT wykorzystujące ułatwienia pracy programisty.

Podsumowując należy stwierdzić, że na obecnym etapie rozwoju system APT dzięki swojej wszechstronności i innym zaletom jest wykorzystywany szeroko, a ciągły rozwój, należy podziwiać się, że jeszcze zwiększy tę popularność.

Literatura

- [1] SHAH R.: Sterowanie numeryczne obrabiarek. Poradnik. Warszawa: WNT 1975
- [2] ZIĘTARSKI S.: Komputerowe systemy sterowania procesami obróbki skrawaniem (skrypt) WPT, Instytut Technologii Mechanicznej, Warszawa 1976
- [3] GUTÓWSKA H., BODIŃSKA I., KWAŚNIEWSKA G.: Opis języka APT - podręcznik programowania. Warszawa: IMM 1977 Archiwum Opracowań nr 18
- [4] System/360 APT Numerical Control Processor (360A-CN-10X) Systems Manual
- [5] GONTARCZYK T.: Systemy komputerowe automatyzacji projektowania technologii. Przegląd Mechaniczny 1976 nr 23 s. 811-812
- [6] DRAYTON C.E.: Automatically Programmed Tools. Leslie W.H.P. (ed): Numerical Control Programming Languages, North-Holland Publishing Company 1970 s. 424-433
- [7] NUSSEY I.D., PINTER H.R.: The IBM System/370 Numerical Control System. W: J. Hatvany (ed) - Computer Languages for Numerical Control. NII 1973 s. 339-347
- [8] SIMON N.: The Numerical Control of Machines Tools. London: Edward Arnold 1973, s. 432-438.

mgr Ewa SZCZEP
Instytut Maszyn Matematycznych

KOMPILATORY SKROŚNE

W grupie osób, biorących udział w przenoszeniu języka BCPL z maszyny IBM/370 na minikomputer MERA 400, posługiwaliśmy się terminem "kompilator skrośny", jako polskim odpowiednikiem angielskiego "cross compiler". Nie napotkawszy w literaturze lepszej propozycji na nazwanie tego typu programów, postanowiłam stosować w prezentowanym tekście ten termin istniejący już w moim słowniku informatycznym. Kompilatory skrośne można zaliczyć do grupy nowoczesnych narzędzi programowych, które zyskują coraz większą popularność wraz ze wzrostem znaczenia mini- i mikrokomputerów. W artykule jest mowa o potrzebie tworzenia, koncepcji i szerokiej użyteczności kompilatorów skrośnych.

1. Kłopoty użytkowników sprzętu mini- i mikrokomputerowego

Pojawienie się na świecie sprzętu mini-, a potem mikrokomputerowego, dużo tańszego, a więc przewyższającego duże maszyny cyfrowe pod względem ekonomicznym, spowodowało jeszcze większe zróźnicowanie w działającym sprzęcie liczącym. Ze względu na rozpoczęcie produkcji krajowej oraz stosunkowo niewielkie koszty minikomputerów sprzęt liczący stał się bardziej dostępny, choć jednocześnie pojawiło się wiele maszyn kompletnie nieoprogramowanych. Przed użytkownikami stanęło zadanie oprogramowania dostępnego sprzętu oraz zachowania, w miarę możliwości, dotychczasowego dorobku programowego. Nie ułatwia im tego zadania fakt, że mini- i mikrokomputery są zazwyczaj produkowane z myślą o szczególnym przeznaczeniu. Można tu wymienić takie klasy zastosowań:

- nauczanie,
- sterowanie procesem przemysłowym,
- projektowanie inżynierskie,
- problemy komunikacyjno,
- sieci minikomputerów wbudowane w większe systemy cyfrowe i przeznaczone do rozwiązywania tylko pewnej klasy problemów,
- programowanie stacje końcowe (inteligentne urządzenia końcowe) dużych maszyn,
- zastosowania biurowe,
- zbieranie i przechowywanie danych,
- nawigacja (ziemna, morska i powietrzna),
- grafika komputerowa.

* Wszystkie to dziedziny wymagają rozwiązywania skomplikowanych problemów, często trudniejszych niż te, które bywają rozwiązywane przez duże komputery z wykorzystaniem całego asortymentu narzędzi programistycznych na nich dostępnych. Mam tu na myśli istniejące na dużych maszynach cyfrowych, obok oprogramowania podstawowego (np. system operacyjny, assembler) często bardzo bogate oprogramowanie narzędziowe* (np. kompilatory, makrogeneratory, edytory) i użytkowe, wchodzące w skład dobrze wyposażonych bibliotek. Natomiast cechy charakterystyczne podstawowego zestawu mini- i mikrokomputerowego to: ograniczona pamięć operacyjna, krótkie słowo (często 8-bitowe), brak pamięci pomocniczych, ubogie środki programowe, dostarczone przez producenta, tzn. assembler (nie-raz prymitywny), z rzadka makroassembler, wyjątkowo kompilator jakiegoś języka wyższego poziomu (Fortran, Basic). Nawet jeśli taki kompilator należy do wyposażenia maszyny, to jest zazwyczaj bardzo wolny, nie generuje efektywnego kodu i w związku z tym jest kosztowny i kłopotliwy w użytku. Obniżka cen pamięci, wraz z możliwością dołączania jej blokami oraz starania producentów w kierunku wzbogacenia oprogramowania oferowanego sprzętu sprawiły, że takie mini, jak PDP, Burroughs czy NOVA dzisiaj w niczym nie ustępują dużym maszynom, a różnią się od nich jedynie krótszym słowem. Natomiast środki programowe, dostępne na minikomputerach mniej renomowanych firm,

* "...narzędzie charakteryzuje się tym, że jest wykorzystywane do tworzenia pewnego produktu, ale samo nie wchodzi do tego produktu jako jego część składowa; te jego cechy uprawniają nas do nazwania narzędziem translatora, a nie dają prawa do określenia tym terminem podprogramów bibliotecznych, które są włączane do naszego produktu". J. Dańda: Przemysłowa produkcja oprogramowania. Materiały na seminarium "Organizacja, metodyka i technika produkcji oprogramowania" Kolobrzeg 1975.

mikrokomputerach oraz na sprzęcie krajowym wymagają rozbudowy od użytkownika, który chce efektywnie wykorzystać sprzęt i własny czas. Proponowaną metodą wzbogacenia oprogramowania ubogo wyposażonych maszyn cyfrowych jest metoda kompilacji skróconej, o której będzie mowa w następnym rozdziale.

Spocyfika sprzętu mini przeznacza je bardziej do "końcowego" użytkownika, tzn. do wykorzystywania programów gotowych, wyprodukowanych dla tego sprzętu za pomocą dużej maszyny cyfrowej niż do produkowania oprogramowania, który to proces wymaga wspomaganie ze strony oprogramowania narzędziowego, a więc oznacza pracę na dużej maszynie cyfrowej. Widać więc, że pracując na mini- i mikrokomputerach nie można pomijać całkowicie maszyn liczących dysponujących dużą mocą obliczeniową oraz bogatym i różnorodnym oprogramowaniem. Ich potencjał służył od początku do przygotowywania programów dla małych maszyn. W szczególnej sytuacji otrzymany tą drogą program może wejść w skład oprogramowania minikomputera (np. odytor czy kompilator), dając użytkownikowi możliwość łatwego poprawiania programu czy też rozwiązywania problemów w języku wysokiego poziomu

2. Pojawienie się nowych narzędzi programowych i ich zalety

Analizując koszt uruchomienia na minikomputerze programu, rozwiązującego jakiś bardzo trudny problem i porównując go z kosztem uruchomienia tego programu na maszynie dużej, bogato oprogramowanej, dającej możliwość zapisania go w problemowo zorientowanym języku wyższego poziomu, stwierdzamy ogromny zysk w drugim wypadku. Zauważmy, że gdyby ten program można było przetłumaczyć na dużej maszynie - nie na jej język symboliczny, lecz na program w postaci dwójkowej lub w języku symbolicznym minikomputera, wówczas duża maszyna stałaby się "generatorem" programów dla maszyny małej. Programy te można by bowiem uruchomić na dużej maszynie, korzystając z dostępnych tam narzędzi, a wykonywać (w razie potrzeby - wielokrotnie) na małej maszynie.

Taka jest geneza powstania asamblerów skróconych i kompilatorów skróconych, dokonujących asambacji czy kompilacji na jednej maszynie - programów, które mają być potem wykonywane na drugiej, nie dysponującej środkami do uruchomienia tych programów.

Rację bytu narzędzi typu "cross" określa przede wszystkim ich ekonomiczność. Od ich istnienia uzależnione jest stworzenie ważnych - z punktu widzenia małych maszyn - możliwości:

- pisanie programów dla mini w języku problemowo zorientowanym, który ułatwia użytkownikowi zapisanie algorytmu rozwiązania zadania, co jest szczególnie ważne przy rozwiązywaniu bardziej skomplikowanych problemów;
- wykorzystania wszystkich cech i zasobów dużej maszyny, a szczególnie jej właściwości poprawiania błędów w programach (debugging);
- przeniesienia części programów z dużej maszyny na mini;
- projektowania sprzętu ściśle dopasowanego do określonej klasy zastosowań minikomputerów;
- równoległego rozwoju sprzętu i oprogramowania minikomputerów, gdyż istnienie symulatora sprawia, że produkcja oprogramowania nie jest uzależniona od dostępu do sprzętu.

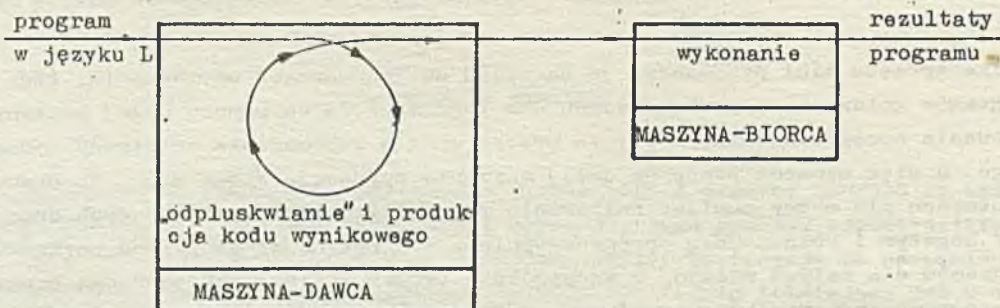
Niezaprzeczalnym zaletom metody "cross", którą można nazwać metodą nowoczesnej produkcji oprogramowania, można przeciwstawić jedną wadę: konieczność zapewnienia sobie dostępu do drugiej, dużej i dobrze oprogramowanej maszyny cyfrowej.

3. Istota metody "cross"

Pojęcie "cross" w odniesieniu do metody czy programów pochodzi od krzyżowania (mieszania) dwóch maszyn. Z reguły na jednej z nich - dobrze wyposażonej - produkowane jest oprogramowanie dla drugiej, dysponującej małą pamięcią i często jedynie podstawowym oprogramowaniem. Pierwszą z nich będziemy nazywali maszyną-dawcą (host computer lub cross computer), a drugą - maszyną-biorcą (aim computer lub target computer).

Takie podejście gwarantuje zapisywanie oprogramowania w języku wyższego poziomu, dobrze dostosowanym do pisania tego typu programów np. BCPL, Pascal, PL/1.

Jeśli w skład oprogramowania maszyny-dawcy wchodzi kompilator skrośny pewnego języka L dla maszyny-biorcy (najczęściej mini- lub mikrokomputera) to dysponując obiema maszynami można uruchomić dowolny program, napisany w języku L, w sposób zilustrowany na rys. 1.



Rys. 1. Koncepcja kompilacji skrośnej

Jak łatwo zauważyć maszyna-dawca może być skrzyżowana metodą "cross" z wieloma maszynami-biorcami, służąc im swoimi zasobami. Zakładając na niej produkcję oprogramowania dla wielu minikomputerów zbliżamy się do koncepcji fabryki oprogramowania, wysuniętej przez Rossa [1].

4. Klasyfikacja programów typu "cross"

Jak już wspomniano, narzędzie typu "cross" są to programy, dostępne na maszynie-dawcy, a pracujące na rzecz innej maszyny (biorcy), na której wykonywany jest wyprodukowany przez nie (często w postaci dwójkowej) kod wynikowy. Na przykład asembler dla mikrokomputera Signetic's 2650, który jest napisany w języku Assembler maszyny IBM/370 i na niej dostępny, nazwiemy asemblerem skrośnym. Programy asemblowane na IBM/370 otrzymuje się w postaci dwójkowej i w tej postaci wykonuje na Signetic's.

Wśród narzędzi typu "cross" można wyróżnić następujące klasy:

- translatory skrośne: kompilatory skrośne i asemblery skrośne
- ładowacze skrośne
- symulatory

Kompilatory skrośne i asemblery skrośne tłumaczą program źródłowy napisany odpowiednio w języku wysokiego poziomu lub symbolicznym (maszynowym) na język wewnętrzny (binarny) maszyny-biorcy. Przeprowadzają także diagnostykę i drukują listę błędów. Podobnie jak translatory, produkujące kod wynikowy dla maszyny-dawcy, translatory skrośne drukują treść programu źródłowego (listing) i opcjonalnie inne, interesujące użytkownika, wiadomości (np. listę odwołań krzyżowych). Otrzymane na wyjściu z translatorów skrośnych programy mogą być poddane na tej samej maszynie-dawcy operacjom przemieszczenia (linkowania), dokonywanym przez ładowacze skrośne. Operacje te polegają na przekształceniu adresów z względnych na bezwzględne i połączeniu w jedną całość tych programów, które się do siebie odwołują. Ładowacze skrośne modyfikują programy na podstawie informacji, które otrzymują zarówno od translatora skrośnego jak i bezpośrednio od programisty.

Gdy maszyna-biorca nie jest dostępna, a maszyna-dawca wyposażona jest w symulator tej pierwszej, wówczas działanie programu wynikowego można sprawdzić za pomocą tego symulatora. Symulator ma wbudowaną strukturę logiczną maszyny docelowej i symuluje wykonywanie się na niej zadanego mu programu. W trakcie symulacji można dodatkowo wykonywać następujące czynności:

- obserwowanie pamięci i rejestrów symulowanej maszyny;
- zatrzymanie wykonywania programu w momencie osiągnięcia określonego adresu lub w momencie zapisu/odczytu informacji do/z zadanego miejsca pamięci;
- śledzenie drogi wykonywania programu i drukowanie pamięci symulowanej maszyny w postaci np. oktalnej, dziesiętnej czy kodu numerycznego.

Symulator dostarcza również pewnych informacji czasowych.

5. Oprogramowanie wymagane dla maszyny-dawcy

W skład oprogramowania każdej maszyny wchodzi programy, umożliwiające na niej pracę i ułatwiające ją; należą do nich także programy, tłumaczące na kod wynikowy, wykonywany na tej maszynie. Wśród tego typu programów można wyróżnić pewne klasy, a mianowicie:

- translatory: kompilatory i asemblory,
- edytory,
- ładowacze,
- programy "odpluskwiające"

Dwie pierwsze klasy różnią się od analogicznych klas typu "cross" jedynie tym, że przekształcają programy, przeznaczone do wykonywania na tej samej maszynie, na której są dostępne (tzn. maszyna-biorca = maszyna-dawca). Edytory umożliwiają wprowadzenie poprawek i modyfikacji w tokacie programu źródłowego.

Zestaw operacji, którymi dysponuje edytor, zawarty jest w liście jego poleceń i stanowi o jego sile dokonywania zmian w tekstach programów. Podczas gdy jedne edytory operują tylko na pełnych liniach tekstu, inne dają możliwość dodania, wyrzucenia lub wymiany dowolnego ciągu znaków. Edytor na maszynie-dawcy może również służyć do modyfikacji tekstów programów, które mają być przetwarzane przez translatory skróśne, a więc wspomaga oprogramowanie na rzecz maszyny-biorcy. Programy odpluskwiające ułatwiają testowanie programu wynikowego. Zakres ich funkcji obejmuje:

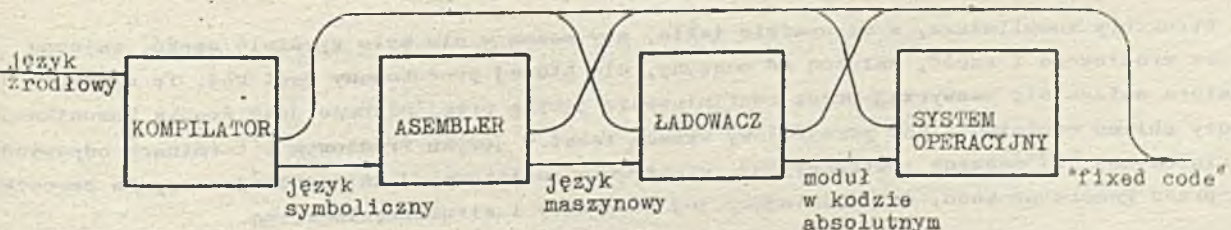
- drukowanie zawartości rejestrów oraz pamięci (w mikrokomputerze: zarówno ROM jak i RAM),
- rozpoczynanie wykonywania programu od zadanego miejsca pamięci,
- zatrzymanie wykonywania programu w momencie napotkania instrukcji z określonego miejsca pamięci lub gdy spełniony jest zadany warunek.

Narzędzia typu "cross" są zazwyczaj produkowane dla maszyny niedostatecznie oprogramowanej lub kompletnie nieoprogramowanej. Natomiast na maszynę-dawcę wybiera się dużą maszynę, bogato wyposażoną w programy, o których była mowa powyżej. Przy wyborze metody "cross", dla wzbogacenia oprogramowania maszyny-biorcy trzeba zdawać sobie sprawę z następujących faktów:

- programy skróśne, należące do maszyny-dawcy, zazwyczaj silniejszej i szybszej niż maszyna-biorca, mogą być efektywniejsze niż analogiczne programy uruchamiane na maszynie-biorcy;
- koszt uruchomienia programu skróśnego jest mniejszy niż analogicznego programu na maszynie-biorcy;
- metoda "cross" jest wygodniejsza ze względu na możliwość zapisania oprogramowania w problemowo zorientowanym języku wyższego poziomu oraz możliwość korzystania z wszystkich zasobów maszyny-dawcy;
- metoda "cross" wymaga znajomości dwóch systemów komputerowych: maszyny-dawcy i maszyny-biorcy.

6. System programów skróśnych

W zależności od stanu oprogramowania maszyny-biorcy, zasobów dostępnej maszyny-dawcy oraz celu, który przyświeca użytkownikowi, może on dla wybranego języka źródłowego stworzyć ciąg programów, którego elementy są albo programami typu "cross", albo programami wchodzącymi w skład oprogramowania maszyny-biorcy. Różnorodność dróg, którymi można przejść z programem w języku źródłowym, choć go wykonać na maszynie-biorcy przedstawia rys. 2.



Rys. 2. Różnorodność sposobów tworzenia łańcucha programów

Kod wykładowy każdego etapu przetwarzania powinien dawać możliwość zmiany maszyny i przejścia z maszyny-dawcy na biorcę. W szczególnej sytuacji, jeśli program dla maszyny-biorcy jest wykonywany na maszynie-dawcy ciąg (łańcuch) programów ma postać: kompilator skrośny - assembler skrośny - ładowacz skrośny - symulator biorcy. Często zdarza się, zwłaszcza gdy maszyna-biorca jest mikrokomputerem zaopatrzonym jedynie w system operacyjny, że translator języka symbolicznego tej maszyny realizuje się na innej maszynie metodą "cross". Chcąc uruchomić na mikrokomputerze programy użytkowe, które umożliwiałyby na nim wygodniejszą pracę, tworzy się na większej, bogatszej w zasoby maszynie tzw. system programów skrośnych (SPS). Za jego pomocą można uruchamiać na maszynie-dawcy dowolne programy, napisane w języku symbolicznym mikrokomputera, a potem przenosić je (już w postaci przyswajalnej) na mikrokomputer. W skład SPS wchodzi:

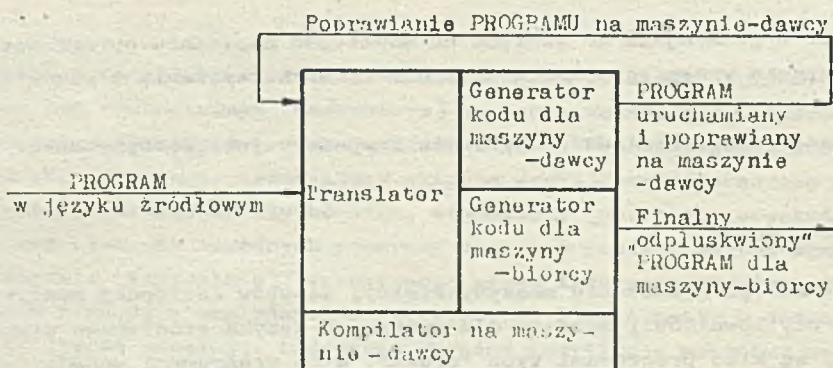
- assembler skrośny,
- ładowacz skrośny,
- ładowacz wstępny,
- symulator.

System ten powinien dawać możliwość łączenia i nakładania na siebie programów, możliwość korzystania z bibliotek obu maszyn oraz powinien mieć aparat, ułatwiający zarządzanie danymi.

Ładowacz wstępny umożliwia łączenie w jedną całość programów oddzielnie asemblowanych i umieszczonych np. na dysku. Następnie przetwarza kod wynikowy, otrzymany na etapie łączenia, na kod absolutny w postaci możliwej do wprowadzenia i wykonania na maszynie-biorcy. Gdy nie jest ona dostępna, przetwarzany program jest przekazywany ładowaczowi skrośnemu, który dokonuje ostatecznego przedadresowania programu i załadowania go do pamięci symulowanej maszyny, wydzielonej w pamięci operacyjnej maszyny-dawcy. Programy przetworzone w opisany wyżej sposób przez SPS wzbogacają oprogramowanie narzędziowe i użytkowe maszyny-biorcy.

7. Kompilacja skrośna a mobilność

Zdarza się, że istniejący na maszynie-dawcy kompilator języka źródłowego daje - przez wymianę generatora kodu - możliwość kompilacji skrośnej na wiele maszyn-biorców. Wówczas wszystkie związane z tą maszyną i nowo powstające programy w języku źródłowym mogą być przeniesione i użytkowane na wielu maszynach docelowych (rys. 3). Sytuację taką urzeczywistnia odpowiednio zaprojektowa-



Rys. 3. Przenoszenie programów

nie struktury kompilatora, a mianowicie takie, aby można w nim było wyróżnić część, zależną od języka źródłowego i część, zależną od maszyny, dla której produkowany jest kod. Tę własność kompilatora osiąga się zazwyczaj przez zdefiniowanie języka przejściowego jako środka komunikacji między oboma częściami. Kod przejściowy wyraża tekst w języku źródłowym w terminach odpowiednio zdefiniowanej [2] maszyny abstrakcyjnej, efektywnie realizowanej na dowolnej maszynie rzeczywistej przez generator kodu, uwzględniający jej parametry i strukturę logiczną.

Jeśli język źródłowy jest ukierunkowany na pisanie oprogramowania uzyskujemy możliwość produkowania oprogramowania przenaszalnego. W szczególności jeśli kompilator jest sam napisany w języ-

ku źródłowym można pokusić się o jego przeniesienie na maszynę docelową, która spełnia odpowiednie warunki (przede wszystkim ma dużą pamięć).

Programy mogą być przenoszone w postaci dwójkowej, gdy zakładamy wymianę generatora kodu na maszynie-dawcy lub w języku przejściowym, jeśli uwzględnimy możliwość zrealizowania generatora kodu na samej maszynie-biorcy. Przykładem takiego postępowania jest przeprowadzona w IMM operacja udostępnienia BCPL-u na komputerze MERA 400. Kompilator tego języka na maszynie IBM/370 daje możliwość wyprowadzania programów w języku pośrednim OCODE, niezależnym od tej maszyny a zdefiniowanym specjalnie dla BCPL-u. Zainstalowanie na maszynie MERA 400 interpretera języka pośredniego OCODE, napisanego w języku symbolicznym tego minikomputera, umożliwi (przy współpracy z IBM/370) uruchamianie na nim programów w BCPL-u, mimo że translator tego języka nie jest na nim dostępny. Dlatego też zestaw obu programów: kompilatora BCPL-u z możliwością wyjścia w OCODE na IBM/370 oraz interpretera języka pośredniego OCODE na maszynie MERA 400 nazwalismy kompilatorem skrótnym języka BCPL na minikomputer MERA 400. Za jego pomocą wzbogacono oprogramowanie maszyny MERA 400 o edytor EDYTA, napisany w BCPL-u, uruchomiony i sprawdzony na maszynie IBM a następnie wyprowadzony w języku OCODE i w tej postaci użytkowany na minikomputerze.

Myślę, że metoda "cross", która - obok wielu innych zalet - umożliwia produkowanie oprogramowania przenaszalnego, będzie zyskiwać coraz więcej zwolenników.

Literatura

- [1] ROSS D.T.: Software development for minicomputers. Minicomputers, Infotech State of the Art Report, s.205.
- [2] POOLE P.C., WAITE W.M.: Portability and adaptability. Advanced Course on Software Engineering. Springer Verlag 1973.
- [3] WATSON I.M.: Comparison of commercially available software tools for microprocessor programming. Proceedings of the IEEE 1976, vol.64 nr 6.
- [4] LASOTA M.: Przenośność oprogramowania (przegląd metod). Informatyka 1976 nr 12.
- [5] RICHARDS M.: The portability of the BCPL compiler. Software, Practice and Experience 1971 vol.1 nr 2.

dr inż. Tadeusz SINKIEWICZ
Instytut Maszyn Matematycznych

ZASTOSOWANIE MIKROPROCESORÓW W MONITORACH EKRAŃOWYCH

1. WSTĘP

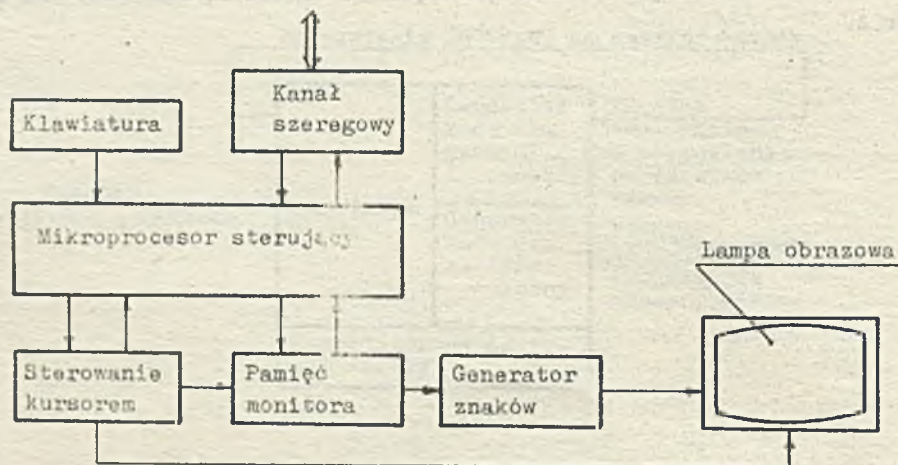
W ciągu ostatnich lat obserwujemy wyraźną tendencję producentów sprzętu elektronicznego do zwiększania możliwości funkcjonalnych zarówno monitorów alfanumerycznych jak i graficznych, przy czym zmiany wprowadzane przez konstruktorów mają na celu maksymalne dostosowanie własności użytkowych urządzenia do wymagań użytkownika. Osiągnięcie tego celu wymaga z reguły stosowania złożonych układów sterowania, których realizacja za pomocą układów o małym stopniu scalenia jest bardzo pracochłonna, a uzyskiwane konstrukcje często nie znajdują uznania odbiorców ze względu na swoje gabaryty i parametry eksploatacyjne.

Z podanych względów konstruktorzy monitorów ekranowych coraz częściej stosują podzespoły elektroniczne wykonane technologią wielkiej skali integracji. Wśród tych podzespołów poczesne miejsce zajmują mikroprocesory, które z powodzeniem są wykorzystywane do realizacji złożonych funkcji sterowania; ich parametry techniczne pozwalają konstruować urządzenia o niewielkich wymiarach, małym poborze mocy i konkurencyjnej cenie.

2. MONITORY ALFANUMERYCZNE

Struktura blokowa i funkcje mikroprocesora

Strukturę blokową typowego monitora alfanumerycznego ze sterowaniem mikroprocesorowym pokazano na rys. 1. Dane wprowadzane są do monitora ekranowego za pomocą klawiatury lub za pośrednictwem kanału szeregowego łączącego monitor z komputerem. Położenie kursora na ekranie zmienia się zgodnie z sygnałami sterującymi dostarczonymi z klawiatury lub z komputera. Układ generatora znaków pobiera dane z pamięci i przetwarza je na punktowe obrazy znaków wyświetlane na ekranie lampy oscyloskopowej.



Rys. 1. Schemat blokowy monitora alfanumerycznego ze sterowaniem mikroprocesorowym

Mikroprocesor łącznie z programowaną pamięcią i układami interfejsu wykonuje podstawowe czynności sterowania, mianowicie steruje przesyłaniem danych między klawiaturą, kanałem szeregowym i pamięcią monitora. Mikroprocesor wydziela i interpretuje kody w miarę ich otrzymywania, przesyła znaki do pamięci i realizuje wymagane funkcje sterowania. Przeprowadza on również manipulacje z danymi znajdującymi się w pamięci monitora i formuje meldunki wysyłane do komputera.

Mikroprocesor może bezpośrednio sterować pracą linii telekomunikacyjnych dołączonych do monitora alfanumerycznego, a w wypadku odpowiednio zaprojektowanego interfejsu można go zaprogramować tak, aby realizował szeregową współpracę (daisy chain) z innymi urządzeniami końcowymi korzystając z danych wysyłanych z komputera, co pozwala zwiększyć efektywność systemu i zmniejszyć jego koszty. Jako ogólną zasadę dotychczas przyjmowano, że mikroprocesor powinien sterować w jak największym stopniu przesyłaniem danych i sygnałami logicznymi, lecz synchronizacja czasowa (np. opóźnienie między sygnałami kasowania i wprowadzania) powinna być raczej realizowana za pomocą sprzętu, który może być dostosowany do wymagań każdego rozwiązania. Wraz ze wzrostem szybkości pracy mikroprocesorów należy sądzić, że sygnały synchronizacji będzie można również generować w mikroprocesorze.

Jedną z najbardziej złożonych funkcji realizowanych w doskonalszych urządzeniach końcowych z monitorami alfanumerycznymi jest "kasowanie wiersza" (delete line). Do wykonania tej funkcji wykorzystuje się licznik, za pomocą którego można opuszczać pełną liczbę znaków linii. Funkcja ta może być łatwo realizowana za pomocą wewnętrznych rejestrów mikroprocesora. Poza funkcjami o charakterze repetycyjnym, mikroprocesor może realizować funkcje zależne od danych, na przykład może być wykonana funkcja "opuszczenie zdania", polegająca na opuszczeniu wszystkich znaków między dowolnymi dwoma ogranicznikami tekstu. Za pomocą mikroprocesora można realizować złożone algorytmy sterowania słowami tekstu, formowania kolumn i specjalnego traktowania różnych znaków występujących w tekście.

Mikroprocesor może przetwarzać informacje znajdujące się w pamięci monitora, na przykład wyszukiwać dane, co pozwala pobrać z pamięci te słowa, które operator chce sprawdzić lub skorygować. Mikroprocesor ułatwia dostęp do komputera głównego i do pamięci monitora, co zwiększa efektywność wymiany informacji w systemie.

Projektant monitora alfanumerycznego, stosując mikroprocesor, może w prosty sposób zmieniać repertuar funkcji realizowanych w urządzeniu końcowym. Funkcje urządzenia końcowego mogą być dostosowane do wymagań użytkownika przez wpisanie odpowiedniego programu do pamięci stałej (ROM) sterującej pracą mikroprocesora, np. klawiatura z przyciskami oznakowanymi tak jak klawiatura dalekopisu może być przekształcona na klawiaturę maszyny do pisania przez zmianę oznakowania kilku klawiszy i translację kodów uzyskiwanych z tych klawiszy, co można łatwo przeprowadzić za pomocą mikroprocesora.

W podobny sposób można zmieniać na klawiaturze położenie klawiszy funkcyjnych przesuując odpowiednie adresy w tablicy mikroinstrukcji odpowiadających realizacji funkcji. Bez trudu mogą być wykonane modyfikacje funkcji. Możliwość nadawania wyspecjalizowanej interpretacji, zarówno kluczom funkcyjnym jak i wyprowadzonym na ekran rysunkom wywołanym z klawiatury, ma zastosowanie również do informacji otrzymywanych z komputera. Zatem kody sterujące, które np. powodują pulsowanie wydzielonych części ekranu lub wyświetlanie czarnych znaków na białym tle, w odróżnieniu od normalnie wyświetlanych białych znaków na ciemnym tle, mogą być przesyłane za pomocą linii telekomunikacyjnych w postaci podwójnie zakodowanych sekwencji i przechowywane w pamięci monitora w postaci pojedynczego kodu. Ciągi kodów mogą być również wykorzystane do sterowania, ustawiania położenia kursora, marginesów lub zakresu tabulacji, przy czym mikroprocesor może przeprowadzać translację danych numerycznych z postaci binarnej do dziesiętnej lub dowolnej innej.

● Kryteria wyboru mikroprocesora

Wybór szybkości mikroprocesora przeznaczonego do stosowania w monitorach alfanumerycznych zależy w głównej mierze od budowy interfejsu znajdującego się między mikroprocesorem i pamięcią monitora oraz od wymagań wynikających z zastosowań monitora.

W sytuacji, gdy mikroprocesor może identyfikować i przetwarzać w przybliżeniu 1200 kodów/s (np. INTEL 8008-1) nie są miarodajne konwencjonalne metody oceny prędkości procesora, na przykład takie, jak czas dodawania dwu liczb. Jeżeli w monitorze alfanumerycznym stosowane są rejestry przesuwne, zdolność przetwarzania mikroprocesora jest zazwyczaj ograniczona przez czas wymagany do odczytywania znaków w miarę przesuwania się kursora w pamięci. Jeżeli w monitorze jest

stosowana pamięć o dostępie swobodnym (RAM), czynnikiem ograniczającym są czynności szeregowego dostępu, takie jak selektywne wymazywanie danych (opatrzonej wskaźnikami), chyba że przewidziano sprzętowe środki realizacji adresowania lub zastosowano rozwiązania pozwalające uniknąć ograniczeń mikroprocesora*).

Przy projektowaniu urządzenia końcowego należy pamiętać o tym, że zastosowanie odpowiednich rozwiązań sprzętowych pozwala prawie zawsze uzyskać taką strukturę monitora, w której mikroprocesor nie będzie ograniczał szybkości działania urządzenia końcowego. Jednak w sytuacji braku dodatkowego sprzętu należy liczyć się z tym, że czas realizacji mikroinstrukcji kolejnego czytania-wymiany-pisania i indeksowania w monitorze z mikroprocesorem trwa około 1 μ s.

Monitory alfanumeryczne konkurują ze sobą pod względem ceny. Wykorzystanie pamięci stałej (ROM) do pamiętania programu powoduje zwiększenie kosztów zastosowanego mikrokomputera, przy czym koszt pamięci stałych wymagających wykonania specjalnych masek jest znacznie wyższy. Koszty te wpływają również na wynikową cenę sprzedaży, gdyż konieczna jest optymalizacja listy instrukcji pozwalająca zminimalizować wykorzystanie pamięci. Wymagana tu jest dobra znajomość programowania mikrokomputerów, np. instrukcje Exclusive-OR mikroprocesora INTEL 8008-1 są bardzo użyteczne do odszukiwania znaków kontrolnych, a wskaźnik (flaga) parzystości jest często wykorzystywany w większości podprogramów krytycznych pod względem prędkości.

Innymi użytecznymi operacjami są operacje logiczne i arytmetyczne z kolejnymi bajtami informacji, szczególnie instrukcja "compare-immediate". Ze względu na to, że wszystkie podprogramy sterujące pracą mikroprocesora, związane z zastosowaniami monitorów alfanumerycznych są proste i krótkie, instrukcje rozgałęzień warunkowych i wywoływania podprogramów są niewygodne do stosowania, a bardziej użyteczne są instrukcje względnego adresowania. Z praktyki wynika, że wykorzystywana jest zagnieżdżanie podprogramów do czterech lub co najwyżej do pięciu poziomów oraz, że bardzo użyteczny jest programowy licznik stosu typu LIFO.

Przerwania nie są w rozważanych zastosowaniach wymagane, gdyż zwykle urządzenie końcowe wykonuje w danej chwili tylko jedną czynność. Możliwość obsługi przerwań jest jednak istotna w systemach z głosowaniem (polling systems), w których dane wysyłane są jednocześnie przez operatora i komputer centralny. Wymaganie to pojawia się również wówczas, gdy urządzenie końcowe może równocześnie wysyłać i przyjmować dane lub przysyłać dane z procesora centralnego do podporządkowanej drukarki w czasie normalnego posługiwania się klawiaturą. Tak więc na wybór mikroprocesora wpływa zastosowanie urządzenia końcowego i konfiguracja jego sprzętu.

● Realizacje praktyczne

Prace nad zastosowaniem mikroprocesorów w alfanumerycznych monitorach ekranowych podjęte były bezpośrednio po pojawieniu się mikroprocesorów scalonych na jednej kostce wielkiej skali integracji, to jest w 1971 r. W pracy [2] opisano kolejne etapy konstruowania monitora alfanumerycznego w Beehive Medical Electronics z zastosowaniem mikroprocesora INTEL 8008, a następnie INTEL 8008-1. W początkowym okresie istotną przeszkodą wykorzystania mikroprocesorów był ich znaczny koszt, który obecnie uległ znacznemu obniżeniu**.)

Autorzy podkreślają, że mikroprocesor pozwolił skonstruować układ samodiagnostyczny z kompletem funkcji edytorskich i własnościami ułatwiającymi tworzenie struktur satelitarnych. Dane mogą być przesyłane w postaci pojedynczych znaków, linii lub stronic. Do monitora mogą być dołączane dodatkowe urządzenia, pamięci lub interfejsy.

W pracy [2] podkreśla się, że zastosowanie mikroprocesora pozwoliło uzyskać wysokie parametry użytkowe monitora ekranowego przy niskiej cenie, znaczne uproszczenie obsługi i łatwość modyfikowania funkcji monitora, osiągnięta dzięki możliwości szybkiego przeprogramowania (wymiany pamięci stałej).

*) Wiąże się to z tym, że urządzenia końcowe z monitorami ekranowymi sprzedawane są zazwyczaj jako urządzenia końcowe z pamięcią realizowaną na rejestrach przesuwnych, a wprawny użytkownik szybko wykonuje kolejne czynności.

***) Koszt mikroprocesora INTEL 8080 w sprzedaży hurtowej wynosi obecnie 10 \$ za sztukę.

W 1975 r. firma Hewlett Packard zastosowała mikroprocesor w seryjnym monitorze alfanumerycznym model 2640A [4]. Monitor ten odznacza się rozbudowanymi możliwościami funkcjonalnymi przy stosunkowo niskiej cenie. Mikroprocesory umożliwiły uzyskanie elastyczności funkcjonalnej pozwalającej dostosowywać urządzenie końcowe do nowych zastosowań i dopasowywać je do zmieniających się wymagań użytkownika. Model 2640A jest pierwszym z planowanej rodziny monitorów alfanumerycznych o podanych własnościach.

Monitor 2640A może przeprowadzać samodiagnozę i sygnalizować sytuacje, w których wymagana jest pomoc serwisowa. Testy umożliwiają obsłudze szybkie zlokalizowanie uszkodzonego modułu.

Programy zapewniające wyżej podane cechy monitora ekranowego są przechowywane w pamięci stałej (ROM) o pojemności 8 k bajtów.

Konstruktorzy zastosowali mikroprocesor dostępny w handlu uznając, że wykorzystanie go będzie znacznie tańsze i mniej czasochłonne od skonstruowania własnego wyspecjalizowanego układu wielkiej skali integracji, ewentualnie od zastosowania standardowych układów logicznych o znacznej objętości i wymagających ponadto znacznie więcej miejsca.

Mikroprocesor ma 48 instrukcji jedno- dwu- lub trójbajtowych, akumulator, zespół rejestrów, licznik programu, stosową pamięć adresów dla siedmiu podprogramów i możliwość adresowania 16 384 miejsc pamięci. Z mikroprocesorem współdziałają 8-bitowe standardowe układy logiczne. Szyna wejścia/wyjścia jest wykorzystywana do przesyłania danych i adresów pamięci.

Do generacji znaków zastosowano w tym modelu matrycę punktową o wysokiej rozróżnialności. Na ekranie można uzyskać 24 linie 80-znakowe. Oprócz standardowych znaków alfabetu łacińskiego mogą być stosowane symbole matematyczne, symbole alfabetu greckiego, planuje się ponadto stosowanie symboli pisma japońskiego. Można uzyskać obraz negatywowy, we wszystkich kombinacjach można łączyć pulsowanie, zmniejszanie jaskrawości i podkreślanie znaków. Miejsca pamięci przydzielane są w sposób dynamiczny. W urządzeniu można jednocześnie pamiętać do 400 linii. Pamięć może być powiększona za pomocą modułów 2 k lub 4 k. Użytkownik może sprawdzić i poprawić dane przed rozpoczęciem transmisji tych danych do komputera.

● Autonomiczny procesor tekstowy

Kolejny, nowy jakościowo etap rozszerzenia możliwości funkcjonalnych monitorów alfanumerycznych stanowi opracowanie tak zwanego autonomicznego procesora tekstowego. W skład systemu tworzącego autonomiczny procesor tekstowy wchodzi monitor alfanumeryczny, pamięć masowa (np. na elastycznym dysku lub taśmie magnetycznej), drukarka wysokiej jakości oraz procesor sterujący tymi urządzeniami.

System tego typu wykonuje bardzo zróżnicowane operacje redakcyjne i edytorskie: manipulacje symbolami, słowami, wierszami, akapitami, stronicami, wprowadzanie poprawek, drukowanie tekstów itd. Przetworzony i skorygowany tekst może być przesłany do pamięci masowej lub wydrukowany za pomocą drukarki.

Do sterowania pracą urządzeń wchodzących w skład systemu procesora tekstowego można z powodzeniem zastosować mikroprocesor. Parametry funkcjonalne i szybkość mikroprocesora muszą umożliwić wykonanie wszystkich wymienionych operacji w czasie rzeczywistym. Mikroprocesor pracujący w rozbudowanym systemie musi sterować przesyłaniem danych między pamięcią i dyskiem z prędkością 250 Kbitów/s oraz odznaczać się dużą szybkością wykonywania operacji redagowania tekstu.

W autonomicznym procesorze tekstowym opisanym w [5] zastosowano specjalnie skonstruowany mikroprocesor AES - SOC charakteryzujący się:

- krótkim czasem wykonania instrukcji (około 240 ns),
- 16-poziomym stosem pozwalającym korzystać z podprogramów wewnątrz instrukcji,
- zdolnością bezpośredniego przetwarzania danych uzyskiwanych z zewnątrz za pomocą urządzeń wejścia/wyjścia,
- niezależną adresacją instrukcji i danych, co pozwala pobierać instrukcje z pamięci stałej jednocześnie z wykonaniem instrukcji pobierania danych.

Mikroprocesor ma bogaty repertuar instrukcji (432 instrukcje) - co powoduje, że jest wygodnym i elastycznym narzędziem przetwarzania informacji tekstowej. Mikroprocesor AES-80C nie wymaga stosowania specjalnych układów sterowania urządzeniami wejścia/wyjścia, niezbędnych dla większości uniwersalnych mikroprocesorów dostępnych na rynku.

3. MONITORY GRAFICZNE

● Struktura systemu graficznego

W uniwersalnych systemach graficznych stosowane są złożone przekształcenia informacji określającej strukturę obrazu. Można wymienić tu takie operacje, jak selekcyjne wycieranie fragmentów obrazu, "okienkowanie" (windowing) lub "skalowanie" (scaling).

W złożonych systemach graficznych niemal powszechnie stosuje się podział systemu na wyspecjalizowane części o zróżnicowanych funkcjach. Względna ważność i moc obliczeniowa każdej części zależy od przeznaczenia systemu. Do bardziej złożonych funkcji graficznych stosuje się realizacje sprzętowe, w celu zwiększenia szybkości działania systemu. Każdy z takich procesorów obrazowych ma bezpośredni dostęp do danych przechowywanych w pamięci głównej maszyny cyfrowej i określa dane niezbędne dla generatora obrazu. Procesor główny w ten sposób jest odciążony od wykonywania wielu czasochłonnych obliczeń i może być całkowicie wykorzystany do interakcyjnych zmian struktury danych.

Jako wyspecjalizowane procesory systemu mogą być zastosowane mikroprocesory, które można programować dla wymaganych w systemie funkcji i które obniżają koszt oraz umożliwiają wyeliminowanie wielu etapów koniecznych przy stosowaniu tradycyjnych metod konstruowania urządzeń cyfrowych.

● Funkcje procesora obrazowego

Obraz wirtualny przechowywany w pamięci procesora głównego jest macierzą punktów określonych przez wartości współrzędnych miejsc pamięci. Rozmiar macierzy zależy od długości słowa, np. słowo szesnastobitowe daje macierz 64K x 64K. Wszystkie segmenty liniowe rysunku kreślone na obrazie wirtualnym muszą rozpoczynać się i kończyć w punktach tej macierzy. Obraz wyświetlany na ekranie odzwierciedlany jest macierzą mniejszego rozmiaru; typowa jest macierz o wymiarach 1024 x 1024.

Jak już wspomniano, w rozbudowanych systemach graficznych duża moc obliczeniowa wymagana jest do realizacji takich funkcji, jak selekcyjne wycieranie fragmentów obrazu, okienkowanie i skalowanie.

Typową funkcją rozbudowanego systemu graficznego, wymagającą dużej mocy obliczeniowej jest funkcja selekcyjnego wycierania, szczególnie gdy w systemie stosowana jest lampa obrazowa typu telewizyjnego, wymagająca periodycznego odświeżania obrazu. Funkcję tę mogą wykonywać odpowiednio zaprogramowane mikroprocesory, przy czym dla zwiększenia prędkości systemu niektóre operacje można realizować sprzętowo.

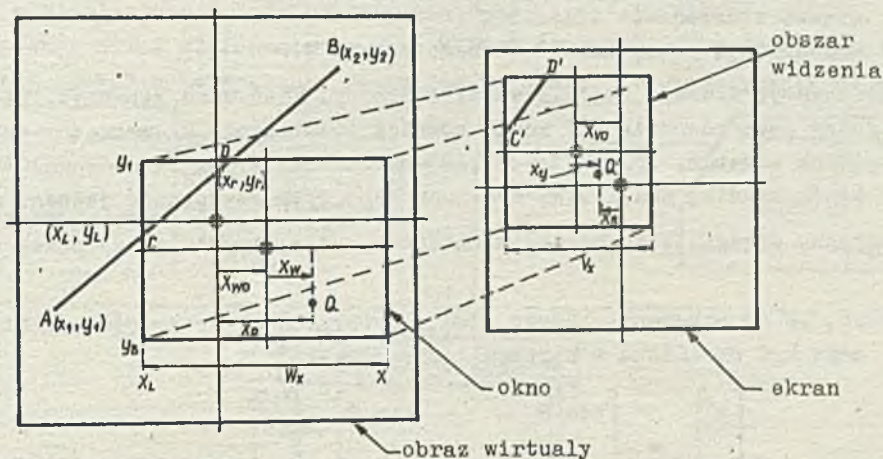
Podobny poziom złożoności cechuje funkcje okienkowania i skalowania. Dla lepszego wyjaśnienia przeznaczenia elementów struktury systemu realizujących te funkcje niezbędne jest omówienie odpowiednich algorytmów (zamieszczonych w punkcie "Okienkowanie" i "Skalowanie").

Wymienione funkcje spełniane są wówczas, gdy procesor obrazowy transformuje współrzędne punktów wybranego obszaru obrazu wirtualnego - okna, w niezależnie wybrany obszar ekranu, tak zwany obszar widzenia (viewport). Transformacja ta wykonywana jest w dwóch etapach.

- Okienkowanie lub obcinanie (clipping), ustala które linie leżą całkowicie lub częściowo w oknie. Na rys. 2 linia AB leży częściowo w oknie, i w procesie okienkowania należy określić współrzędne punktów C i D, to jest granice "widzialnej" części AB.
- Skalowanie polega na transformacji współrzędnych punktów (CD) obrazu wirtualnego i innych części rysunku w oknie na odpowiednie współrzędne ekranu w obszarze widzenia.

Procesor obrazowy wykonuje ponadto pomocnicze operacje na współrzędnych obrazu wirtualnego: skalowanie wstępne (prescaling) i obrót (rotation). Skalowanie wstępne obejmuje mnożenie zbioru

wartości współrzędnych przez jednakowy współczynnik, w jakim pewna część rysunku na obrazie wirtualnym jest zwiększana lub zmniejszana. Dla wykonania obrotu zbiór punktów obracany jest dokoła określonego środka o określony kąt. Przetransformowane współrzędne obrazu wirtualnego można następnie okienkować i skalować.



Rys. 2. Schematyczna ilustracja transformacji graficznych

W prowadzonych dalej rozważaniach pominięte będą występujące w algorytmach operacje dodawania, ponieważ wykonywane są one w znacznie krótszym czasie niż mnożenie lub dzielenie i nie wpływają w istotny sposób na ogólny czas przetwarzania.

● Okienkowanie

W celu określenia, gdzie leży wybrany segment linii, badana jest odległość końców każdego segmentu linii obrazu wirtualnego od krawędzi okna. Linie leżące całkowicie poza oknem są odrzuca-
no, leżące całkowicie wewnątrz - podlegają skalowaniu, a te, które przecinają się z krawędzią-
mi okna są obcinane w celu znalezienia punktów przecięcia.

Program obcinania rozwiązuje równanie, które można otrzymać na podstawie rys. 2.

$$y_L = (x_L - x_1) \frac{(y_2 - y_1)}{(x_2 - x_1)} + y_1 \quad (1)$$

Inna metoda polega na zastosowaniu algorytmu "punktu środkowego", w którym linia dzielona jest stopniowo, w krokach binarnych do chwili, w której punkt środkowy pokryje się z krawędzią okna. Programowe wykonanie tego algorytmu byłoby wolniejsze niż rozwiązanie bezpośrednie, lecz wykorzystując tylko operacje przesuwania i dodawania, możliwa jest szybka realizacja sprzętowa obliczania każdego przecięcia za pomocą dwóch sprzężonych sumatorów.

Rozważmy jaki sprzęt jest niezbędny do rozwiązania równania przecięcia wymagającego wykonania dzielenia $(x_L - x_1)$ przez $(x_2 - x_1)$ i mnożenia przez $(y_2 - y_1)$.

Obie operacje wykonywane są metodą przesuwania i dodawania i mogą być przeprowadzane jednocześnie, przy czym bity ilorazu, w miarę ich pojawiania się, sterują urządzeniem mnożącym. Po dokładniejszym przeanalizowaniu okazuje się, że nie tylko wymagania dotyczące sprzętu są w tej sytuacji takie same jak dla algorytmu punktu środkowego, lecz jeżeli zastosuje się proces dzielenia bez pamiętania reszty można zauważyć, że strategia sterowania jest taka sama jak strategia wyboru następnego punktu środkowego. A więc z punktu widzenia realizacji sprzętowej, istnieje tylko jeden algorytm okienkowania.

W realizacji mikroprocesorowej, obliczenia potrzebne do wykonania operacji okienkowania, zarówno według pierwszego jak i według drugiego algorytmu mogą być zatem realizowane za pomocą prostego mikroprocesora specjalizowanego, zawierającego dwa sumatory współpracujące z rejestrami przesuwymi.

● Skalowanie

Dla punktu Q na rys. 2 transformację skalowania dla współrzędnej x określa zależność

$$x_s = (x_p - x_{wo}) \frac{v}{w} + x_{vo} \quad (2)$$

gdzie: x_s jest współrzędną przetransformowanego punktu Q.

Podobną transformację stosuje się dla współrzędnej y. Nietrudno zauważyć, że równania (1) i (2) są podobne, przy czym równanie (2) można również rozwiązywać za pomocą specjalizowanych układów zawierających sumator, wykonujących zarówno dzielenie jak i mnożenie. W wypadku, gdy znana jest wartość współczynnika skali v/w , wymagane jest tylko wykonanie jednego mnożenia.

Skalowanie wstępne wymaga tylko mnożenia wartości każdej współrzędnej przez określony współczynnik.

Należy zauważyć, że do wykonania obrotu przetransformowane współrzędne punktu (x, y) obróconego o kąt θ mogą być określone w następującej postaci

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (3)$$

Wartości współrzędnych (x, y) i (x', y') odnoszą się do punktu środkowego, który nie musi być punktem środkowym obrazu wirtualnego. Obliczenia sprowadzają się zatem do wykonania dwóch mnożeń dla każdej współrzędnej, co w realizacji sprzętowej można uzyskać za pomocą dwóch sumatorów na współrzędną. Funkcje $\sin \theta$ i $\cos \theta$ są obliczane programowo lub ze względu na szybkość, odszukiwane w tablicy umieszczonej np. w pamięci stałej (ROM).

Podane algorytmy mają na celu minimalizację czasu przetwarzania bez nieproporcjonalnego wzrostu kosztu, przez co większa ilość danych może być wyświetlana bez migotania.

● Możliwe realizacje praktyczne

Cechą charakterystyczną współczesnych procesorów ogólnego zastosowania, w tym również większości dostępnych na rynku mikroprocesorów monolitycznych jest stosowanie równoległych rejestrów i akumulatorów współpracujących tylko z jednym arytmometrem. Procesor obrazowy musi być natomiast przystosowany do przetwarzania dużych ilości danych, przy czym dla tych danych wykonuje się wiele niezależnych obliczeń arytmetycznych, np. przy skalowaniu segmentu liniowego każda z czterech wartości współrzędnych jest przedmiotem podobnych przekształceń i w celu zwiększenia szybkości przetwarzania pożądane byłoby zastosowanie czterech ciągów realizujących operacje arytmetyczne.

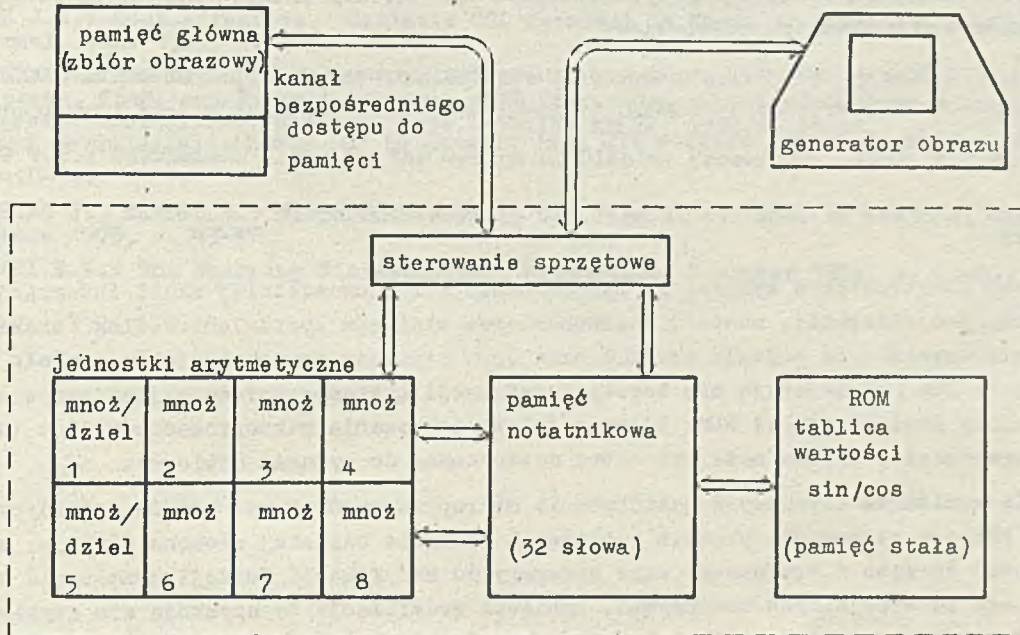
Równoległa realizacja operacji arytmetycznych jest pożądana również dla operacji okienkowania, przy wykonywaniu której zdarza się, że trzeba obliczyć współrzędne przecięcia linii nawet z czterema krawędziami. Szybkie wykonanie tych obliczeń za pomocą maszyny uniwersalnej wymaga stosowania kosztownego komputera z rozbudowaną jednostką arytmetyczną.

Znacznie efektywniejsze w tym wypadku jest zastosowanie urządzenia specjalizowanego np. w postaci sieci mikroprocesorów monolitycznych lub w postaci zespołu jednakowych arytmometrów, kompletowanych w zależności od potrzeb z odpowiedniej liczby mikroprocesorów segmentowych. Wynika to z faktu, że w danym momencie wykonywane są podobne operacje na tym samym zbiorze danych, przy czym operacje te mogą być wykonywane za pomocą prostych jednostek arytmetycznych sterowanych jednostką sterującą, zaprojektowaną specjalnie dla systemu graficznego.

● System graficzny HIEROGLIPH

Przy budowie systemu HIEROGLIPH (High-speed Interpretation and Execution of a Ring-Organised Graphics List by Interactive Processing Hardware) przyjęto omówioną wyżej koncepcję urządzenia specjalizowanego. Podstawowe elementy systemu przedstawiono na rys. 3. HIEROGLIPH jest systemem autonomicznym, sterowanym instrukcjami przechowywanymi w zbiorze obrazowym, umieszczonym w dzie-

lonej pamięci głównej. Współrzędne mogą tworzyć część zbioru obrazowego lub zarówno dane jak i zbiór mogą istnieć jako część większej struktury danych. Zbiór instrukcji systemu HIEROGLIPH zaprojektowano specjalnie do sterowania przepływem danych w urządzeniach grafiki komputerowej, a także do wykonywania stosunkowo prostych manipulacji ze strukturą danych, włączając w to możliwość łączenia w sieci podprogramów graficznych, pośrednie adresowanie danych i skoki warunkowe. Mikroprogramowany układ sterowania systemu HIEROGLIPH interpretuje instrukcje, a następnie szczegółowo ustala sposób wykorzystania rejestrów i kolejność przetwarzania danych.



Rys. 3. Schemat blokowy systemu graficznego HIEROGLIPH

Procesor obrazowy systemu HIEROGLIPH zawiera pamięć notatnikową o szybkim dostępie i pojemności 32 słowa, wykorzystywaną w procesie przetwarzania do przechowywania okna, obszaru widzenia i innych stałych oraz wyników pośrednich otrzymywanych z jednostek arytmetycznych. W celu uzyskania dużej szybkości przetwarzania, stosuje się 8 jednostek arytmetycznych, pozwalających efektywnie realizować algorytmy graficzne omówione wcześniej.

Do wykonania obrotu segmentu linii należy wykonać dwa mnożenia w celu określenia transformowanych wartości każdej z czterech podanych współrzędnych. Aby uzyskać większą szybkość mnożenie jest wykonywane jednocześnie w ośmiu jednostkach arytmetycznych. Wartości funkcji sinus i cosinus pobierane są z pamięci stałej (ROM) o szybkim dostępie, przy czym w razie potrzeby wykonuje się prostą interpolację, co zastępuje stosunkowo wolne ładowanie występujące w przypadku sterowania programowego przez procesor główny.

Kody wyjściowe (outcodes) są generowane za pomocą ośmiu jednostek arytmetycznych. Kody te uzyskiwane są w wyniku odejmowania współrzędnych końców linii od współrzędnych krawędzi okna, co pozwala stwierdzić, czy linia przecina jedną lub więcej krawędzi. Jak pokazano poprzednio, rozwiązanie równań przecięć wymaga wykonania dla każdej krawędzi operacji dzielenia i mnożenia. W tym wypadku potrzebne są jednak tylko dwa ilorazy: gradient $\Delta y/\Delta x$ i jego odwrotność $\Delta x/\Delta y$, a więc wykonanie operacji obcinania wymaga dwóch jednostek arytmetycznych dla dzielenia (1 i 5, rys. 3) i czterech jednostek (2, 3, 6, 7) dla mnożenia. Sześć jednostek działa jednocześnie, operacje mnożenia sterowane są przez bity ilorazu w miarę ich otrzymywania.

Realizacja algorytmu skalowania jest bardzo podobna do realizacji algorytmu obcinania, chociaż na ogół potrzebny jest tylko jeden iloraz. W skalowaniu zwykle wprowadza się współczynnik v/w . Ponieważ w ogólnym wypadku okno i pole widzenia nie mają identycznych skal dla obu współrzędnych, należy przewidzieć dwie operacje dzielenia. Zatem konfiguracja jednostek arytmetycznych w operacji skalowania jest taka sama jak w operacji obcinania. W pewnych sytuacjach może być pożyteczne zapamiętanie w pamięci notatnikowej ilorazu v/w , co redukuje skalowanie tylko do czterech mnożeń, jak w skalowaniu wstępnym.

Współrędne przetworzone przez system HIEROGLIPH przesyłane są do generatora obrazu, wykonanego według standardowych metod konstruowania cyfrowego analizatora różniczkowego o działaniu autonomicznym.

W pierwszej wersji konstrukcyjnej procesora HIEROGLIPH zastosowano standardowe układy TTL, które są dostosowane do szybkości współczesnych pamięci i wykonują przetwarzanie niezbędne dla segmentu linii w czasie około 10 μ s. Ta szybkość przetwarzania umożliwia wyświetlenie złożonych obrazów, z częstotliwością powtarzania wystarczająco wysoką, aby uniknąć migotania obrazu przy stosowaniu luminoforu o krótkiej poświacie.

Autorzy [1] wskazują na celowość realizacji mikroprocesorowej wersji systemu HIEROGLIPH. Jest to zgodne z powszechnie przyjętym poglądem, że stosowanie mikroprocesorów jest bardzo opłacalne w urządzeniach, w których liczba układów TTL jest większa od 50, a według niektórych źródeł już nawet od 15.

4. PODSUMOWANIE

Z doświadczeń konstruktorów wynika, że zastosowanie struktur wielkiej skali integracji zmniejsza koszty projektowania, montażu i uruchamiania systemów cyfrowych. Według Yasaki [8] zastosowanie mikroprocesorów pozwala skrócić czas opracowywania nowych systemów o około 50%. Znacznie (o około 30%) zmniejszają się koszty modyfikacji systemów sprowadzające się w wielu wypadkach do wymiany pamięci stałej ROM. Istotną zaletą stosowania mikroprocesorów jest to, że projektowane urządzenie cyfrowe może być łatwo dostosowane do wymagań odbiorcy.

W dziedzinie monitorów ekranowych zastosowanie mikroprocesorów w porównaniu z tradycyjnymi rozwiązaniami układów sterowania pozwala realizować znacznie bardziej złożone funkcje, zmniejszyć ogólny koszt sprzętu i zredukować czas wymagany do modyfikacji funkcji urządzenia końcowego przeznaczonego do specjalnych zastosowań. Ponieważ modyfikacje te uzyskuje się zwykle za pomocą tego samego mikroprocesora, znakomicie upraszczają się zagadnienia testowania nowych rozwiązań, nie zmienia się niezawodność i zasady obsługi serwisowej. Różne modyfikacje monitorów wyposażone w ten sam mikroprocesor mogą być wytwarzane na tej samej linii produkcyjnej, a programy specjalizowane wprowadzane są do pamięci mikroprocesora po zakończeniu testowania wstępnego. Monitory ekranowe wyposażone w mikroprocesory spełniają zatem wymagania procesu produkcji wielkoseryjnej, a jednocześnie mogą realizować funkcje specjalne, wymagane tylko przez nielicznych odbiorców.

Z przeglądu praktycznie zrealizowanych konstrukcji wynika, że w standardowych monitorach alfanumerycznych z powodzeniem mogą być wykorzystywane mikroprocesory dostępne w handlu, natomiast bardziej złożone systemy wykorzystujące monitory alfanumeryczne np. omówiony autonomiczny procesor tekstowy, jak i monitory graficzne z reguły wymagają stosowania specjalizowanych układów wielkiej skali integracji.

Należy przewidywać, że tendencja do wykorzystywania mikroprocesorów w monitorach ekranowych będzie się umacniać w miarę tego, jak tradycyjne układy MOS będą zastępowane układami szybszymi (SOS/CMOS, LOC MOS, bipolarnymi) oraz w miarę doskonalenia procesów technologicznych pozwalających obniżać koszt podzespołów wielkiej skali integracji. Dotyczy to przede wszystkim monitorów graficznych, w których stosowane są złożone układy elektroniczne o dużej szybkości działania.

W najbliższej przyszłości należy zatem oczekiwać pojawienia się nowych strukturalnych rozwiązań monitorów ekranowych charakteryzujących się tym, że wiele złożonych funkcji wykonywanych za pomocą programów w komputerze centralnym będą realizowały standardowe i specjalizowane mikroprocesory. Urządzenia peryferyjne z monitorami ekranowymi będą wykonywały coraz bardziej złożone operacje predystynujące je do miana inteligentnych terminali i coraz częściej pojawiać się będą systemy z monitorami ekranowymi o dużych możliwościach przetwarzania informacji, czego zwiastunem jest omówiony autonomiczny procesor tekstowy. Należy również przypuszczać, że takie parametry, jak cena i gabaryty tych rozbudowanych urządzeń nie będą różniły się znacznie od odpowiednich parametrów urządzeń stosowanych obecnie.

5. LITERATURA

- [1] CHAMBERLAIN J.T., HANKINS H.C.A., GRAY J.P.: Digital Hardware Generation of Computer Graphic Picture Functions with Fast Windowing Properties. 1974 Comp. Systems and Technology Conf. Publ. Nr 121, s. 167-72.
- [2] CROPPER L.C., WHITING J.W.: Microprocessors in CRT Terminals. Computer, August 1974, s. 48-53.
- [3] Mc DERMOTT J.: Improved Solid-state Memories and Microprocessors Altering the Structure of Computers. Electr. Design. 22, Oct. 25, 1974, s. 34-38.
- [4] DOUB J.A.: Cost-effective, Reliable CRT Terminal Is First of a Family. Hewlett-Packard Journal, June 1975, s. 2-5.
- [5] MATTHEWS S.B.: Microprocessor Architecture in Stand Alone CRT Word Processors - Int. Electron. Conf. and Exposit. Toronto 1975 Conf. Dig. s. 146-47, streszczenie: Ekspres Informacija-Vyčislitel. naja Technika nr 22, 1976, s. 14-16.
- [6] MYER T.H.; SUTHERLAND I.E.: On the Design of Display Processors. Comm. A.C.M., 1968, 11, s. 410-14.
- [7] WHITING J., NEWMAN S.: Microprocessors in CRT Terminals. Proc. of National Computer Conference 1975, s. 41-45.
- [8] YASAKI E.K.: The Emerging Microcomputer. Datamation. December 1974, s. 81-86.

mgr inż. Nina BUDZYŃSKA
Instytut Maszyn Matematycznych

MIKROPROCESOROWY STEROWNIK KASETY CAMAC TYPU CMC 8080

1. Wstęp

Idea zastosowania maszyn cyfrowych do bezpośredniego sterowania procesami produkcyjnymi narodziła się wraz z powstaniem maszyn cyfrowych. Wykorzystanie ich zaczęło być realne dopiero wówczas, gdy poziom niezawodności maszyn cyfrowych zbliżył się do poziomu niezawodności urządzeń obiektu przemysłowego.

Zadania maszyn cyfrowych zastosowanych do automatyzacji produkcji mogą być bardzo różne, poczynając od prostej kontroli (np. temperatury) aż do sterowania procesami i obiektami produkcyjnymi. Każdy system sterowania obiektem, prócz maszyny cyfrowej (MC) i obiektu, musi zawierać specjalne urządzenia sprzęgające MC z obiektem i specjalne oprogramowanie. Najczęściej systemy stanowią rozwiązanie jednostkowe; stąd duże koszty wdrażania systemów automatyzacji produkcji.

Zastosowanie mikroprocesorów do systemów automatyzacji produkcji zwiększa niezawodność systemów. Masowa produkcja mikroprocesorów zmniejsza koszty systemów. Szerokie zastosowanie sprzętu CAMAC* przyjętego przez wiele krajów jako standardowe rozwiązanie systemów pozwala wykorzystywać układy produkowane przez różnych producentów w kraju i za granicą. Zmniejsza to czas i koszty opracowania systemów, upraszcza ich konserwację.

Poniżej podano krótki opis konstrukcji i struktury logicznej CMC 8080 (CAMAC microprocessor controller). CMC 8080 jest to sterownik kasety CAMAC oparty na mikroprocesorze INTEL 8080. Znajomość możliwości CMC 8080 może wpłynąć na decyzję wykorzystania kaset CAMAC i sterowników typu CMC 8080 w projektowanych w Polsce systemach automatyzacji produkcji.

2. Konstrukcja mechaniczna i elektryczna

Sterownik CMC 8080 zbudowany jest na 2 standardowych pakietach CAMAC zajmujących zawsze pozycje 24 i 25 w kasecie CAMAC. Pakiet 24 ma standardowy interfejs CAMAC obowiązujący dla (1 ÷ 23) pakietów kasety. Pakiet 25 ma również standardowy interfejs CAMAC dla sterownika znajdującego się na tej pozycji. Pakiety sterownika są połączone między sobą łączami elastycznymi zakończonymi z obu stron identycznie jak mikroukład TTL. Takich łączy jest 5. Cztery łączy 14-przewodowe korzystają z podstawek mikroukładów 14-nóżkowych, jedno łączy 16-przewodowe korzysta z podstawek 16-nóżkowych. Wszystkie mikroukłady na pakietach są również umieszczone w podstawkach.

Płyta czołowa sterownika zawiera:

- przycisk RESET (zerowanie ogólnie),
- przełącznik ON-OFF do wykorzystania przez programistę w systemie np. przełączanie programów system - użytkownik,
- 15-kontaktowe złącze do dołączania urządzenia wejścia-wyjścia do mikroprocesora,
- 50-stykowe złącze, za pomocą którego można włączyć kasetę w system CAMAC lub dołączyć niezależną transmisję do pamięci operacyjnej CMC 8080 (Direct Memory Access - DMA); złącze to przykryte jest płytą czołową.

*

CAMAC jest to system, w którym znormalizowane są wymiary pakietów, wymiary kasety, sygnały interfejsu; styki łączówki przyporządkowane są poszczególnym sygnałom.

Sygnały interfejsu wspólne dla wszystkich pakietów są połączone w tzw. magistralę obejmującą 24 pakiety. Sygnały indywidualne dla każdego pakietu, jak np. numer pakietu lub przerwanie - zebrane są na stykach kasety w miejscu 25. Określa to położenie sterownika czyli urządzenia nadzorującego operacje na magistrali. Sterownik zajmuje co najmniej 2 miejsca w kasecie 24 i 25. Dodać należy opis normalizacji sprzętu CAMAC zawiera Polska Norma PN-72/T06530.

Rozmieszczenie układów na poszczególnych pakietach sterownika jest zależne od miejsca zajmowanego przez pakiet w kasecie czyli od interfejsu CAMAC przynależnego temu miejscu oraz od zależności logicznej między układami (pozwala to zminimalizować połączenia między pakietami).

Pakiet 25 zawiera układy przyjmowania przerwań LAM z kasy CAMAC, dekodery numeru pakietu CAMAC, mikroprocesor INTEL 8080, generator impulsów, rejestr identyfikacji, pamięć operacyjną, układ priorytetu.

Pakiet (w miejscu 24) zawiera układ adaptera INTEL 8080 - CAMAC, układ transmisji szeregowej wejściowej i wyjściowej z programowaną zmianą częstotliwości transmisji, programowany zegar czasu rzeczywistego oraz rejestry wejścia-wyjścia.

3. Opis sterownika CMC 8080

Sterownik CMC 8080 jest sterownikiem aktywnym, w którym zastosowano mikroprocesor INTEL 8080, pamięć operacyjną o pojemności 4 kbajtów i urządzenie wejścia-wyjścia typu monitor dalekopisowy - TTY. Podstawowym zadaniem sterownika CMC 8080 jest przetwarzanie instrukcji przeznaczonych do bloków wykonawczych CAMAC na interfejs CAMAC, tzn. dopasowanie interfejsu INTEL 8080 i interfejsu CAMAC. Bloków wykonawczych CAMAC w jednej kasecie może być do 23. Przeznaczenie ich jest dowolne. Mogą one zawierać układy sprzęgające mikrokomputer z obiektem, dodatkowe pamięci operacyjne, sprzężenie z urządzeniami wejścia-wyjścia lub większymi komputerami itd.

● Schemat blokowy sterownika CMC 8080

Schemat blokowy CMC 8080 przedstawia rys. 1. Można wydzielić na nim następujące układy:

- mikrokomputer składający się z:
 - procesora
 - pamięci operacyjnej (PO),
 - rejestr identyfikacji,
- adapter INTEL 8080 - CAMAC,
- układ przerwań,
- generator impulsów.

● Procesor INTEL 8080

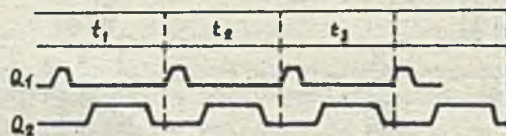
Krótką charakterystyką mikroprocesora

Procesor INTEL 8080 może wykonać 72 instrukcje podstawowe tworzące następujące grupy:

- instrukcje arytmetyki dwójkowej i korektę dziesiętną,
- instrukcje logiczne,
- instrukcje skoków warunkowych i bezwarunkowych,
- instrukcje maskujące lub zezwalające na przyjęcie przerwania,
- instrukcje przesłań między rejestrami procesora INTEL 8080,
- instrukcje przesłań między rejestrami procesora a pamięcią operacyjną,
- instrukcje wejścia/wyjścia.

Procesor INTEL 8080 jest zbudowany techniką MOS. Pracę jego taktuje dwufazowy zegar (nie nachodzące na siebie impulsy Q_1 i Q_2). Takt pracy procesora może zmieniać się od 0,48 μ s do 2 μ s. Dla mikroprocesora w omawianym sterowniku CMC 8080 takt pracy wynosi 1 μ s.

Takt jest to czas od czola impulsu ϕ_1 do czola następnego impulsu ϕ_1 . Między tymi impulsami znajduje się impuls ϕ_2 (w przyjętym rozwiązaniu sygnał ϕ_1 trwa 200 ns, a sygnał ϕ_2 trwa 500 ns).



Rys. 2. Przebieg impulsów Q_1 i Q_2 .

Najkrótszy czas wykonania instrukcji procesora wynosi 4 taktów. Jest to tzw. cykl maszynowy. Liczba cykli maszynowych, z wyjątkiem sumowania długich rejestrów wewnątrz procesora, zależy od liczby dostępow do pamięci potrzebnych do wykonania instrukcji. Korzystanie z urządzeń wejścia-wyjścia wydłuża wykonanie instrukcji o 1 cykl maszynowy.

Instrukcje wykonywane przez INTEL 8080 mogą wymagać od 1 do 5 cykli maszynowych. Liczba taktów poszczególnych cykli maszynowych jest zróżnicowana. Najkrótsze instrukcje składają się z 4 taktów, najdłuższe z 18 taktów.

Interface INTEL 8080

Mikroprocesor INTEL 8080 jest zbudowany w układzie scalonym 40-nóżkowym, którymi wychodzą lub wychodzą następujące sygnały:

• zasilanie (0V, +5V, -5V, +12V)	4 styki	• wejściowe sygnały sterujące	6 styków
• szyna adresowa (16-bitowa)	16 styków	• wyjściowe sygnały sterujące	6 styków
• szyna danych (8-bitowa)	8 styków		

Szyna adresowa. 16-bitowa szyna adresowa umożliwia adresowanie pamięci operacyjnej do 64 K bajtów. Przy wykonaniu operacji wejścia-wyjścia (przez INTEL) na szynę adresową podany jest numer urządzenia, którego dotyczy wykonywany rozkaz.

Szyna informacyjna. Szyna danych jest 8-bitowa, równoległa, dwukierunkowa, co odpowiada długości słowa mikroprocesora INTEL 8080. Szyna danych umożliwia dwukierunkową transmisję danych między procesorem, a pamięcią operacyjną. Szyną tą wysyłany jest bajt identyfikacji, czyli informacja o stanie wewnętrznych przerzutników procesora. Informacja ta jest niezbędna do współpracy układów zewnętrznych z mikroprocesorem. Za pomocą szyny danych realizowana jest transmisja między procesorem, a urządzeniami wejścia-wyjścia w trakcie wykonywania instrukcji IN i OUT.

Wejściowe sygnały sterujące. Wejściowe sygnały sterujące doprowadzone do mikroprocesora INTEL 8080 możemy podzielić na sygnały umożliwiające pracę procesora (zegary Q_1 i Q_2) oraz sygnały powodujące zmianę rytmu procesora: INT, RESET, HOLD, READY. Zegary Q_1 i Q_2 są to impulsy taktujące pracę procesora (patrz: krótka charakterystyka mikroprocesora)

INT	-	interrupt request - żądanie przerwania programu wykonywanego przez procesor i wykonanie programu określonego przez przyjęte przerwanie;
RESET	-	zerowanie; pojawienie się tego sygnału powoduje wyzerowanie licznika rozkazów, przerzutników INTE i HLDA; nie ulegają wyzerowaniu: akumulator, rejestry robocze i wskaźnik stosu;
HOLD	-	trzymaj; sygnał HOLD umożliwia pracę procesora w systemie CAMAC lub DMA (Direct Memory Access - bezpośredni dostęp do P.O.); jest to sygnał żądający od procesora wstrzymania pracy, zwolnienia szyn adresowych i szyn danych;
READY	-	gotowość szyn adresowych i szyn danych. Sygnał ten służy do synchronizacji pracy procesora z urządzeniami zewnętrznymi przez zatrzymanie pracy procesora przy zajętej szynie adresowej i szynie danych. Sygnał READY umożliwia pracę procesora z wolniejszymi pamięciami operacyjnymi. Sygnał READY może być wykorzystany do wymuszania pracy krokowej procesora.

Wyjściowe sygnały sterujące. Są to sygnały wychodzące na zewnątrz mikroprocesora INTEL 8080. Wyjściowe sygnały sterujące dzielą się na:

- krótkie sygnały sterujące tzw. stroby, strobujące moment wykorzystania szyn danych przez procesor - sygnały WR, DBIN, SYNC;
- długie sygnały sterujące, jest to wyjście przerzutników procesora, których stan określa przyjęcie przez procesor żądań wymuszonych z zewnątrz - sygnały HLDA, INTE, WAIT.

WR - WRITE - jest to sygnał strobujący, trwający 1 takt zegarowy (dla sygnału READY=1 log), informujący, że dane z procesora znajdujące się na szynie danych są ustabilizowane i można je pobrać;

- DBIN(DATA BUS IN) - jest to sygnał strobujący trwający 1 takt zegara (przy sygnale READY=1 log), informujący, że szyna danych pracuje jako wejście procesora i należy na nią podać informację z rejestru określonego rozkazem; czas ustabilizowania się danych na szynie określa sam procesor;
- SYNC - (synchronizacji) sygnał strobujący trwający 1 takt zegarowy, który określa początek cyklu maszynowego; w trakcie trwania sygnału SYNC procesor wysyła na szyny danych bajt identyfikacji, określający działanie procesora w tym cyklu maszynowym;
- HLDA - (HOLD ACKNOWLEDGE) potwierdzenie wstrzymania pracy procesora; sygnał ten jest odpowiedzią na sygnał HOLD; sygnał ten w CMC 8080 wymusza blokadę szyn adresowych i szyn danych; sygnał HLDA zezwala urządzeniu wysyłającemu sygnał HOLD na zajęcie szyn adresowych i szyn danych;
- INTE - (Interrupt enable) - odblokowanie przerwania; sygnał ten określa stan przerzutnika (wewnętrznego procesora) zezwalającego na przerwanie; przerzutnik ten może być ustawiony w stan wysokiego potencjału-log "1" tylko rozkazem E1 (interrupt on), a w stan niskiego potencjału-log "0" rozkazem D1 (interrupt off), wykonaniem przerwania lub kluczem RESET; bez ustawienia przerzutnika INTE w stan log "1" żadne przerwanie nie może być przyjęte;
- WAIT - procesor znajduje się w czasie oczekiwania. W stan WAIT procesor wprowadza sygnał READY = 0 oraz rozkaz WAIT (HLT).

● Pamięć operacyjna

Pamięć operacyjna (PO) minikomputera zawiera 4096 słów 8-bitowych. Połowa tej pamięci jest zbudowana na układach scalonych typu PROM i stanowi pamięć niewymazywalną w normalnych warunkach pracy. Jest ona wykorzystana do zapisania "systemu operacyjnego CMC 8080". Pamięć ta wykonana jest na 8 układach scalonych PROM typu SN 1702 po 256 bajtów w każdym i wykorzystuje adresy 0 ÷ 2047.

Druga część jest zbudowana na układach scalonych typu RAM. Pamięć RAM wykorzystana jest do zapisu programów użytkowych. Pamięć RAM umożliwia bezpośredni zapis i odczyt informacji. Niestety nawet krótkotrwały zanik napięcia powoduje zmiany w zapisanej informacji. Pamięć ta mieści się na 16 układach scalonych RAM typu INTEL 2102 po 1024 bitów w każdym. W celu otrzymania bajtu 8-bitowego wysterowuje się jednocześnie 8 układów RAM.

● Urządzenia wejścia-wyjścia

Urządzeniem wejścia-wyjścia dla mikrokomputera w CMC 8080 jest monitor dalekopisowy sterowany za pomocą układu scalonego TMS6011.. Częstotliwość transmisji dla TMS 6011 określa generator programowy zbudowany w układzie CD4059. Zamiast rejestru programowalnego użyto mikroprzełączników. Układ oparty na układach scalonych typu TMS6011 i CD4059 daje duże możliwości wyboru urządzenia wejścia i wyjścia ze względu na częstotliwość pracy. Układy wejścia-wyjścia umożliwiają procesorowi kontakt z operatorem oraz wprowadzanie i wyprowadzanie informacji z zewnątrz.

● Rejestr identyfikacji

Na początku każdego cyklu maszynowego procesor wysyła na szyny danych informację dotyczącą operacji, która będzie wykonana w danym cyklu maszynowym. Informacja ta jest strobowana sygnałem SYNC i zapamiętana w rejestrze identyfikacji.

Bajt identyfikacji zawiera następującą informację:

Tabela 1

Szyna danych	Skrót nazwy sygnału	Wartość aktywna	Instrukcja dotyczy
1	2	3	4
0	INTA	1	Potwierdzenie przyjęcia przerwania

Tabela 1 (c.d.)

1	2	3	4
1	W ₀	0 1	Zapis do PO Odczyt z PO
2	STACK	1	Na szynie adresowej adres wskaźnika stosu (stack pointera) Zapis lub odczyt stosu
3	HLTA	1	Wstrzymanie wykonania instrukcji
4	OUT	1	Zapis informacji do urządzeń zewnętrznych
5	M ₁	1	Początek instrukcji (pobranie instrukcji z pamięci operacyjnej)
6	INP	1	Odczyt informacji z urządzeń zewnętrznych
7	MEMR	1	Czytanie z PO

U w a g a: "1" w tabeli oznacza ustawienie określonego przerzutnika w stan wysokiego potencjału, "0" w tabeli oznacza ustawienie przerzutnika w stan niskiego potencjału.

Wykonanie instrukcji jest podzielone na wykonanie poszczególnych poleceń. W każdym cyklu maszynowym wykonane jest najwyżej jedno polecenie. Różnych rodzajów poleceń wykonywanych przez procesor INTEL 8080 jest dziewięć. Każdemu wykonywanemu poleceniu procesora towarzyszy odpowiedni bajt identyfikacji. W tabeli 2 pokazano przyporządkowanie każdemu poleceniu odpowiedniego bajtu identyfikacji.

Tabela 2

Lp.	Polecenie procesora		Stan rejestru identyfikacji							
			INTA	W ₀	STACK	HLTA	OUT	M ₁	INP	MEMR
1	FETCH	Pobranie instrukcji z PO	0	1	0	0	0	1	0	1
2	MEMORY READ	CZYTAJ z PO	0	1	0	0	0	0	0	1
3	MEMORY WRITE	PISZ do PO	0	0	0	0	0	0	0	0
4	STACK READ	CZYTAJ STOS	0	1	1	0	0	0	0	1
5	STACK WRITE	PISZ STOS	0	0	1	0	0	0	0	0
6	INPUT	CZYTAJ z układu WE	0	1	0	0	0	0	1	0
7	OUTPUT	PISZ do układu WY	0	0	0	0	1	0	0	0
8	INTERRUPT	PRZERWANIE	1	1	0	0	0	1	0	0
9	HALT	CZEKAJ	0	1	0	1	0	0	0	1
10	HALT INTERRUPT	CZEKAJ z PRZERWANIEM	1	1	0	1	0	1	0	0

Każde z poleceń w Tabeli 2 (prócz czekaj) jest wykonane w jednym cyklu maszynowym składającym się z 3 ÷ 5 taktów. Moment, w którym procesor wysyła lub pobiera informację określony jest odpowiednim strobem (WR, DBIN, SYNC).

● Adapter INTEL 8080 - CAMAC

Magistrala kasety CAMAC zawiera następująco sygnały:

Nazwa linii	Oznaczenie	Liczba styków	Uwagi ogólnie
Szyna wejściowa	$R_1 \div R_{24}$	24	Oznaczenia sygnałów stosowane w dokumentacji CMC 8080 są zgodne z normą CAMAC
Szyna wyjściowa	$W_1 \div W_{24}$	24	
Adres wewnętrzny	A_1, A_2, A_4, A_8	4	
Kod instrukcji	$F_1, F_2, F_4, F_8, F_{16}$	5	
Nr pakietu	$N_1 \div N_{23}$	1 na pakiet	
Sygnały sterujące	$S_1, S_2, I, Z, B, C,$	6	
Odpowiedź na instrukcję	X, Q	2	
Przerwanie	LAM	1 na pakiet	

Współpraca procesora z kasetą CAMAC jest zorganizowana dwójście. Ładowanie i pobieranie informacji z rejestrów R i W odbywa się za pomocą rozkazów OUT i IN. Natomiast do wysłania instrukcji do kasety CAMAC wykorzystano instrukcję pisania do pamięci operacyjnej. Stan bitów adresowych $A_{12} \div A_{15}$ określa, czy instrukcja dotyczy kasety CAMAC, czy określa adres pamięci operacyjnej. Gdy spełniony jest warunek: $A_{12} \wedge A_{13} \wedge A_{14} \wedge A_{15} = 1$ instrukcja dotyczy kasety CAMAC, natomiast gdy spełniony jest warunek: $A_{12} \wedge A_{13} \wedge A_{14} \wedge A_{15} = 0$ instrukcja dotyczy zapisu do PO.

Powyższe rozwiązanie zmniejsza maksymalną pamięć operacyjną, którą można dołączyć do INTEL 8080 o 4 K bajtów (zostaje 60 K bajtów), co dla omawianego rozwiązania, mającego tylko 4 K bajty PO nie ma żadnego znaczenia.

Interpretacja instrukcji INTEL 8080 przez adapter jest następująca:

S z y n a a d r e s o w a							
A_0 A_1 A_2 A_3 A_4	A_5 A_6 A_7	A_8 A_9 A_{10} A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	
N	0 0 0	A	1	1	1	1	
Numer pakietu CAMAC		Nr rejestru lub wskaźnika					

S z y n a d a n y c h							
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
F_1	F_2	F_4	F_8	F_{16}	I	C	Z
ROZKAZ							

W skład adaptera INTEL 8080-CAMAC wchodzi:

- rejestry - sterujące
- wejściowe
- wyjściowe
- generator impulsów sterujących
- przerzutniki X i Q.

R e j e s t r y s t e r u j ą c e służą do pamiętania rozkazów dotyczących bloku wykonawczego CAMAC, ponieważ wykonanie rozkazu na magistrali CAMAC trwa dłużej niż cykl maszynowy zapisu do pamięci operacyjnych. Impulsem ładującym rejestry sterujące jest koincydencja sygnałów $A_{12} \wedge$

$\Lambda_{13} \wedge \Lambda_{14} \wedge \Lambda_{15} = 1$ oraz $\overline{WR} = 0$. Ten sam impuls uruchamia generator impulsów sterujących (B, S_1, S_2). Generator impulsów sterujących jest to 12-pozycyjny dynamiczny rejestr szeregowy sterowany z generatora 10 MHz. Przy koincydencji sygnałów jak wyżej - na wejście rejestru podany jest stan wysokiego potencjału - "1" log trwający jeden cykl zegarowy. Następne cykle zegarowe przesuwać tę zawartość przez cały rejestr. Wyjścia odpowiednich pozycji rejestru szeregowego zapalają lub gaszą przerzutniki sygnałów B, S_1, S_2 .

Rejestry wejściowe (R) to rejestry pamiętające informację przesłaną przez blok wykonawczy CAMAC na polecenie "czytaj rejestr", lub bramki umożliwiające pobranie przerwań z poszczególnych bloków wykonawczych CAMAC. Procesor ma dostęp do tej informacji.

Rejestry wyjściowe są to rejestry (W) ładowane z procesora przed wykonaniem rozkazu "pisz rejestr CAMAC".

Wykonanie rozkazu przez bloki wykonawcze CAMAC sygnalizowane jest przez stroby X i Q, które są pamiętane w CMC 8080 i dostępne dla procesora.

● Rozkazy IN i OUT

Rozkazy IN i OUT wchodzi w skład listy rozkazów procesora INTEL 8080. Są one przeznaczone do przysyłania informacji między akumulatorem procesora a urządzeniami wejścia-wyjścia.

Rozkazy IN i OUT są rozkazami dwubajtowymi, przy czym drugi bajt określa adres urządzenia. Ósmobitowa informacja umożliwia zaadresowanie 256 urządzeń. Rozkaz IN może dotyczyć 256 urządzeń wejściowych. Rozkaz OUT może dotyczyć 256 urządzeń wyjściowych. Urządzeniem zewnętrznym może być monitor, czytnik, lub dowolne inne urządzenie peryferyjne, albo po prostu rejestr służący do sterowania lub do przechowywania informacji. CMC 8080 uwzględnia wyżej wymienione możliwości wykorzystania rozkazów IN i OUT. W obecnej konstrukcji CMC 8080 jest 8 rejestrów wejściowych i 6 rejestrów wyjściowych.

Zawartość rejestrów z przypisanym numerem podają tabele 3 i 4. Tabela 3 podaje zawartość rejestrów wejściowych pobieranych za pomocą instrukcji INTEL "IN".

Tabela 3

Adres UWW	S z y n a d a n y c h								U w a g i
	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	
0	LSB							MSB	INFORMACJA z rejestru wyjściowego TTY
1	R1	R2	R3	R4	R5	R6	R7	R8	Rejestry pamiętające zawartość szyn R po wykonaniu rozkazu "czytaj rejestr"
2	R9	R10	R11	R12	R13	R14	R15	R16	
3	R17	R18	R19	R20	R21	R22	R23	R24	
4	Q	X	LAM FLAG	CI ON MANU AL FLAG	ERROR FLAG	REC FLAG	TRANS FLAG	RTCL FLAG	Rejstr sterujący dokładnie omawia Tabela 5
5	L A M								Przerwania przychodzące z poszczególnych pakietów kasety CAMAC
	1	2	3	4	5	6	7	8	
6	L A M								
	9	10	11	12	13	14	15	16	
7	L A M								
	7	18	19	20	21	22	X IRQ		

LSB - najmniej znaczący bit bajtu

MSB - najbardziej znaczący bit bajtu

LAM - (look at me) przerwanie z bloku wykonawczego CAMAC, numer przerwania odpowiada pozycji bloku wykonawczego wysyłającego przerwanie w kasecie CAMAC

X-IRQ - X-interrupt request, żądanie przerwania gdy instrukcja wysłana do bloku wykonawczego CAMAC nie została przyjęta

CIION - CAMAC interrupt on. Przerwania z kasety CAMAC.

Tabela 4 podaje zawartość rejestrów wyjściowych, ładowanych za pomocą instrukcji INTEL "OUT".

Tabela 4

Adres UWW	S z y n a d a n y c h								U w a g i
	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	
0	LSB							MSB	INFORMACJA dla rejestru wyjściowego TTY
1	W1	W2	W3	W4	W5	W6	W7	W8	Rejestry pamiętające informacje dla kasety CAMAC. Przy rozkazie pisz CAMAC zawartość rejestru wysyłana jest na szyny W
2	W9	W10	W11	W12	W13	W14	W15	W16	
3	W17	W18	W19	W20	W21	W22	W23	W24	
4	READ EN	REC EN	TRANS EN	CIION	X-IRQ ON	RTCL ON			Rejestr przechowujący dokładnie podaje Tabela 6
5	2 ⁰	R ¹	2 ²	2 ³					Rejestr wyjściowy dla układu zegara czasu rzeczywistego

Rozkaz IN powoduje przesłanie zawartości rejestru, o określonym w instrukcji adresie, na szynę danych, skąd są one wpisane do akumulatora procesora. Sterownik CMC 8080 nie dubluje pamiętania. Przerwania (LAMY z kasety CAMAC) są pamiętane na poszczególnych pakietach. W sterowniku CMC 8080 są tylko bramki do przesyłania informacji na szyny danych. Rejestr o numerze "0" znajduje się bezpośrednio wewnątrz układu scalonego TMS 6011.

Rozkaz OUT powoduje przesłanie zawartości akumulatora na szyny danych i wysterowanie rejestru, o adresie określonym instrukcją do zapamiętania tej informacji. Rejestr 0 znajduje się wewnątrz układu scalonego TMS 6011.

Najbardziej interesujące dla programisty są rejestry o adresie 4 (IN i OUT), pozwalają one programiście pracować z przerwaniami (interrupt) lub w systemie przeszukiwań (pulling).

Szczegółowe określenie funkcji poszczególnych pozycji rejestru 4 (OUT adres 4) podaje tabela 5.

Tabela 5

Numer szyny D	Oznaczenie sygnału	Stan przetrzutnika	U w a g i
0	READ EN	1	Włącz czytnik taśmy TTY (automatycznie)
1	REC EN	1	Powoduje przerwanie gdy DR = 1 (DR - DATA READY)
2	TRANS EN	1	Powoduje przerwanie gdy TBRE = 1 (TBRE - TRANSMITTER BUFFER REGISTER)
3	CIION	1	Odblokowanie pozycji D3 w rejestrze IN 4
4	X-IRQ ON	1	Powoduje przerwanie gdy \bar{X} = 1 (odrzuć instrukcji CAMAC)
5	RTCL ON	1	Powoduje przerwanie zegarowe RTCL

Szczegółowe określenie funkcji poszczególnych pozycji rejestru 4 (IN adres 4) podaje tabela 6.

Pozycje D0, D1, D4, D5, D6 rejestru IN 4 są pozycjami nieblokowanymi. Zaistniałe zdarzenie ustawia przetrzutnik zgodnie z podanym opisem i umożliwia pracę w systemie przeszukiwań (pulling) z kaseta CAMAC i urządzeniem wejścia-wyjścia TTY.

Pozycje D2, D3, D7 rejestru IN 4 są programowo blokowane lub odblokowywane odpowiednimi pozycjami rejestru OUT 4 "1" log na pozycji D1, D2, D4, D5 rejestru OUT 4 zezwala na pracę z przerwaniami (interrupt).

Tabela 6

Numer szyny D	Oznaczenie sygnału	Stan przerzutnika	U w a g i
0	Q	X	Stan przerzutnika odpowiada wartości sygnału Q ostatnio wykonywanego rozkazu CAMAC
1	X	X	Stan przerzutnika odpowiada wartości sygnału X ostatnio wykonanego rozkazu CAMAC
2	LAM FLAG	1	Gdy suma przerwń GS = 1 (jest chociaż 1 przerwanie)
3	CION MANUAL FLAG	1	Gdy przełącznik jest w położeniu ON sygnał "CION" (D3 OUT 4) = 1
4	ERROR FLAG	1	Gdy przy odczycie wystąpił błąd dotyczący: nieprawidłowej parzystości, nionadążania transmisji, braku bitu stopu (dotyczy TTY)
5	REC FLAG	1	Gdy DR = 1 (rejestr wyjściowy TTY jest zapełniony)
6	TRANS FLAG	1	Gdy TBRE = 1 (rejestr wejściowy TTY jest pusty)
7	RTCL FLAG	1	Przerwanie zegarowe (RTCL)

• Układ przerwń

Do układu przerwń w CMC 8080 prócz omówionych wyżej pozycji rejestru OUT 4 należy układ priorytetu i rejestr RST.

Układ priorytetu jest zbudowany na wejściu kodera. Przerwania są podzielone na 8 grup i priorytet przerwania zależy tylko od pozycji wejścia na koder. Koder nie ma pamiętania. Przyjście przerwania o wyższym priorytecie zmienia stan wyjścia kodera.

Przy realizacji przerwania aktualny stan wyjść kodera jest wpisany do rejestru RST (restart), pozycje odpowiadające (D3 + D5). Na pozostałe pozycje rejestru RST są wpisane "1" logicznie (stan wysokiego potencjału), co daje rozkaz RST. Stan pozycji (D3 + D5) określa adres, od którego rozpocznie się restart, czyli program przerwania. Poniższa tabela określa grupę przerwń dla określonego wyjścia i adres początku programu.

Tabela 7

Rodzaj przerwania	Wyjście układu priorytetu przerwń			Adres restartu	U w a g i
	A2	A1	A0		
	D5	D4	D3		
TTY - IRQ	0	0	0	0	Przerwanie z TTY i zegara
LW1 - LAM1	0	0	1	8	
LW2 - LAM2	0	1	0	16	Priorytet LAN można zmieniać, zmieniając wyjście na układ priorytetu
LW3 - LAM3	0	1	1	24	
LW4 - LAM4	1	0	0	32	
LW5 - LAM5	1	0	1	40	
LW6 - LAM6	1	1	0	48	
LW(7 ÷ 23) LAM(7 ÷ 23)	1	1	1	56	

Jak wynika z tabeli 7 dla układu przerwń zarezerwowane są pierwsze 64 adresy pamięci operacyjnej - systemowej, 8 słów na każde przerwanie. Obszar ten nie mieści programu przerwania, który zresztą jest zmienny, zależnie od przyczyny wywołującej przerwanie i od potrzeb systemu, ale pozwala przyjąć przerwanie i wejść w obszar pamięci użytkownika, gdzie mieści się program przerwania.

● Generator impulsów

W CMC 8080 zastosowano generator kwarcowy typu E PAD 20 MHz, co pozwoliło na pracę z procesorem INTEL 8080 taktami zegarowymi o okresie 0,5 μ s.

Obecnie ta częstotliwość nie jest wykorzystana. Z procesorem INTEL pracuje się taktami zegarowymi o okresie 1 μ s, prawdopodobnie ze względu na potrzebę stosowania mikroukładów o krótszych czasach przełączania pracujących na częstotliwości 20 MHz oraz układy typu RAM.

Za pomocą licznika dwójkowego otrzymuje się częstotliwość 10 MHz, a za pomocą licznika dziesiętnego i dwójkowego - częstotliwość 1 MHz. Zdekodowanie wyjść licznika dziesiętnego pozwala wybrać dowolny z 10 impulsów. Umożliwia to zapalenie i gaszenie przerzutników impulsów Q_1 i Q_2 w odpowiednich chwilach.

● Zegar czasu rzeczywistego

Sterownik CMC 8080 wyposażony jest w programowany układ zegara, zbudowany na układzie scalonym MK 5009. Na ten mikroukład wchodzi sygnał o częstotliwości 0,5 MHz, która jest dzielona w stosunku zależnym od wartości ustawionej programowo w rejestrze RTCL (rejestr OUT adres 5). Zmiana zawartości rejestru zmienia krotność podziału częstotliwości wejściowej, zmienia częstotliwość zegara CMC 8080.

Programowo można nie tylko zmienić częstotliwość zegara CMC 8080, ale i blokować lub odblokowywać przerwania zegarowe.

4. Zakończenie

CMC 8080 jest sterownikiem zbudowanym przez firmę austriacką. W IMM obecnie trwają prace nad zastosowaniem CMC 8080 w komputerowym systemie sterowania procesami dyskretnymi w przemyśle maszynowym.

CMC 8080 można łatwo połączyć z maszyną cyfrową przez specjalny układ lub pakiet typu 500 produkowany przez Zakłady POLON. Można też zwiększać liczbę kaset CAMAC wchodzących do systemu (co określa norma). Systemy oparte na kasecie CAMAC sterowane sterownikiem aktywnym (z mikroprocesorem i pamięcią operacyjną), są bardzo elastyczne. Znaczenie wpływu typizacji urządzeń na koszty i czas opracowania projektu nie wymaga dodatkowego omawiania.

Moim zdaniem korzystniejsze jednak byłoby produkowanie w Polsce sterowników podobnych do CMC 8080, ale bardziej dostosowanych do naszych potrzeb i do naszych możliwości, np. należałoby zwiększyć liczbę mikroukładów typu SN 74173 i DM 8097. Są to układy trójstanowe nie produkowane obecnie w Polsce, a stosowane w CMC 8080 nie tylko przy wyjściu na szynę, ale i jako rejestry o nioblokowanym wyjściu. Niedopracowany wydaje się również system przerwań. Zbyt dużo zadań zostało dla programu np. wyróżnienie przyczyny przerwania o najwyższym priorytecie, albo zerowanie przerzutników powodujących przerwanie typu RTCL, XIRQ. Niektóre sprawy są niemożliwe do rozwiązania przez programistę, np. rozdzielenie zsumowanych błędów powstających przy szeregowym czytaniu do CMC 8080 z TTY.

mgr inż. Krzysztof DZIK
mgr inż. Wojciech KUBERA
mgr inż. Włodzimierz ZAPENDOWSKI
Instytut Maszyn Matematycznych

SPRZĘT A OPROGRAMOWANIE
O PEWNYCH KIERUNKACH ROZWOJU ARCHITEKTURY
ELEKTRONICZNYCH MASZYN CYFROWYCH

Pojawienie się w połowie lat sześćdziesiątych rodzin maszyn cyfrowych trzeciej generacji miało sygnalizować przewrót w technice obliczeniowej. Szumne reklamy zapowiadały pełne zaspokojenie wszystkich potrzeb każdego użytkownika.

Wydajny, niezawodny sprzęt i bogate oprogramowanie w połączeniu z koncepcją rodziny maszyn miało być uniwersalnym antidotum na wszystkie dotychczasowe bolączki użytkowników i producentów. Okres euforii minął jednak dość szybko po skonfrontowaniu oczekiwań z rzeczywistością. Zaczęto dostrzegać wiele negatywnych zjawisk określanых terminem "kryzys software'owy". Kryzys ten polega na:

- nadmiernym przeciążeniu systemu funkcjami administracyjnymi, które w znacznym stopniu obniżają rzeczywistą wydajność systemu jako całości; funkcje systemu operacyjnego związane z wieloprogramowością, czy stosowaniem przez użytkowników języków programowania wysokiego poziomu pochłaniają znaczną część czasu pracy jednostki centralnej, dość powiedzieć, że w niektórych systemach na samą translację programów zużywa się 70% czasu pracy; w tym kontekście staje się jasne stanowisko użytkownika, którego przestaje interesować fakt, że procesor, którym dysponuje wykonuje kilka milionów dodawań na sekundę; dla użytkownika jest to zupełnie nieistotne; przecież jego maszyna z tą właśnie zawrotną prędkością nie robi nic innego tylko "liczy" - system operacyjny!
- olbrzymich kosztach oprogramowania (we współczesnych systemach specjalnych koszt sprzętu nie przekracza 25%), które odznaczają się tendencją wzrostową i są niewspółmierne w stosunku do osiągniętych efektów.

Naszym zdaniem jedną z podstawowych przyczyn "kryzysu software'owego" jest niedopasowanie tradycyjnej, pochodzącej jeszcze od von Neumanna struktury systemu do potrzeb współczesnego użytkownika. Można zaryzykować twierdzenie, że cechą charakterystyczną sprzętu większości dotychczas projektowanych maszyn cyfrowych jest logiczny prymitywizm. W większości znanych systemów jedynym elementem aktywnym jest sumator. Prymitywizm sprzętu w konsekwencji powoduje ograniczonąność języka wewnętrznego maszyny.

Z drugiej strony struktura pamięci w większości znanych rozwiązań jest liniowa. Nie odpowiada to złożonym strukturom zbiorów danych, charakterystycznym dla realnych problemów. Istnieje zatem sprzeczność między językiem wewnętrznym maszyny a językami programowania oraz między liniową strukturą pamięci a złożonymi strukturami danych stosowanymi przez użytkowników.

Przepaść dzieląca skrajnie złożone problemy, które są rozwiązywane za pomocą maszyn cyfrowych i skromne możliwości logiczne współczesnego sprzętu, musi być zasypana przez oprogramowanie, które siłą rzeczy rozrasta się do monstrualnych rozmiarów. Ta sytuacja zmusza projektantów systemów do nowego spojrzenia na sposób projektowania maszyn cyfrowych.

Systemy cyfrowe projektowane były dotychczas "z dołu do góry", tzn. od szczegółu do ogółu: wychodząc z ogólnych założeń konstruowano pewien sprzęt, dla którego tworzono systemy operacyjne, translatory, programy użytkowe - wszystko po to, aby system cyfrowy był użyteczny. Jak głęboko przeniknęła ta metoda do świadomości konstruktorów, świadczy samo słowo "oprogramowanie", które w oczywisty sposób kojarzy się etymologicznie z czynnością wyposażenia czegoś w programy, czyli z pisaniem programów dla zadanego, istniejącego już sprzętu.

Zdaniem autorów, właściwszym od tradycyjnego sposobem projektowania byłoby podejście "z góry w dół"; można sobie wyobrazić, że wszystkie funkcje projektowanego systemu, realizowane w nim al-

gorytmy, przebiegające w nim procesy będą opisane w języku wysokiego poziomu. Na podstawie takiego opisu można by określić jakie procesy, procedury lub ich części będą realizowane przez sprzęt.

W ten sposób może być urzeczywistniona koncepcja "osprzętowania" systemu (a nie oprogramowania). Spodziewamy się, że tego rodzaju podejście, o ile uda się je zrealizować, będzie prowadzić do lepszych niż dotychczasowe rozwiązań, chociażby dlatego, że w momencie projektowania sprzętu wszystkie funkcje przyszłego systemu będą już dobrze określone.

Naturalną konsekwencją podejścia "z góry w dół" będzie tendencja do wbudowywania w system coraz bardziej złożonych mechanizmów sprzętowych, realizujących funkcje, które dotychczas były wykonywane przez oprogramowanie. Zmniejszy to znacznie dysproporcje pomiędzy możliwościami sprzętu i potrzebami użytkownika. Przy konwencjonalnym podejściu do projektowania systemów rozbudowywanie mechanizmów sprzętowych było nieopłacalne ze względu na ryzyko ich niewykorzystania.

Przypuszcza się, że w najbliższych latach będą realizowane przez sprzęt przede wszystkim funkcje systemu operacyjnego (przykład: Hardware Control Program w IBM System/370 model 158).

Tendencja ta okaże się tym bardziej uzasadniona jeżeli weźmie się pod uwagę wpływ nowych technologii (zwłaszcza zastosowanie układów LSI) na cenę sprzętu oraz fakt, że w obecnych systemach koszt jednostki centralnej stanowi około 3% kosztów całości systemu.

Zdaniem autorów wymieniona tendencja rozwojowa będzie się przejawiać najsilniej w dwóch podstawowych zagadnieniach omówionych niżej.

● Organizacja (system adresowania) pamięci

Jedną z podstawowych trudności występujących w rozwiązywaniu problemów za pomocą maszyny cyfrowej jest konieczność zanurzenia struktury informacji, specyficznej dla danego problemu, w liniową strukturę pamięci maszyny. Zastosowanie elastyczniejszej struktury pamięci pozwoliłoby na odciążenie programistów oraz uproszczenie i skrócenie programów translujących systemu operacyjnego.

● Język wewnętrzny maszyny

Zastosowanie języka o wyższym poziomie niż było to dotychczas przyjęte, jako realizowanego sprzętowo języka wewnętrznego, z pewnością ułatwi w znacznym stopniu tworzenie oprogramowania, pozwoli na jego uproszczenie, zmniejszenie objętości i zwiększenie efektywności.

W niniejszej pracy omówione zostaną dwa systemy, ilustrujące sposób realizacji przedstawionej tendencji: maszyna Iliffe'a i maszyna Symbol.

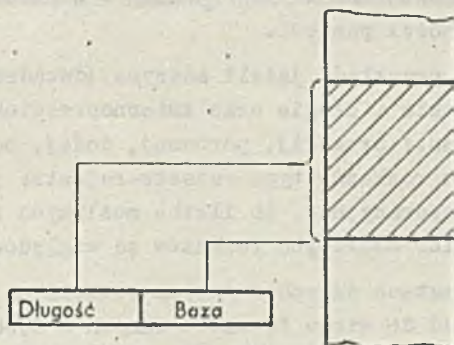
Maszyna Iliffe'a jako przykład systemu o rozwiniętej strukturze pamięci

Punktem wyjścia koncepcji tzw. maszyny bazowej (basic machine) Iliffe'a było odrzucenie tradycyjnej liniowej struktury pamięci. To, że tradycyjna struktura jest niewygodna, zauważono dość wcześnie. Przedstawiono też pewne propozycje zmierzające do uelastycznienia organizacji pamięci, Iliffe nie tylko zaproponował nową strukturę pamięci, ale w sposób bardzo systematyczny przeanalizował konsekwencje, jakie przyjęcie tej struktury wnosi do architektury systemu. Dlatego też koncepcja Iliffe'a jest szczególnie interesująca.

Konieczność strukturalizacji pamięci przez system wyniknęła po raz pierwszy w związku z wprowadzeniem wieloprogramowości. Zaistniała wtedy konieczność dynamicznego przydzielania obszarów pamięci operacyjnej poszczególnym programom. Najprostszym sposobem realizacji takiego podziału jest zadawanie obszarów pamięci przydzielanych programom przez tzw. pary graniczne (rys. 1). Para graniczna zawiera bazę, czyli adres początku obszaru oraz długość (liczbę słów) obszaru. Ze względu na konieczność zapobieżenia wzajemnym oddziaływaniom między programami dostęp do par granicznych musi być ograniczony. Z drugiej strony system operacyjny musi mieć możliwość zmiany tych par granicznych. Z tych właśnie powodów konieczne jest wprowadzenie pewnego wyróżnionego stanu maszyny - stanu supervisor'a. Wpływa to oczywiście na skomplikowanie sterowania maszyny.

Koncepcja Iliffe'a jest uogólnieniem idei par granicznych, które spełnia następujące postulaty:

- powinna być możliwa realizacja bardziej złożonych struktur przez ich hierarchizację (struktura drzewa o wielu węzłach),
- powinna istnieć tylko jedna uniwersalna metoda dostępu do danych,
- operowanie strukturą powinno być dostatecznie efektywne - konieczna jest sprzętowa interpretacja struktury.



Rys. 1. Opis obszaru pamięci przez parę graniczną

Pamięć maszyny bazowej można sobie wyobrazić jako strukturę "drzewopodobną". Elementy tego samego rodzaju łączone są w bloki różnych typów zajmujące kolejne miejsca pamięci. Są to w szczególności bloki danych i rozkazów. Z każdym takim blokiem związana jest pewna informacja strukturalna, będąca uogólnieniem pary granicznej, nazywana słowem kodowym. Słowa kodowe mogą być również łączone w bloki, które odpowiadają punktom rozgałęzienia drzewa. Każdy blok opisywany jest przez dokładnie jedno słowo kodowe, a każde słowo kodowe, z wyjątkiem tzw. głównego słowa kodowego, należy do bloku "wyższego rzędu".

Struktura taka ma liczne zalety i jest stosunkowo prosta i nieskomplikowana w realizacji, stanowi proste uogólnienie struktury liniowej - można więc oczekiwać, że wszystkie osiągnięcia w dziedzinie gospodarowania pamięcią w maszynach tradycyjnych dadzą się łatwo przenieść do maszyny bazowej. Struktura drzewopodobna odznacza się dużą elastycznością, łatwością wymiany pewnych elementów, co jest nie bez znaczenia dla uruchamiania i rozbudowy programów. Ponieważ struktura interpretowana jest sprzętowo - możliwe jest wiązanie informacji diagnostycznej ze strukturą programu i danych, co powinno znacznie przyspieszyć proces wykrywania i korygowania błędów w programach. Łatwość oddzielenia części struktury ma duże znaczenie w realizacji procesów równoległych, co jest konieczne dla wieloprogramowości i realizacji programów symulacyjnych. Bardzo ważną własnością jest również to, że każda podstruktura ma budowę analogiczną do całości.

Sprzętowe rozpoznawanie struktury wymaga umieszczenia w słowach kodowych pewnych informacji o zawartości opisywanych bloków. Stwarza to jednocześnie pewne dodatkowe możliwości. Możliwe jest na przykład oddzielenie rozkazów od danych. W modelu von Neumanna rozkazy i dane traktowane były jednakowo jako binarne słowa pamięci. Możliwe było więc operowanie na rozkazach tak, jak na danych. W pierwszych maszynach cyfrowych mechanizm ten był przydatny - pozwalał na modyfikację rozkazu i umożliwiaił organizację pętli programowej. Szybko pojawiły się jednak specjalne mechanizmy służące do tego celu (mechanizm indeksacji) i możliwość operowania na rozkazach była wykorzystywana tylko do programistycznych "sztuczek". Maszyna z możliwością operowania na rozkazach albo nie mogła pobierać rozkazów z pamięci "na zapas" celem przyspieszenia swego działania, albo musiała mieć wbudowane specjalne mechanizmy kontrolne nadzwyczaj rzadko wykorzystywane. Zdolność programu do samomodyfikacji stwarza najtrudniejsze problemy w przenoszeniu programów z jednej maszyny na drugą i jednocześnie niepotrzebnie uszczyplnia format rozkazów maszyny.

Jeżeli oddzielimy rozkazy od danych, co jest bardzo łatwe w strukturze maszyny bazowej, przez co uniemożliwiamy modyfikację programu w czasie jego wykonywania, stanie się możliwe dostosowanie formatu rozkazu do aktualnych potrzeb, np. będzie można dostosować długość części adresowej roz-

kazu do rzeczywistej pojemności pamięci operacyjnej - pozwoli to na oszczędność pamięci zajmowanej przez program.

Dalszą konsekwencją umieszczania w słowach kodowych informacji o typie danych zawartych w opisywanym bloku jest związanie informacji o formacie danych z danymi, a nie z kodem rozkazu - jak to było w maszynach tradycyjnych. Prowadzi to do skrócenia liczby rozkazów, skrócenia kodu operacji, a ponieważ dane umieszczane są najczęściej w blokach zawierających dane tego samego rodzaju - również do oszczędności pamięci.

Illiffe podaje następujący przykład: jeżeli maszyna (dwuadresowa) rozpoznaje cztery typy argumentów (stałoprzecinkowe krótkie i długie oraz zmiennoprzecinkowe krótkie i długie) i sześć typów operacji elementarnych (np.: prześlij, porównaj, dodaj, odejmij, mnoż, dziel) to teoretycznie możliwych jest 96 różnych rozkazów typu rejestr-rejestr; jeżeli każdy z argumentów może znajdować się również w pamięci operacyjnej, to liczba możliwych rozkazów wzrośnie do 384. W praktyce znacznie zmniejsza się zbiór możliwych rozkazów ze względów czysto ekonomicznych.

Wiązanie informacji o formatach danych z danymi pozwoli na rozszerzenie funkcji rozkazu. Jeden rozkaz daje się zastosować do wielu formatów danych o dowolnej długości. Ten sam rozkaz "dodaj" można użyć do wykonania operacji na liczbach rzeczywistych stało- i zmiennoprzecinkowych, do liczb zespolonych, wektorów, a nawet macierzy!

Oprócz zwiększenia mocy pojedynczego rozkazu, jawne definiowanie struktury danych pozwoli na ujednoczenie i uproszczenie niektórych podstawowych algorytmów. Jeżeli, na przykład, należy wykonać pętlę programową, która dokonuje ciągu operacji na kolejnych elementach tablicy, zbyteczne jest określanie jej granic w programie. Sprzętowa interpretacja struktury może zapewnić wyjście z pętli po wyczerpaniu elementów tablicy.

Omówimy teraz bliżej strukturę zaproponowaną przez Illiffe'a. Z każdą informacją w maszynie bazowej związany jest wskaźnik (ang. tag) określający sposób interpretacji danej informacji przez maszynę. Istnieją cztery wartości wskaźnika. Ich znaczenie jest następujące:

- 0 - słowo binarne pojedynczej długości
- 1 - informacja wymaga interpretacji programowej przez przerwanie (np. jeszcze nie zakończono formowania bloku)
- 2 - adres (takim wskaźnikiem opatrywane są słowa kodowe)
- 3 - liczba podwójnej długości

Z każdym blokiem danych wiąże się informacja o rodzaju zawartych w nim danych, tzw. typ, który jest częścią słowa kodowego. Format słowa kodowego przedstawiono na rys. 2.

WSK	TYP	DŁUGOŚĆ	BAZA
-----	-----	---------	------

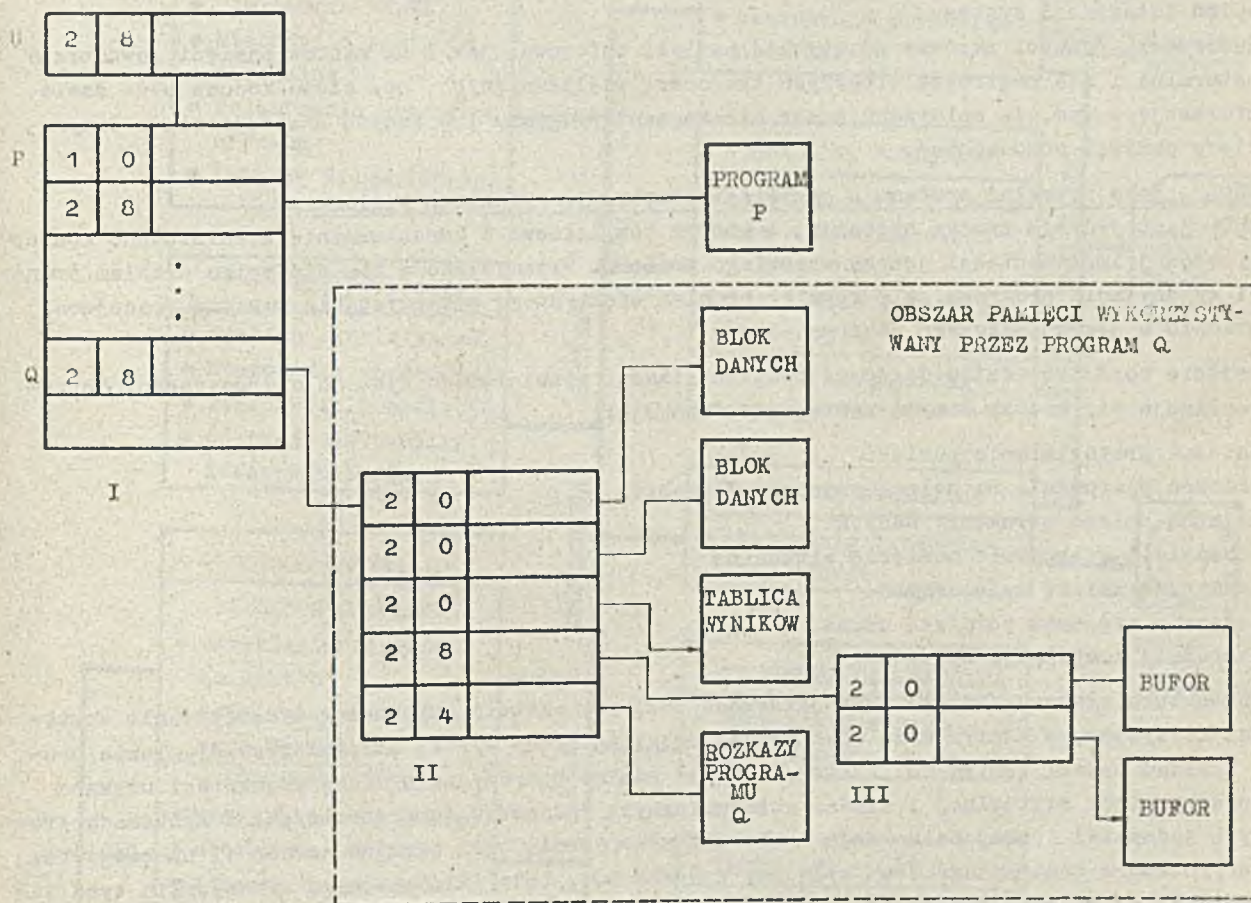
Rys. 2. Format słowa kodowego

Baza (ang. datum) określa adres bazowy (adres początku bloku) informacji opisywanych przez słowo kodowe. Długość (ang. limit) określa rozmiar bloku. WSK oznacza omówiony już wskaźnik. Pole TYP określa rodzaj informacji w odpowiednim bloku. Illiffe wprowadza 16 następujących typów:

- 0 - słowo binarne
- 1 - bajt (znak)
- 2 - słowo binarne (tylko odczyt - próba zapisu powoduje błąd)
- 3 - bajt (tylko odczyt)
- 4 -
- 5 - rozkazy w różnych rodzajach pracy (np. wykonywanie z monitorowaniem)
- 6 -
- 7 -
- 8 - absolutne słowo kodowe (baza jest adresem absolutnym)
- 9 - względne słowo kodowe (baza jest adresem względnym)
- 10 - typ mieszany - w bloku są elementy różnych typów

- 11 - typ mieszany (tylko odczyt)
- 12 - długi element liczbowy
- 13 - krótki element liczbowy
- 14 - długi element liczbowy (tylko odczyt)
- 15 - krótki element liczbowy (tylko odczyt)

Przykład organizacji pamięci w maszynie bazowej przedstawiono na rys. 3.



- U - główne słowo kodowe systemu
- I - słowa kodowe opisujące różne programy w systemie wieloprogramowym
- II - słowa kodowe opisujące strukturę programu i danych
- III - słowa kodowe opisujące strukturę pewnego zbioru danych (tu bufor danych wydzielone dla interakcji)

· Rys. 3. Przykład struktury pamięci w maszynie Iliffe'a

Koncepcja Iliffe'a doczekała się praktycznej realizacji w maszynie R-2 zbudowanej na uniwersytecie Rice. Pewnymi szczegółami różni się ona od maszyny bazowej, np. wprowadzono w niej specjalny rodzaj słów kodowych, tzw. słowa sterujące, które związane są z sekcjami programów, czyli ciągami rozkazów.

Konstruktorzy maszyny R-2 podkreślają ułatwienia jakie daje opisana architektura w pisaniu kompilatorów, w programach zarządzających pamięcią i w diagnostyce systemu.

Podsumowując można powiedzieć, że system zaproponowany przez Iliffe'a charakteryzuje się niżej wymionionymi cechami.

- **Elastyczność**

Możliwe jest dostosowanie struktury pamięci do wymagań problemu. Łatwa jest dynamiczna zmiana struktury. Poszerza to, w stosunku do modelu von Neumanna, klasę zadań, rozwiązywanie których jest z ekonomicznego punktu widzenia możliwe za pomocą maszyny cyfrowej.

- **Dokładność reprezentacji**

Sprawdzanie dopuszczalnych zakresów i formatów możliwe jest na poziomie części struktury, a nie całego programu. Zwiększa to efektywność diagnostyki.

- **Stopień integracji systemu**

Rozszerzenie pamięci zarówno na szybkie pamięci buforowe, jak i na masowe pamięci zewnętrzne jest naturalne i nie nastrocza większych trudności realizacyjnych, np. słowo kodowe może zawierać informację o tym, że opisywany przez nie segment programu lub danych znajduje się w danym momencie w pamięci pomocniczej.

SYMBOL - jako przykład systemu z językiem wewnętrznym wysokiego poziomu

SYMBOL jest jedynym znanym systemem, w którym kompleksowo i konsekwentnie zrealizowano koncepcję sprzętowej implementacji języka wysokiego poziomu. Przemyślano w nim nie tylko problem translacji i wykonywania programu, ale również problem efektywnego wykorzystania maszyny pracującej bezpośrednio w języku wyższego poziomu.

Podejście to doprowadziło do sprzętowej realizacji rozwiniętego systemu operacyjnego. Sprzętowo realizuje się między innymi następujące funkcje:

- dynamiczne przydzielanie pamięci
- dynamiczne operowanie na polach zmiennej długości
- dynamiczną zmianę struktury danych
- zarządzanie i sterowanie pamięcią wirtualną
- automatyczną zmianę typu danych
- zarządzanie systemem podziału czasu
- bezpośrednią kompilację sprzętową

Architektura systemu została zaprojektowana tak, aby ułatwić maksymalne wykorzystanie sprzętu. Jest to system, w którym dużą efektywność uzyskano przez wysoki stopień zrównoleglenia operacji. W skład części centralnej systemu wchodzi pamięć operacyjna o dużej pojemności używana jako bufor pamięci wirtualnej i siedem autonomicznych jednostek funkcjonalnych. W systemach tradycyjnych jednostki funkcjonalne mają możliwość wykonywania bądź pewnych sekwencji operacji bez udziału procesora centralnego (tak było np. w GAMMA 60), bądź jednoczesnego wykonywania tych samych operacji na różnych danych (ILLIAC IV). Nowością w systemie SYMBOL jest wyspecjalizowanie jednostek funkcjonalnych systemu do wykonywania poszczególnych etapów procesu przetwarzania i sprzętowa realizacja, związanych z tymi etapami czynności.

SYMBOL jest interesującym systemem również dlatego, że przyjęte w nim rozwiązania zostały sprawdzone w praktyce. Firma Fairchild zbudowała model systemu, który był następnie eksploatowany przez uniwersytet Iowa.

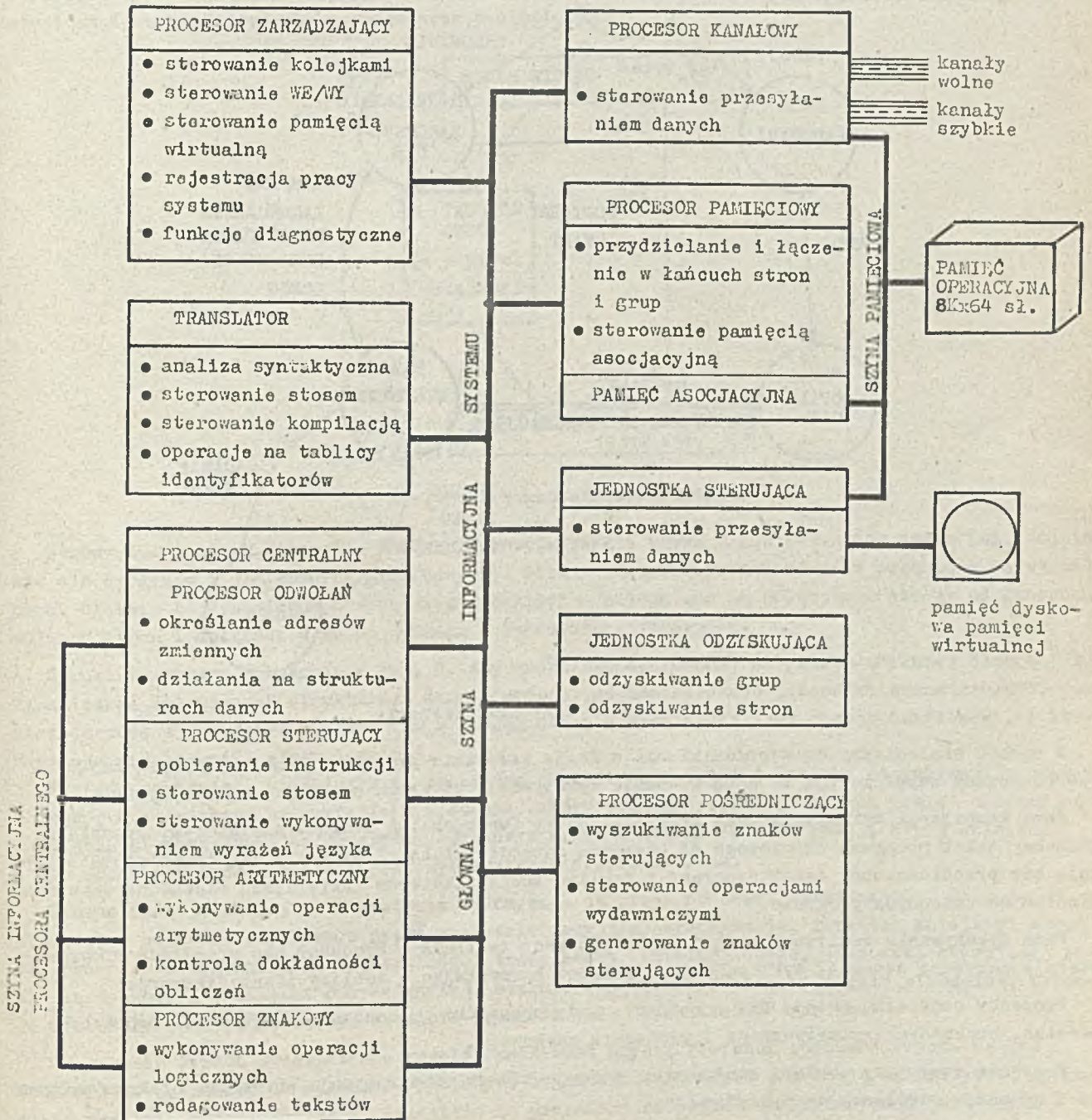
Rys. 4 przedstawia ogólną strukturę systemu. Pokazano na nim główne jednostki funkcjonalne i wchodzące w ich skład układy wyspecjalizowane do wykonywania określonych funkcji.

Na rys. 5 przedstawiono podstawowe fazy przetwarzania. Po nawiązaniu łączności z systemem użytkownik przechodzi do wykonywania fazy ładowania, podczas której wczytywany jest program użytkownika, następnie do fazy translacji i wykonania.

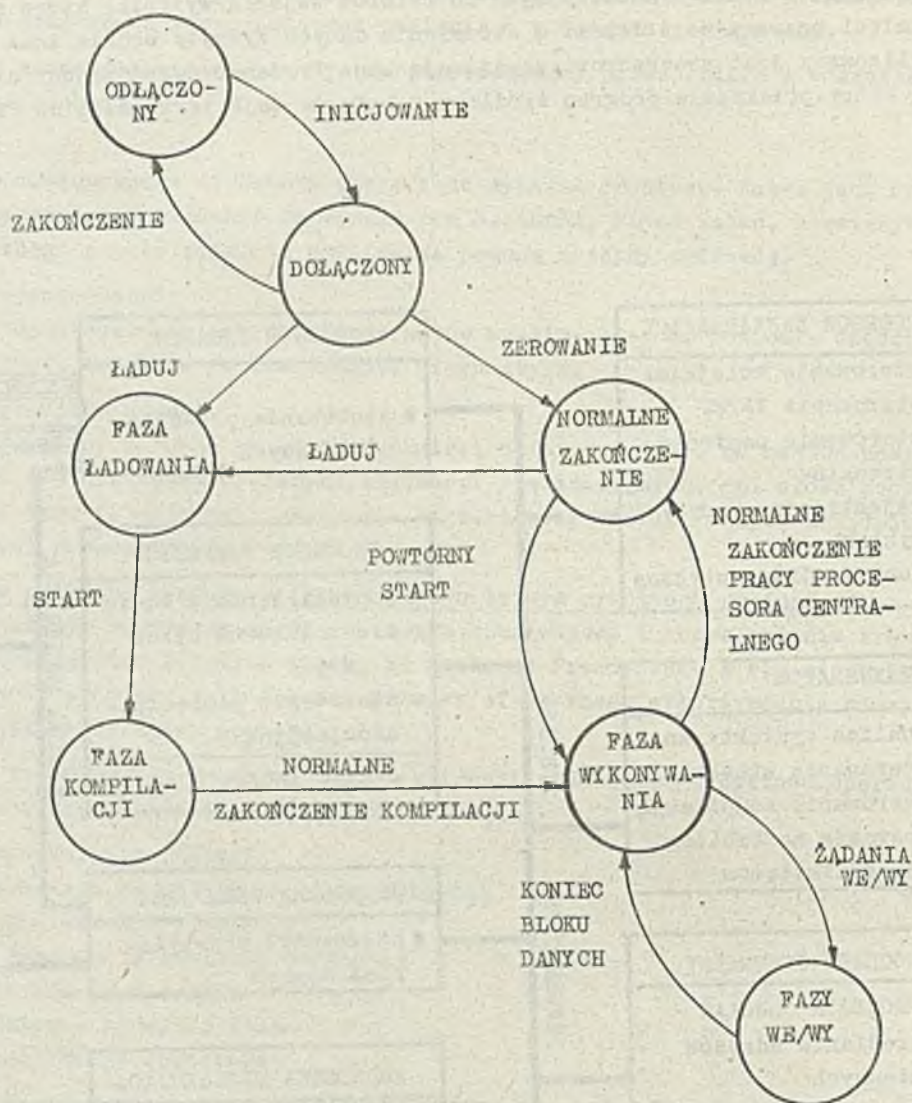
Omówimy teraz współdziałanie procesorów systemu w poszczególnych fazach przetwarzania.

Faza ładowania służy do wprowadzenia programu źródłowego do wydzielonego obszaru pamięci wirtualnej tzw. TWA (ang. Transient Working Area). Procesor kanałowy sterujący kanałami transmisji

danych, przesyła dane z linii transmisyjnych do buforów wejścia/wyjścia, które stanowią wydzielony obszar pamięci operacyjnej. Jeżeli w strumieniu danych wykryty będzie znak końca bloku - fakt ten sygnalizowany jest procesorowi zarządzającemu. Procesor zarządzający uruchamia procesor pośredniczący, który przekazuje program źródłowy z buforów wejścia/wyjścia do TWA,



Rys. 4. Struktura systemu SYMBOL



Rys. 5. Główne fazy przetwarzania

W sposób analogiczny do współdziałania w fazie ładowania procesory kanałowy, zarządzający i pośredniczący współpracują ze sobą w czasie wykonywania zwykłych operacji wejścia/wyjścia.

Faza kompilacji wykonywana jest przez jednostkę funkcjonalną zwaną tłumaczem. Tłumacz tłumaczy tekst programu źródłowego na odwrotną notację polską. W wyniku pracy tłumacza uzyskuje się przetłumaczony tekst programu i tablicę identyfikatorów zawierającą niezbędne dane dla odwołań do zmiennych programu.

Faza wykonywania realizowana jest przez procesor centralny. Procesor ten przegląda program, steruje pamięcią stosową, wybiera dane z pamięci korzystając z tablicy identyfikatorów.

Procesor centralny składa się z czterech podprocesorów: procesora sterującego, procesora odwołań, procesora arytmetycznego i procesora znakowego.

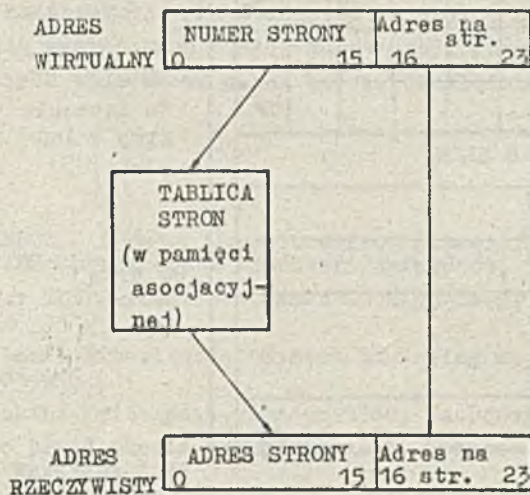
Procesor sterujący pobiera instrukcje, steruje stosem, interpretuje strukturę blokową programu i wykonuje odwołania do procedur.

Procesor odwołań organizuje czynności związane z operowaniem na strukturach danych, a także określa adresy zmiennych na podstawie tablicy identyfikatorów.

Procesor arytmetyczny wykonuje operacje arytmetyczne na zmiennoprzecinkowych liczbach dziesiętnych o zmiennej długości.

Procesor znakowy służy do wykonywania operacji logicznych i redakcyjnych.

Oprócz opisanych już procesorów, w skład systemu wchodzi jeszcze inne jednostki wypełniające funkcje sterujące i usługowe dla pozostałej części systemu, takie jak np. zarządzanie pamięcią operacyjną wykonywane przez procesor pamięciowy. Ze względu na założoną w projekcie sprzętową realizację dynamicznie zmienianych struktur danych i pamięci wirtualnej konieczne było odpowiednie zorganizowanie pamięci operacyjnej. Podstawową jednostką podziału w systemie pamięci wirtualnej jest strona o stałej wielkości (256 słów 64-bitowych). Wymiana stron odbywa się między pamięcią operacyjną i szybką pamięcią dyskową. Strony pamięci wirtualnej mają ustalone położenie w pamięci dyskowej. Translacja adresów wirtualnych na rzeczywiste dokonywana jest za pomocą pamięci asocjacyjnej (rys. 6) w procesorze pamięciowym.



Rys. 6. Przekształcanie adresów

Strony łączy się w łańcuchy. Każdy użytkownik zwykle używa kilku łańcuchów stron (np. oddzielnie dla programu i dla różnych grup danych). Strony niewykorzystane również połączone są w łańcuchy. Ułatwia to dynamiczne przydzielanie pamięci - polega ono na odłączeniu strony od łańcucha wolnych stron i dołączeniu go do jednego z łańcuchów użytkownika.

Strukturę strony pokazano na rys. 7. Aby umożliwić operowanie na polach zmiennej długości i dynamicznie zmienianych strukturach danych wydzielono ośmiosłowe jednostki zwane grupami, również łączone w łańcuchy. Dlatego też w każdej stronie wydzielony jest obszar (nagłówek), w którym zapisuje się łączniki stron i grup.

Funkcje pomocnicze w stosunku do procesora pamięciowego pełni jednostka odzyskiwania. Jest to procesor o najniższym priorytecie w systemie, który systematycznie przegląda pamięć operacyjną i dołącza niepotrzebne już programom użytkowym grupy i strony do list grup i stron wolnych. W ten sposób struktura pamięci jest systematycznie porządkowana.

Główne funkcje sterujące w systemie wykonywane są przez procesor zarządzający. Steruje on przejściem programu użytkowego przez wszystkie fazy jego wykonywania, zarządza kolejkami zadań dla procesorów wykonawczych (centralnego, tłumacza interfejsowego, jednostki sterującej pamięcią dyskową i jednostki odzyskiwania), steruje wejściem/wyjściem, pamięcią wirtualną, wykonuje procedury reakcji na ołędę, wreszcie prowadzi całą "buchalterię" systemu.

W systemie SYMBOL przewidziano również możliwość użycia procedur software'owych. Są one szczególnie potrzebne do wykonania złożonych czynności związanych z reakcją na nietypowe sytuacje w systemie wejścia/wyjścia. Przejście do procedury software'owej wykonywane jest przez przerwanie, które z punktu widzenia sterowania systemem traktowane jest jako start fikcyjnego urządzenia końcowego.

Z użytkowego punktu widzenia SYMBOL jest efektywnym i przez przemysłany dobór języka, bardzo elastycznym narzędziem obliczeniowym. Podstawowymi założeniami dla określenia języka SYMBOL były:

Zakończenie

Na przykładzie dwóch systemów, które oceniliśmy jako charakterystyczne, staraliśmy się pokazać pewne kierunki rozwojowe architektury i metodologii projektowania systemów cyfrowych. Znamienny jest fakt, że obydwa podejścia zaprezentowane w niniejszej pracy doprowadziły do rozwiązań mających wiele punktów wspólnych. Zarówno w maszynie Illiffe'a, jak i w systemie SYMBOL, stosowana jest drzewopodobna struktura pamięci, w obu maszynach informacja o typie danych związana jest z danymi. Maszyna bazowa również wymaga zastosowania języka poziomu wyższego niż w maszynach tradycyjnych (konieczne są dość złożone operacje na strukturach danych), co zbliża ją do koncepcji maszyny SYMBOL. Fakty te dowodzą, że różnorodne koncepcje architektur systemów, stawiające sobie za cel przybliżenie struktury maszyny do wymagań problemów użytkownika, prowadzą do podobnych rozwiązań konstrukcyjnych. Zbieżność ta wydaje się nam nieprzypadkowa. Budzi ona nadzieje, że wkrótce będą wypracowane zasady konstrukcji nowych maszyn cyfrowych, różnych od modelu von Neumanna, których podstawową cechą będzie uwzględnienie rzeczywistych potrzeb użytkownika.

Literatura

- [1] RICE R., SMITH R.: SYMBOL - A Major Departure from Classic Software Dominated von Neumann Computing Systems. AFIPS Spring Joint Computer Conference, 1971, s. 575-587
- [2] CHESLEY G.D., SMITH R.: The Hardware-Implemented High-Level Machine Language for SYMBOL. Ibid., s. 563-573
- [3] SMITH R.: SYMBOL - A Large Experimental System Exploring Major Hardware Replacement of Software. Ibid., s. 601-616
- [4] ILLIFE J.K.: Basic Machine Principles. London 1968, MacDonald
- [5] MYSIOR J.: Sprzęt jako przedłużenie oprogramowania. Opracowanie nr 4 Zakładu Teorii Systemów Operacyjnych IMM, Warszawa 1974
- [6] FEUSTEL E.A.: The Rice Research Computer - A Tagged Architecture. AFIPS Spring Joint Computer Conference 1972, t. 40, s. 369-377

JEDNOSTKI CENTRALNE JEDNOLITEGO SYSTEMU EMC

Przedstawiono krótko kilka charakterystycznych cech poszczególnych jednostek centralnych Jednolitego Systemu EMC.

Najważniejsze dane jednostek centralnych podane są w załączonych zestawieniach. Oceny szybkości działania dokonano za pomocą tzw. mieszanek. Mieszanki tworzy się z bliskich używanym w praktyce - sekwencji rozkazów typowych, grup rozkazów lub typowych zadań. Rozróżnia się mieszanki rozkazów oraz mieszanki problemów lub procedur. Mieszanka JS EMC (modyfikacja I) jest mieszanką rozkazów, stanowi rozszerzoną wersję mieszanki GIBSON-I i przeznaczona jest do oceny wydajności dla zadań naukowo-technicznych. Mieszanka GIBSON-III jest mieszanką problemów zorientowaną na zadania naukowo-techniczne. Zawiera ona około 20% operacji zmiennoprzecinkowych, przy czym w mieszance GIBSON-III-E są używane operacje zmiennoprzecinkowe krótkie, a w mieszance GIBSON-III-D operacje zmiennoprzecinkowe długie. Mieszanka GPO-WO-II zawiera zestaw zadań typowych dla zastosowań komercyjnych. Charakterystyczna dla niej jest orientacja na przetwarzanie informacji symbolicznych (30% informacji symbolicznych i 70% numerycznych), konwersja kodów oraz absolutny brak operacji zmiennoprzecinkowych. Mieszanka ta pozwala na uwzględnienie wielu właściwości modeli takich, jak np. wpływ pracy kanałów i urządzeń wejścia-wyjścia na funkcjonowanie jednostki centralnej, które odgrywają ważną rolę przy rozwiązywaniu zadań planowania i innych zadań ekonomicznych.

Podane czasy wykonywania rozkazów dotyczą rozkazów typowych lub często używanych operacji listy rozkazów JS EMC. Zestawienie A uwzględnia dokładnie te same rozkazy, które występują w mieszance JS EMC (modyfikacja I). Czasy wykonywania rozkazów mogą być podane tylko jako wartości z pewnych przedziałów, ponieważ wykonanie operacji jest zależne m.in. od formatu rozkazu, położenia operandu w pamięci, nałożenia taktów roboczych (wykonywania, odczytu i dekodowania rozkazów), struktury programu pomiaru czasu. Formaty i treść rozkazów jednostek JS 2610 i JS 2612 odbiegają znacznie od zasad działania JS EMC, a więc bezpośrednie porównywanie czasów wykonywania rozkazów z innymi jednostkami centralnymi JS jest możliwe tylko pod pewnymi warunkami. W celu porównania parametrów lub umiejscowienia jednostek centralnych JS 2610 i JS 2612 w stosunku do innych niżej podano czasy wykonania niektórych rozkazów w JS 2610 (Zestawienie B).

• Operacje logiczne

Mnożenie logiczne	2,1... 3,1 μs
Sumowanie logiczne	2,1... 3,1 μs
Różnica symetryczna	2,1... 3,1 μs

• Operacje przesłania

Ladowanie rejestru A, E lub X	2,2... 3,1 μs
Pamiętanie rejestru A, E lub X	2,2... 3,2 μs
Ladowanie bajtu według X	2,6... 3,6 μs
Pamiętanie bajtu według X	2,5... 3,5 μs

Zestawienia podają maksymalne szybkości wymiany danych z pamięcią operacyjną w bajtach na sekundę.

$$\text{Szybkość przesyłania danych} = \frac{\text{liczba bajtów przesyłanych jednocześnie}}{\text{czas cyklu pamięci operacyjnej}} \times \text{współczynnik nakładania}$$

Chociaż na ogół urządzenie przetwarzające nie wykorzystuje w pełni tak wyliczonej szybkości przesyłania, razem z innymi parametrami daje ona dokładniejszą informację o szybkości pamięci operacyjnej niż czas cyklu. Umożliwia ona również wyliczenie stopnia wykorzystania pojemności pamięci operacyjnej przez jednostkę centralną.

Jednostka centralna **JS 2610**

Jednostka centralna JS 2610 jest najmniejszą jednostką centralną w JS EMC z punktu widzenia wewnętrznej wydajności przetwarzania, możliwej wielkości pamięci operacyjnej i możliwości przyłączenia urządzeń wejścia-wyjścia. Przede wszystkim jest ona przeznaczona do wykonywania niewiel-

ZESTAWIENIE A: Jednostki centralne podstawowych modeli JS EMC

Jednostka centralna	JS 2620	JS 2622	JS 2630	JS 2632	JS 2633	JS 2640	JS 2650
Kraj produkujący	ZSRR/DRL	ZSRR	ZSRR	PRL	ZSRR	NRD	ZSRR
Procesor centralny							
Szybkość wykonywania operacji (operacje na sekundę)							
• Mieszanka JS EMC (Modyfikacja I)	9000	91000	60000	230000	200000	380000	500000
• GIBSON-III-E	19000	125000	70000	303000	200000	411000	500000
• GIBSON-III-D	10000	92000	45000	212000	150000	375000	408000
• GPO-VU-II	24000	95000	38000	176000	130000	240000	192000
Lista instrukcji (liczba instrukcji)	143	143	143	143	143	143	143
Sposób sterowania	Mikroprogramowe	Mikroprogramowe	Mikroprogramowe	Mikroprogramowe	Mikroprogramowe	Kombinowane sprzętowo-mikroprogramowe	Sprzętowe
Sterowanie bezpośrednie	istnieje	istnieje	istnieje	istnieje	istnieje	istnieje	istnieje
Pamięć mikroprogramów:							
• pojemność słów	8192	8192	4096	2816	2048	3072	-
• czas cyklu μs	1000	500	500	320	300	450	-
• długość słowa (liczba bitów)	64	64	72	86	128	130	-
Czasy wykonywania wybranych instrukcji (μs):							
• Dodawanie/Odejmowanie (stały przecinek)	20,1 33,4	2,9 6,5	8,1 13,3	2,1...2,6	1,5 2,0	1,6 1,9	0,7 1,9
• Dodawanie/Odejmowanie (zmienny przecinek, liczby krótkie)	57,3 70,8	13,2 15,5	12,8 16,3	4,0...5,0	4,5 5,0	2,8 3,7	1,0 1,9
• Dodawanie/Odejmowanie (zmienny przecinek, liczby długie)	84,2 94,3	14,9 17,8	19,9 25,9	4,2...6,3	4,1 5,9	3,3 4,2	1,1 1,8
• Mnożenie (stały przecinek)	338...349	23,0 27,6	9,6 14,1	18,5...9,3	10,0 11,0	6,7 7,6	2,0 2,2
• Mnożenie (zmienny przecinek, liczby krótkie)	411...413	33,9 36,3	31,9 37,5	13,1...13,2	8,5 9,0	6,0 6,4	1,8 2,1
• Mnożenie (zmienny przecinek, liczby długie)	1100 1200	52,6 54,7	134...140	27,3 28,7	18,0 19,0	11,3 12,1	3,0 3,1
• Dzielenie (stały przecinek)	389...398	62,7 65,0	108...112	17,5 17,7	14,5 15,2	10,6 11,4	8,2 8,5
• Dzielenie (zmienny przecinek, liczby krótkie)	389...399	50,2 52,4	56,4 58,7	13,8 14,5	12,5 13,2	7,6 8,4	5,8 5,9
• Dzielenie (zmienny przecinek, liczby długie)	2140 2150	125...128	212...218	46,6 48,2	28,5 29,3	15,2 16,1	10,5 10,8

ZESTAWIENIE A: c.d.

1	2	3	4	5	6	7	8
• Imo krótkie operacje	10,2 25,3	4,1 8,6	10,2 19,3	2,5 4,0	1,5 2,0	2,4 3,5	0,8 1,7
Pamięć operacyjna							
• Pojemność (k bajtów)	64 256	128 512	128...512	256 1024	512	256 1024	256 1024
• Czas cyklu (µs)	2,0	2,0	1,25	1,20	1,25	1,35	1,25
• Czas dostępu (µs)	1,0	1,0	0,75	0,66	0,75	0,45	0,95
• Szerokość dostępu bajtów	2	4	4	4	4	8	8
• Maksymalna szybkość przesyłania (Megabajtów/s)	1,0	2,0	3,2	3,3	3,2	17,7	8,5
• Ochrona pamięci	Odczyt/ /Zapis	Odczyt/ /Zapis	Odczyt/ /Zapis	Odczyt/ /Zapis	Odczyt/ /Zapis	Zapis	Odczyt/ /Zapis
System wejścia/ wyjścia							
Kanały selekto- rowe:							
• Maksymalna liczba kanałów	2	2	3	3	3	6	6
• Maksymalna szybkość przesyłania (k bajt/s)	300	500	800	1030	800	1300	1250
• Maksymalna liczba dołączonych jednostek sterujących	8	10	8	8	10	10	10
• Maksymalna liczba adresowanych urządzeń wejścia/wyjścia	256	256	256	256	256	256	256
Kanały multipleksorowe							
• Maksymalna liczba kanałów	1	1	1	1	1	1	1
• Maksymalna przepustowość (k bajt/s) w trybie multipleksorowym	20	50	40	110	70	50	110
• w trybie blokowym	140	300	300	240	350	720	180
• Maksymalna liczba dołączonych jednostek sterujących	8	16	8	10	10	10	50
• Maksymalna liczba adresowanych urządzeń wejścia/wyjścia	224	248	256	256	256	256	256

UWAGI:

- 1) Jednostka centralna JS 2620 nie jest już produkowana.
- 2) W NRD są aktualnie stosowane jednostki centralne JS 2610, JS 2620 i JS 2640.
- 3) W przyszłości będą stosowane jednostki centralne JS 2612 w ograniczonej liczbie i JS 2622.

ZESTAWIENIE B: Jednostki centralne JS 1010, JS 1012 i JS 1021

Jednostka centralna	JS 2610	JS 2612	JS 2621
Kraj produkujący	WRL	WRL	CSRS
Procesor centralny			
Lista instrukcji (liczba instrukcji)	86	109	71
Sposób sterowania	Mikroprogramowy	Mikroprogramowy	Mikroprogramowy
Pamięć mikroprogramów:			
• Pojemność (liczba słów)	512...2048	8192	3072
• Czas cyklu (ns)	300	400	250
• Długość słowa (liczba bitów)	16	16	72
Czasy wykonywania wybranych instrukcji:			
• Dodawanie/Odejmowanie (stały przecinek)	2,1...3,1	2,1...2,9	20,3...30,1
• Mnożenie (stały przecinek)	8,0...9,0	8,0...9,0	80...120
• Dzielenie (stały przecinek)	9,0...10,0	9,0...10,0	145...155
• Inne krótkie operacje	2,2...3,2	2,2...3,0	14,3...15,4
Pamięć operacyjna			
• Pojemność (k bajt)	8...64	8...64	16...64
• Czas cyklu (μs)	0,8	1,0	2,0
• Czas dostępu (μs)	0,4	0,5	1,0
• Szerokość dostępu (bajt)	2	2	1
• Maksymalna szybkość przesyłania (Mogabajt/s)	2,5	2,0	0,5
• Ochrona pamięci	zapis	zapis	brak
System wejścia/wyjścia			
Kanale selektorowe:			
• Maksymalna liczba kanałów		Kanał specjalny (Minibus)	2
• Maksymalna szybkość przesyłania (k bajt/s)			236
• Maksymalna liczba dołączanych jednostek sterujących			10
• Maksymalna liczba adresowanych urządzeń wejścia/wyjścia			160
Kanale multipleksorowe:		Adapter multipleksorowy w kanale specjalnym	1
• Maksymalna liczba			
• Maksymalna szybkość przesyłania (k bajt/s)			
w trybie multipleksorowym	40	45	27
w trybie blokowym	200	120	210
• Maksymalna liczba dołączanych jednostek ster.	8	8	10
• Maksymalna liczba adresowanych urządzeń wo/wy	256	239	136

lejujących stanach. Gdy bit zabezpieczenia pamięci równa się 0, zezwala się zawsze na dostęp do adresowanych danych. Bit kontrolny uzupełnia liczbę bitów równych 1 w każdym półsłowie do nieparzystej. Jeśli podczas odczytu półsłowa wykryty jest błąd parzystości - następuje przerwanie programu.

Oprócz głównej pamięci operacyjnej istnieje jeszcze pomocnicza pamięć operacyjna składająca się z co najmniej 16, a co najwyżej ze 128 szesnastobitowych rejestrów, które są potrzebne do organizacji programu, ale również mogą służyć programiście do przechowywania informacji.

Storowanie wejściom/wyjściom

Zasadniczą część sterowania wejściem/wyjściem stanowi zintegrowany adapter, który przejmuje funkcje jednostki sterującej. Ogółem może być podłączonych do 12 adapterów tego rodzaju. Są one połączone z jednej strony z procesorem centralnym przez specjalny kanał (tzw. minibus). Z drugiej strony adaptery są sprzężone z urządzeniami zewnętrznymi. W celu przyłączenia urządzeń, które mogą współpracować tylko z interfejsem JS EMC, opracowano specjalny adapter, który realizuje funkcje interfejsu wejścia/wyjścia JS EMC. Adapter ten umożliwia dołączenie zarówno takich urządzeń, które pracują w trybie multipleksorowym, jak również urządzeń, które mogą transmitować dane tylko w trybie blokowym. Organizacja wejścia-wyjścia jest przedstawiona schematycznie na rysunku. Ramową konfigurację sprzętu z pamięcią operacyjną o pojemności 16k bajtów pokazano jako podstawową konfigurację modelu JS 1010.

Jednostka centralna JS 2612

Jednostka centralna JS 2612 powstała w ramach modernizacji JS EMC. Jest ona rozwiniętą wersją jednostki JS 2610 i jest z nią wymienna pod względem programów i danych. Jednostka centralna JS 2612 przewidziana jest do obliczeń w zadaniach sterowania, zdalnego przetwarzania danych, przetwarzania danych w czasie rzeczywistym oraz przetwarzania danych komercyjnych. Poza tym, analogicznie do modelu JS 2610, może być dołączona do innych komputerów JS jako komputer satelitarny.

Procesor centralny

Lista rozkazów jednostki JS 2612 w porównaniu z JS 2610 jest rozszerzona o rozkazy arytmetyki dziesiętnej i wykonywanie operacji logicznych. Format, kod i budowa rozkazu są identyczne z analogicznymi rozkazami innych jednostek JS EMC, przez co obok wykonywania rozkazów jednoadresowych możliwe jest także wykonywanie rozkazów, w których spacyfikowane są dwa operandy. Równocześnie, w stosunku do jednostki centralnej JS 2610, zaprojektowano zwiększenie pojemności pamięci mikroprogramów z 2k na 8k słów.

Ważną cechą procesora, z punktu widzenia wydajności jest jego wyposażenie w szybkie rejestry zbudowane na elementach półprzewodnikowych, umożliwiające szybkie zapamiętanie wyników działania mikrorozkazu.

Pamięć operacyjna

Pamięć operacyjna EMC JS 2612 zachowuje podstawowe elementy konstrukcji pamięci operacyjnej modelu JS 2610. Łatwe rozszerzenie jest osiągnięte przez dołączanie bloków pamięci po 8k bajtów każdy. Układy dostępu do pamięci operacyjnej obsługują drogi dostępu, do których mogą być przyłączone jednostki przetwarzające, z których jedna pracuje jako jednostka centralna, a pozostałe jako jednostki transmisji danych lub kanały bezpośredniego dostępu. Określenie kolejności dostępu jednostki funkcjonalnej do pamięci następuje wg zasady stałego priorytetu. Jednostka centralna ma najniższy priorytet.

Storowanie wejściem/wyjściem

Storowanie wejściem/wyjściem jednostki centralnej JS 2612 realizowane jest przez 32 zintegrowane adaptery, przez które następuje przyłączanie określonych urządzeń wejścia/wyjścia. W podstawowym wyposażeniu modelu JS 2612 przewidziane są wbudowane bloki sterowania dla konsoli operatora i dla pamięci dyskowej ze stałymi głowicami JS 5060. Dodatkowym urządzeniem jest wbudowany adapter do przyłączenia pamięci z wymiennymi dyskami JS 5052, pamięci na taśmie magnetycznej JS 5017-02, jak również dla synchronicznej i asynchronicznej transmisji danych.

Blok sterowania do synchronizacji transmisji danych może obsługiwać nie więcej niż 8 półduplexowych albo 4 duplexowe łącza z szybkością transmisji danych od 600 do 20000 bitów/s i dla asynchronicznej transmisji danych nie więcej niż 64 duplexowe łącza z szybkością transmisji danych 50-1200 bitów/s. Poza tym przewidziane jest przyłączenie urządzeń wejścia/wyjścia, dostosowanych do interfejsu wejścia/wyjścia JS EMC.

Jednostka centralna JS 2620

Jednostka centralna JS 2620 jest aktualnie najmniejszą jednostką JS EMC, całkowicie kompatybilną funkcjonalnie, programowo i pod względem danych, z pozostałymi jednostkami JS. Przeznaczona jest ona do rozwiązywania różnorodnych zadań, przede wszystkim jednak do przetwarzania danych komercyjnych. Ponadto może znaleźć zastosowanie także w obliczeniach naukowych, jak również w zautomatyzowanych systemach zarządzania.

Procesor centralny

Konstrukcja procesora centralnego charakteryzuje się, podobnie jak pozostałych małych i średnich jednostek JS EMC, efektywnym zastosowaniem sterowania mikroprogramowego. Blok sterowania mikroprogramowego JS 2620 zawiera: pamięć stałą do przechowywania mikroprogramów, rejestr mikrorozkazów oraz układy przelączające do tworzenia adresów mikrorozkazów.

Struktura mikrorozkazu umożliwia zakodowanie oraz wykonanie operacji dwuargumentowych i jednocześnie sterowanie pamięcią, jak również tworzenie adresu następnego mikrorozkazu i reakcję na zaistniałe warunki logiczne. Operacje te są wykonywane w czasie trwania jednego cyklu maszynowego. Jednostka arytmetyczno-logiczna procesora centralnego może wykonywać wiele różnorodnych mikrooperacji elementarnych, takich jak iloczyn logiczny, suma logiczna, różnica symetryczna i ich modyfikacje, przesunięcia o jedną i o 4 pozycje w lewo niezbędne do normalizacji liczb zmiennoprzecinkowych.

Pamięć operacyjna

Blok pamięci modelu JS 2620 produkowany w trzech podanych niżej wariantach:

Tabela 1

Pojemność pamięci w bajtach				
Jednostka pamięci	Pamięć operacyjna	Pamięć lokalna	Pamięć kanału multipleksorowego	Liczba modułów
JS 3220-1	64k	256	768	1
JS 3220-2	128k	256	1,536	2
JS 3220-3	256k	256	1,536	4

Pamięć lokalna zawiera rejestry uniwersalne i zmiennoprzecinkowe, rejestry diagnostyczne i pamięć słów sterujących podkanału niezbędnych do pracy kanału multipleksorowego. Przydział cykli pamięci operacyjnej procesorowi centralnemu i kanałom następuje w JS 2620 dynamicznie. Wyższy priorytet otrzymuje system wejścia/wyjścia przed procesorem centralnym, a w ramach systemu wejścia/wyjścia kanały selektorowe mają wyższy priorytet niż kanał multipleksorowy.

Kanały

Kanał multipleksorowy należy do standardowego wyposażenia modelu. Jest on sterowany mikroprogramowo, tzn. sterowanie mikroprogramowe procesora centralnego jest wykorzystane do sterowania transmisją danych. Liczba podkanałów, w zależności od pojemności pamięci operacyjnej wynosi od 40 do 104, liczba adresowanych urządzeń od 160 do 224. Dla poszczególnych pamięci kształtuje się jak w zestawieniu A.

Jednostka centralna JS 2620 może być wyposażona w jeden lub dwa kanały selektorowe. Kanały selektora są logicznie i fizycznie samodzielnymi jednostkami funkcjonalnymi, które równolegle i niezależnie od pracy procesora centralnego mogą przeprowadzać transmisję danych. Jeżeli obydwa kanały selektorowe i kanał multipleksorowy pracują jednocześnie, zmniejsza się maksymalna prędkość

Tabela 1a

Pojemność pamięci operacyjnej	Liczba podkanałów	Liczba adresowanych urządzeń przyłączana do kanałów		
		Razem	w tym w podkanałach nie dzielonych	w tym w podkanałach dzielonych
64k bajtów	40	160	32	128
128k bajtów	104	224	96	128
256k bajtów	104	224	96	128

transmisji danych każdego kanału selektorowego do 200k bajtów/s. Elektronika kanałów znajduje się we wspólnej szafie z procesorem centralnym.

Jednostka centralna JS 2621

Jednostka centralna JS 2621 jest najbardziej wydajna wśród małych jednostek JS EMC. Jest ona przewidziana przede wszystkim do rozwiązywania zadań w przemyśle, handlu i zarządzaniu.

W głównych założeniach logiczno-funkcjonalnych jednostka centralna JS 2621 jest zgodna z innymi jednostkami JS i jest wymienna pod względem programów i danych. Tylko w kilku detalach, jak np. repertuar rozkazów, kody operacji dla niektórych rozkazów, system przerw - różni się od pozostałych modeli.

Procesor centralny

W procesorze centralnym modelu JS 2621 jest realizowanych 71 rozkazów; nie przewidziano rozkazów arytmetyki zmiennoprzecinkowej, ochrony pamięci jak również niektórych rozkazów arytmetyki binarnej działających na słowach. Przyjęty sposób sterowania systemem przerw wymaga poczwórnej liczby rejestrów ogólnych i uniwersalnych procesora centralnego. 16 rejestrów przyporządkowane jest każdemu z poziomów przerw. Przyjęcie sygnału przerwania powoduje obok wymiany słowa stanu programu także dołączenie rejestrów uniwersalnych, przyporządkowanych danemu poziomowi przerw.

Pamięć operacyjna

Pamięć operacyjna JS 2621 jest dostępna na rynku w pojemnościach 16k bajtów, 32k bajtów, 64k bajtów. Szerokość dostępu wynosi 1 bajt; przy pobieraniu rozkazu formatu SS do rejestru rozkazów konieczny jest 6-krotny dostęp do pamięci operacyjnej. Pamięć operacyjna pozwala więc osiągnąć tylko relatywnie małą szybkość wykonywania operacji. Pamięć operacyjna JS 2621 nie jest wyposażona w urządzenia zabezpieczające przed niedozwolonym zapisem lub nieuprawnionym odczytem.

Kanały

Jednostka centralna JS 2621 wyposażona jest w kanał multipleksorowy i maksymalnie 2 kanały selektorowe. Kanał multipleksorowy i kanały selektorowe są sterowane mikroprogramowo tzn. sterowanie mikroprogramowo procesora centralnego jest także wykorzystywane do sterowania transmisją danych. Procesor centralny i kanał multipleksorowy mogą pracować jednocześnie. Procesor centralny otrzymuje jednak sterowania tylko wtedy, kiedy w kanale multipleksorowym zaistnieje przerwa między transmisją kolejnych bajtów, tzn. jest to możliwe tylko podczas pracy w trybie multipleksorowym. Jeżeli kanał multipleksorowy pracuje w trybie blokowym następuje zawieszenie pracy procesora centralnego. Podczas pracy kanałów selektorowych procesor centralny nie może wykonywać żadnych operacji, również kanał multipleksorowy nie może podjąć operacji transmisji danych. Działanie kanałów selektorowych zorganizowane jest w ten sposób, że podczas pracy jednego kanału selektorowego niemożliwe jest podjęcie pracy przez drugi kanał.

Jednostka centralna JS 2622

Jednostka centralna JS 2622 jest rozwinięciem modelu JS 2620. Baza konstrukcyjno-technologiczna JS 2620 uległa tylko niewielkim zmianom. Nowością jest przede wszystkim zastosowanie rozszerzonego zestawu układów podstawowych, co pozwala 6-7-krotnie zwiększyć wydajność w stosunku do JS 2620. Istotnie rozszerza to zakres zastosowań.

Procesor centralny

W procesorze centralnym wprowadzono istotne zmiany w stosunku do JS 2620:

- podwójono liczbę pozycji (32 bity) jednostki arytmetyczno-logicznej,
- skrócono cykl pracy na 250 ns (w JS 2620 = 500 ns),
- skrócono czas cyklu pamięci mikroprogramów z 1000 ns na 500 ns,
- wbudowano pamięć lokalną jako samodzielny blok w centralnym procesorze.

W przeciwieństwie do JS 2620, gdzie pamięć lokalna jest pamięcią rdzeniową (stanowi część pamięci operacyjnej) zastosowano pamięć półprzewodnikową o cyklu 250 ns.

Pamięć operacyjna

Pojemność pamięci operacyjnej jest rozszerzona w stosunku do JS 2620 z 256k bajtów do 512k bajtów. Jednocześnie z poszerzeniem jednostki arytmetyczno-logicznej zwiększono szerokość dostępu do pamięci operacyjnej z 2 na 4 bajty. Pamięć operacyjna zestawiona jest z modułów o pojemności 64k bajtów. W jednej szafie umieszcza się od dwóch do czterech takich bloków, w ten sposób przy pojemności pamięci 128k bajtów i 256k bajtów wystarcza jedna szafa, a przy pojemności 512k bajtów potrzebne są dwie.

Kanały

W systemie wejścia/wyjścia JS 2622 dokonano pewnych ulepszeń w stosunku do JS 2620. Przy zachowaniu liczby kanałów, drogą zmian logiczno-funkcjonalnych podniesiono prędkość transmisji danych w kanałach selektorowych z 300k bajtów/s do 600k bajtów/s, a w kanale multipleksorowym z około 20k bajtów/s do 80k bajtów/s w trybie multipleksorowym i z 140k bajtów/s do około 400k bajtów/s w trybie blokowym. Dalszą innowacją jest możliwość podłączenia do kanału multipleksorowego do 16 jednostek sterujących (w JS 2620 do 8). W kanale multipleksorowym zainstalowano 128 podkanałów, w tym 120 niedzielonych i 8 dzielonych.

Jednostka centralna JS 2630

Jednostka centralna JS 2630 ze względu na swoją wydajność, pojemność pamięci operacyjnej i parametry systemu wejścia/wyjścia zaliczana jest do średniej klasy jednostek JS EMC. Jej wydajność umożliwia rozwiązywanie różnorodnych zadań naukowo-technicznych, ekonomicznych i z innych dziedzin przetwarzanie informacji.

Procesor centralny

Centralny procesor JS 2630 jest sterowany mikroprogramowo. Mikroprogramy umieszczone są w 32k bajtowej pamięci stałej. Czas dostępu wynosi 400 μ s.

Jednostka centralna JS 2620 może być wyposażona, podobnie jak inne jednostki JS EMC w układy sterowania bezpośredniego, które pracują niezależnie od urządzeń wejścia/wyjścia. Przez interfejs sterowania bezpośredniego przesyłany jest pojedynczy bajt danych za pomocą sygnałów przerwania zewnętrznego i dwóch sygnałów sterowania bezpośredniego, będących w dyspozycji rozkazów PISZ BEZPOŚREDNIO I CZYTAJ BEZPOŚREDNIO.

Pamięć operacyjna

Pamięć operacyjna zestawiana jest z modułów pamięci o pojemności 128k bajtów. Możliwe jest umieszczenie maksimum 2 modułów pamięci z pojemnością 128k bajtów w jednej szafie pamięci. Przy maksymalnej pojemności pamięci do 512k bajtów wymagane są dwie szafy pamięci. Dla każdej szafy pamięci konieczna jest szafa z urządzeniami zasilania w energię elektryczną.

Kanały

Panele kanału multipleksorowego i panele kanałów selektorowych są montowane w oddzielnej jednostce (szafie).

Kanał multipleksorowy należy do standardowego wyposażenia modelu. Istnieje możliwość przyłączenia od jednego do trzech kanałów selektorowych. Każdy z tych kanałów stanowi samodzielną logiczną

nie i fizycznie jednostkę funkcjonalną. Zapewniona jest także ich równoległa i niezależna praca w stosunku do procesora centralnego.

Liczba podkanałów kanału multipleksorowego wynosi 128 przy pojemności pamięci 128k bajtów i 256k bajtów, a 256 przy pojemności pamięci 512k bajtów. Spośród 128 podkanałów pierwsze 8 może być dzielone, pozostałe 120 są podkanałami niedzielonymi.

Jednostka centralna JS 2632

Jednostka centralna JS 2632 jest rezultatem modernizacji, którą w PRL podjęto w stosunku do modelu JS 2630. Szybkość operacji (230000 Op/s, wg mieszanki JS EMC modyfikacja I) jest prawie trzykrotnie wyższa od modelu JS 2630. Dlatego też model ten wypełnia lukę w JS EMC między modelami JS 2630 (60000 Op/s) i JS 2640 (380000 Op/s). Model JS 2632 przeznaczony jest do rozwiązywania różnorodnych zadań naukowo-technicznych i ekonomicznych oraz specjalnych w autonomicznych ośrodkach obliczeniowych, jak również w systemach zdalnego przetwarzania danych.

Procesor centralny

W modelu JS 2632 jest w zasadzie zachowana podstawowa struktura procesora centralnego JS 2630. Lepsze parametry techniczno-ekonomiczne uzyskano głównie w wyniku zastosowania rozszerzonego asortymentu układów podstawowych, jak również przez nową technologię okablowania.* Pozwoliło to na łatotne zmniejszenie liczby podzespołów scalonych i połączeń stykowych oraz na podniesienie niezawodności procesora centralnego.

Procesor centralny JS 2632 składa się z pamięci mikroprogramów, pamięci lokalnej z czasem cyklu 300 ns, bloku arytmetyczno-logicznego, urządzenia sterującego, co gwarantuje wzajemną współpracę poszczególnych składników centralnego procesora, jak również współpracę z kanałami oraz z pulpitem sterowania. Czas trwania cyklu maszynowego wynosi 300 ns.

Pamięć operacyjna

Pamięć operacyjną zestawia się z logicznie niezależnych bloków, przyłączanych do procesora centralnego przez wspólny interfejs. Jeden blok pamięci składa się z 16 jednakowych jednostek (każda po 16k bajtów) i adaptera pamięci, który zawiera układy sterujące pamięcią, jak również 128 bajtów pamięci kluczy ochrony pamięci. Przesyłanie danych między blokami pamięci operacyjnej a procesorem centralnym, odbywa się przez szyny o szerokości 4 bajtów.

Pamięć operacyjna jest sterowana sprzętowo. Pamięć operacyjna może być rozbudowana w zależności od zastosowania do następujących pojemności: 256, 512, 768 i 1024k bajtów.

Kanały

Do kanałów JS 2632 należą: kanał multipleksorowy, 3 kanały selektorowe i koordynator kanałów. Kanał multipleksorowy ma sterowanie zintegrowane ze sterowaniem procesora centralnego. Wszystkie operacje kanałowe wykorzystują wspólną pamięć mikroprogramów. Wykonanie dowolnej mikroprogramowej operacji w kanale wymaga przerwania mikroprogramu procesora centralnego na określony czas. Urządzenia wejścia/wyjścia w kanale multipleksorowym mogą być dołączane w 256 niedzielonych albo 8 dzielonych i 120 niedzielonych podkanałach.

Kanały selektorowe mają sterowanie częściowo zintegrowane ze sterowaniem procesora centralnego. Umożliwiają one skompletowanie słowa danych niezależnie od pracy procesora centralnego. Wymiana danych z pamięcią operacyjną, jak również pobieranie, zapamiętanie i uaktualnianie słów sterujących jest nadal sterowane mikroprogramowo. Umożliwia to częściowo współbieżne wykonywanie operacji wejścia/wyjścia.

Współpraca kanałów z procesorem centralnym i pamięcią operacyjną odbywa się za pośrednictwem koordynatora kanałów. Przymuje on instrukcje wejścia/wyjścia z procesora centralnego, a z drugiej strony żądania dostępu do pamięci z kanałów. Układy logiczne koordynatora wybierają spośród żą-

* Informacja ta nie jest zupełnie ścisła. Podstawową zmianą technologiczno-konstrukcyjną było zastosowanie wielowarstwowych obwodów drukowanych. Poza tym w stosunku do modelu 2630 wprowadzono dość poważne zmiany strukturalne, między innymi nowy repertuar mikroinstrukcji (przyp. red.)

dań kanałów zgłoszenie o najwyższym priorytecie i przesyłają je do sterowania procesora centralnego. Procesor centralny przyjmuje takie zgłoszenie i po 600 ns zaczyna realizować mikroprogram obsługi kanałów. Po zakończeniu mikroprogramu obsługi kanału podejmowany jest znowu przerwany mikroprogram procesora centralnego pod warunkiem, że nie ma dalszych zgłoszeń kanałów.

Jednostka centralna JS 2633

Jednostka centralna JS 2633 powstała w ZSRR jako wynik dalszego rozwoju modelu JS 2630, tj. modernizacji bazy technologiczno-konstrukcyjnej. Zakres zastosowań rozszerza się jak w jednostkach JS 2630 i JS 2632 na obliczenia naukowo-techniczne oraz handlowe.

Procesor centralny

Procesor centralny modelu JS 2633 rozporządza własnym sterowaniem mikroprogramowym, niezależnym od kanałów. Gwarantuje to równoległą pracę procesora centralnego i kanału multipleksorowego oraz kanału selektorowego. Pamięć mikroprogramów procesora centralnego jest przeznaczona do przechowywania mikroprogramów, które obok realizacji 143 rozkazów wykonują także procedury pulpitu operatora, uruchamiania i diagnozowania.

Pamięć operacyjna

Pamięć operacyjna modelu JS 2633 ma pojemność 512k bajtów. Zarówno w konstrukcji jak i w parametrach technicznych jest analogiczna do JS 2630. Cykl pamięci operacyjnej odpowiada czterem cyklom maszynowym; operand jest wybierany z pamięci w trzecim cyklu maszynowym. Czas cyklu szynowego jest określony przez czas cyklu i czas dostępu pamięci mikroprogramów.

Kanały

System wejścia/wyjścia JS 2633 zawiera analogicznie do JS 2632 kanał multipleksorowy, 3 kanały selektorowe i kordynator kanałów. Koordynator kanałów przejmując także funkcje przesyłania rozkazów wejścia/wyjścia z centralnego procesora do kanałów, realizuje procedury interfejsowe dołączania jednostki sterującej do kanału. Do realizacji tych zadań służy własny mikroprogram niezależny od centralnego procesora (pamięć o pojemności 2048 słów, czasie cyklu 300 ns, długości słowa 64 bitów). Kanał multipleksorowy wykorzystuje sterowanie mikroprogramowe koordynatora kanałów. Praca kanałów selektorowych sterowana jest sprzętowo celem zwiększenia przepustowości.

Jednostka centralna JS 2640

Jednostka centralna JS 2640 ze względu na wydajność przetwarzania, pojemność pamięci operacyjnej i efektywność sterowania wejścia/wyjścia jest zaliczana do średniej klasy maszyn JS. Jest ona drugą co do wielkości jednostką centralną w JS EMC. Może być zastosowana do rozwiązywania licznych skomplikowanych zadań naukowo-technicznych i ekonomicznych. Istnieje również możliwość jej zastosowania jako komputera nadrzędnego w systemie komputerowym.

Procesor centralny

Dużą wydajność procesora centralnego osiągnięto przez zastosowanie zrównoleżenia operacji. Można tu wymienić:

- nakładające się dostępy do różnych modułów pamięci operacyjnej,
- niezależny jednoczesny dostęp do pamięci operacyjnej i rejestrów (uniwersalnych i zmienno-przebiegowych),
- nakładanie się fazy pobierania rozkazów z fazą wykonywania rozkazów.

Umożliwia to np. uszeregowanie faz wykonawczych, kolejno następujących po sobie rozkazów bez przerw czasowych. Ten równoczesny sposób pracy staje się możliwy m.in. przez zastosowanie niezależnego sterowania dla "nakładających się" jednostek funkcjonalnych. Tak więc na przykład pobieranie rozkazów i współpraca z pamięcią główną są sterowane sprzętowo, natomiast wykonanie rozkazów a zatem działanie jednostki przetwarzającej jest sterowane mikroprogramowo.

Pamięć operacyjna

Możliwe konfiguracje pamięci operacyjnej

Pojemność 256k bajtów	- 2 moduły po 128k bajtów
Pojemność 512k bajtów	- 4 moduły po 128k bajtów lub 2 moduły po 256k bajtów
Pojemność 1024k bajtów	- 4 moduły po 256k bajtów

W jednej szafie znajdują się po dwa moduły pamięci, również urządzenia zasilające moduły znajdującą się w tej samej szafie. Współpraca jednostki centralnej JS 2640 z pamięcią operacyjną jest tak zorganizowana, że w jednym cyklu maszynowym (450 ns) może być uruchomiony nie więcej niż jeden moduł pamięci. Chociaż każdy moduł pamięci ma czas cyklu około 1,1 μ s, to może on wznowić pracę nie wcześniej niż po 1,35 μ s. Aby móc zwracać się do pamięci w każdym cyklu maszynowym konieczne jest aby trzy kolejno następujące po sobie zgłoszenia do pamięci dotyczyły trzech różnych modułów. To założenie jest spełniane z dużym prawdopodobieństwem przy zachowaniu zasady "adresowania poziomego". Przy tym kolejne słowa podwójne umieszczone są w różnych modułach pamięci. Z tego wynika, że następny dostęp do pamięci z dużym prawdopodobieństwem nie będzie dotyczył tego samego modułu co poprzedni. W innym wypadku żądanie nie będzie tak długo przyjmowane, aż odpowiedni moduł będzie ponownie wolny.

Opisany stan rzeczy pozwala stwierdzić, że praca trzech modułów pamięci jest możliwa dla pamięci złożonej z 4 modułów o jednakowej pojemności (3-krotne nakładanie). W konfiguracji, w której pamięć operacyjna składa się tylko z 2 modułów o jednakowej pojemności, tylko te dwa moduły mogą pracować równolegle (2-krotne nakładanie).

Kanał multipleksorowy

Kanał multipleksorowy jest logicznie i fizycznie samodzielną jednostką funkcjonalną. Należy on do standardowego wyposażenia jednostki centralnej JS 2640. Liczba podkanałów kanału multipleksorowego zależy od pojemności pamięci operacyjnej. Przy pojemności 256k bajtów liczba podkanałów wynosi 128; przy pojemności 512k albo 1024k bajtów po 256 podkanałów. Liczba adresowanych urządzeń w kanale multipleksorowym wynosi 256. Spośród 128 podkanałów pierwsze 8 jest dzielonych a pozostałe 120 są niedzielone.

Kanały selektorowe

Kanały selektorowe są logicznie i fizycznie samodzielnymi jednostkami funkcjonalnymi, które umieszczone są w szafie jednostki centralnej. Maksymalna szybkość transmisji danych wynosi dla każdego kanału selektorowego 1,3 mln megabajtów/s. Przy równoległej pracy wszystkich kanałów selektorowych prędkość transmisji danych zmniejsza się aż do częstotliwości, z którą poszczególne kanały selektorowe zgłaszają żądania dostępu do pamięci operacyjnej. W zależności od priorytetu dostępu do pamięci operacyjnej kanały selektorowe dzielą się na trzy grupy:

- I - kanał selektorowy 1;
- II - kanały selektorowe 2 i 3;
- III - kanały selektorowe 4, 5 i 6.

Przedstawiony układ priorytetów pozwala osiągnąć przy równoległej pracy wszystkich kanałów selektorowych następujące prędkości transmisji danych między pamięcią operacyjną a jednostką sterującą.

Kanał selektorowy grupy I:	1,3 mln bajtów/s;
Kanały selektorowe grupy II:	po 550k bajtów/s;
Kanały selektorowe grupy III:	po 300k bajtów/s.

Jeżeli przyłączonych jest mniej niż 6 kanałów selektorowych albo nie wszystkie pracują równocześnie wtedy podnosi się szybkość transmisji danych dla kanałów selektorowych w tej z grup II lub III w której brak co najmniej jednego kanału albo co najmniej jeden kanał nie pracuje. Jeśli brakuje jednego kanału grupy II, drugi kanał tej grupy może osiągnąć szybkość transmisji danych 1100k bajtów/s, jeśli brak dwóch kanałów grupy III - trzeci kanał tej grupy osiąga szybkość 900k bajtów/s.

Jednostka centralna JS 2650

Jednostka centralna JS 2650 jest najbardziej wydajną jednostką centralną Jednolitego Systemu EMC. Jest przystosowana do rozwiązywania różnorodnych zadań naukowo-technicznych, handlowych oraz specjalnych zadań przetwarzania informacji.

Procesor centralny

Cechą charakterystyczną budowy centralnego procesora jednostki centralnej JS 2650 jest sterowanie całkowicie sprzętowe. Ten rodzaj sterowania, za pomocą którego są zrealizowane wszystkie operacje jednostki centralnej, zapewnia dużą szybkość przetwarzania. Jest ona w istocie zależna od szybkości zastosowanych układów podstawowych i częstotliwości zegara. Wykonywanie rozkazów w procesorze centralnym następuje analogicznie jak w modelu JS 2640, z nakładaniem operacji, to znaczy, że kilka następujących po sobie rozkazów znajduje się jednocześnie w różnych fazach przetwarzania. Dlatego jest możliwe nakładanie się faz pracy przygotowania rozkazu i wykonania rozkazu.

Pamięć operacyjna

Organizacja pamięci operacyjnej oparta jest na modułach o pojemności 128k bajtów. Dwa moduły po 128k bajtów umieszczane są w jednej szafie pamięci, której pojemność wynosi zatem 256k bajtów. Pamięć operacyjna o pojemności 512k bajtów jest realizowana w postaci 2 szaf, każda po 256k bajtów; pojemność 1024k bajtów wymaga 4 szaf, każda po 256k bajtów. Ponadto dla każdej szafy niezbędna jest szafa zawierająca urządzenia zasilające.

Współpraca z pamięcią jest tak zorganizowana, że możliwa jest równoległa praca dwóch modułów (dwukrotne nakładanie). Dwukrotne nakładania umożliwiają już moduły pamięci po 128k bajtów - umieszczone w jednej szafie, adresowane poziomo. Sąsiednie słowa podwójne umieszczone są w różnych modułach pamięci w tej samej szafie. Zasada poziomego adresowania wynika z założenia o małym prawdopodobieństwie tego, że kolejne zgłoszenie nastąpi do tego samego modułu. Jeżeli tak nie będzie, to realizacja żądania dostępu do pamięci będzie zawieszona do momentu, w którym odpowiedni moduł będzie ponownie wolny. Te właściwości strukturalne pamięci nie powodują żadnych ograniczeń techniczno-programowych, natomiast pozwalają osiągnąć parametry wymagane dla tej klasy jednostek centralnych.

Kanał multipleksorowy

Kanał multipleksorowy jednostki centralnej JS 2650 jest logicznie i fizycznie samodzielną jednostką funkcjonalną. Składa się z typowej szafy zawierającej układy sterowania operacjami transmisji oraz z szafy na urządzenia zasilające.

Kanał multipleksorowy wykazuje w stosunku do kanałów multipleksorowych innych kompatybilnych jednostek centralnych kilka specyficznych właściwości logiczno-funkcjonalnych, które będą poniżej krótko objaśnione. Podczas pracy kanałów multipleksorowych w trybie blokowym urządzenia wejścia/wyjścia pracujące wyłącznie w trybie multipleksorowym nie są sterowane, przez co transmisja danych w trybie blokowym opóźnia ich pracę. Okoliczność ta staje się jeszcze bardziej uciążliwa, gdy urządzeniami pracującymi w trybie blokowym są urządzenia charakteryzujące się niedużą prędkością transmisji danych, ponieważ one stosunkowo długo blokują kanał.

Chociaż przyłączenie tych urządzeń do kanału selektorowego obciążałoby go tylko w niewielkim stopniu to utrudniałoby pracę urządzeń o dużej szybkości transmisji danych często używanych przez system. Uwzględniając ten stan rzeczy, kanał multipleksorowy jest tak zaprojektowany, że umożliwia równoległą pracę urządzeniom pracującym w trybie multipleksorowym oraz urządzeniom pracującym sposobem blokowym o średniej prędkości transmisji danych. Kanał multipleksorowy zawiera część multipleksorową, do której może być dołączanych do 10 jednostek sterujących, pracujących w trybie multipleksorowym i urządzenia sterowania dla maksymalnie 4 podkanałów selektorowych, które wprowadzają i wyprowadzają dane w trybie blokowym. Do każdego z 4 podkanałów selektorowych może być dołączonych 10 jednostek sterujących. Przy tym do każdego podkanału selektorowego może być dołączone do 16 urządzeń.

Możliwe konfiguracje kanału multipleksorowego przedstawiono w tabeli 2, natomiast zależność szybkości transmisji danych od konfiguracji kanału multipleksorowego podano w tabeli 3.

Tabela 2

Możliwe konfiguracje kanału multipleksorowego JS 2650

Część multipleksorowa Liczba podkanałów	Część selektorowa Liczba podkanałów selektorowych	Liczba jednostek sterujących przyłączanych do kanału multipleksorowego	Maksymalna liczba dołączanych urządzeń		
			do części multipleksorowej	do podkanałów selektorowych	razem
256	0	10	256	0	256
240	1	20	240	16	256
224	2	30	224	32	256
208	3	40	208	48	256
192	4	50	192	64	256

Tabela 3

Przepustowość kanału multipleksorowego JS 2650

Część multipleksorowa	Pierwszy podkanał selektorowy	Drugi podkanał selektorowy	Trzeci podkanał selektorowy	Czwarty podkanał selektorowy	Ogólna przepustowość kanału multipleksorowego
k/bajtów/s	k bajtów/s	k bajtów/s	k bajtów/s	k bajtów/s	k bajtów/s
110	0	0	0	0	110
85	180	0	0	0	265
65	180	180	0	0	425
40	180	180	180	0	580
30	180	180	180	180	670

Kanały selektorowe

Kanały selektorowe, podobnie jak kanał multipleksorowy, są logicznie i fizycznie samodzielnymi jednostkami funkcjonalnymi. Do jednostki centralnej JS 2650 można dołączyć do 6 kanałów selektorowych. W typowej szafie instalowane są 3 kanały selektorowe, z których każdy zajmuje po 3 panele. Maksymalną konfigurację stanowią dwie tego rodzaju szafy. Poza tym dla każdej z nich wymagana jest szafa zawierająca urządzenia zasilające.

Maksymalna przepustowość kanału selektorowego wynosi 1,25 milionów bajtów/s. Przy jednoczesnej pracy wszystkich kanałów selektorowych szybkość przesyłania danych określona jest przez częstotliwość przyjmowania zgłoszeń od kanałów selektorowych przez pamięć operacyjną.

Adapter - kanał - kanał

Do każdego kanału selektorowego lub multipleksorowego jednostki centralnej JS 2650 może być dołączony adapter kanał-kanał JS 4060, który jest wykonany w formie panelu. Umożliwia to zrealizowanie połączeń kanał-kanał z jakimkolwiek innym kanałem tej samej albo innej jednostki centralnej. Sprzęgając dwa kanały tej samej jednostki centralnej można informacje z jednego obszaru pamięci operacyjnej przesłać w inne miejsce pamięci, przy czym do tej operacji wykorzystany będzie system wejścia/wyjścia, a w procesorze centralnym mogą być jednocześnie wykonywane inne prace. Adapter kanał-kanał wymaga dołączenia jednostki sterującej do każdego z łączonych kanałów.

Lista rozkazów głównych modeli JS EMC

Lp.	Nazwa rozkazu	mniemotoczniczne kody operacji	hexadecymalnie kody operacji	Format rozkazu	operandy
1	2	3	4	5	6
1	DODAJ	A	5A	RX	R1, D2 (X2, B2)
2	DODAJ	AR	1A	RR	R1, R2
3	DODAJ DZIESIĘTNE	AP	FA	SS	D1 (L1, B1) D2 (L2, B2)
4	DODAJ I NORM. ZMP D	AD	6A	RX	R1, D2 (X2, B2)
5	DODAJ I NORM. ZMP D	ADR	2A	RR	R1, R2
6	DODAJ I NORM. ZMP K	AE	7A	RX	R1, D2 (X2, B2)
7	DODAJ I NORM. ZMP K	AER	3A	RR	R1, R2
8	DODAJ BEZ NORM. ZMP K	AU	7E	RX	R1, D2 (X2, B2)
9	DODAJ BEZ NORM. ZMP K	AUR	3E	RR	R1, R2
10	DODAJ BEZ NORM. ZMP D	AW	6E	RX	R1, D2 (X2, B2)
11	DODAJ BEZ NORM. ZMP D	AWR	2E	RR	R1, R2
12	DODAJ PÓLSŁOWO	AH	4A	RX	R1, D2 X2, B2
13	DODAJ LOGICZNE	AL	5E	RX	R1, D2 (X2, B2)
14	DODAJ LOGICZNE	ALR	1E	RR	R1, R2
15	WYKONAJ	EX	44	RX	R1, D2 (X2, B2)
16	DZIEL	D	5D	RX	R1, D2 (X2, B2)
17	DZIEL	DR	1D	RR	R1, R2
18	DZIEL DZIESIĘTNE	DP	FD	SS	D1 (L1, B1), D2 (L2, B2)
19	DZIEL ZMP D	DD	6D	RX	R1, D2 (X2, B2)
20	DZIEL ZMP D	DDR	2D	RR	R1, R2
21	DZIEL ZMP K	DE	7D	RX	R1, D2 (X2, B2)
22	DZIEL ZMP K	DER	3D	RR	R1, R2
23	REDAGUJ	ED	DE	SS	D1 (L, B1) D2 (B2)
24	REDAGUJ I ZNACZ	EDMK	DF	SS	D1 (L, B1), D2 (B2)
25	(CZYTAJ KLUCZ PAMIĘCI)	ISK	09	RR	R1, R2
26	PRZECZYTAJ ZNAK	IC	43	RX	R1, D2 (X2, B2)
27	ROZPAKUJ	UNPK	F3	SS	D1 (L1, B1), D2 (L2, B2)
28	RÓŻNICA SYMETRYCZNA	X	57	RX	R1, D2 (X2, B2)
29	RÓŻNICA SYMETRYCZNA PÓL	XC	D7	SS	D1 (L, B1), D2 (B2)
30	RÓŻNICA SYMETRYCZNA ZNAKÓW	XI	97	SI	D1 (B1) 12
31	RÓŻNICA SYMETRYCZNA	XR	17	RR	R1, R2
32	DZIEL PRZEZ DWA ZMP D	HDR	24	RR	R1, R2
33	DZIEL PRZEZ DWA ZMP K	HER	34	RR	R1, R2
34	(ZATRZYMAJ WE/WY)	HIO	9E	SI	D1 (B1)
35	ZMIEŃ NA POSTAĆ BINARNĄ	CVB	4F	RX	R1, D2 (X2, B2)
36	ZMIEŃ NA POSTAĆ DZIESIĘTNA	CVD	4E	RX	R1, D2 (X2, B2)
37	LADUJ	L	58	RX	R1, D2 (X2, B2)
38	LADUJ	LR	18	RR	R1, R2
39	LADUJ ADRES	LA	41	RX	R1, D2 (X2, B2)
40	LADUJ ZMP D	LD	68	RX	R1, D2 (X2, B2)
41	LADUJ ZMP D	LDR	28	RR	R1, R2
42	LADUJ ZMP K	LE	78	RX	R1, D2 (X2, B2)
43	LADUJ ZMP K	LER	38	RR	R1, R2

1	2	3	4	5	6
44	ŁADUJ PÓLSŁOWO	LHI	48	RX	R1, D2 (X2, B2)
45	ŁADUJ UZUPEŁNIENIE	LCR	13	RR	R1, R2
46	ŁADUJ UZUPEŁNIENIE ZMP D	LCDR	23	RR	R1, R2
47	ŁADUJ UZUPEŁNIENIE ZMP K	LCER	33	RR	R1, R2
48	ŁADUJ WIELOKROTNIĘ	LM	98	RS	R1, R3, D2 (B2)
49	ŁADUJ UJEMNIE	LNR	11	RR	R1, R2
50	ŁADUJ UJEMNIE ZMP D	LNDR	21	RR	R1, R2
51	ŁADUJ UJEMNIE ZMP K	LNER	31	RR	R1, R2
52	ŁADUJ DODATNIO	LPR	10	RR	R1, R2
53	ŁADUJ DODATNIO ZMP D	LPDR	20	RR	R1, R2
54	ŁADUJ DODATNIO ZMP K	LPER	30	RR	R1, R2
55	(ŁADUJ PSW)	LPSW	82	SI	D1 (B1)
56	ŁADUJ I TESTUJ	LTR	12	RR	R1, R2
57	ŁADUJ I TESTUJ ZMP D	LTDR	22	RR	R1, R2
58	ŁADUJ I TESTUJ ZMP K	LTER	32	RR	R1, R2
59	(CZYTAJ BEZPOŚREDNIO)	RDD	85	SI	D1 (B1), 12
60	PRZESUŃ W LEWO ALG. K	SLA	8B	RS	R1, D2 (B2)
61	PRZESUŃ W LEWO ALG. D	SLDA	8F	RS	R1, D2 (B2)
62	PRZESUŃ W LEWO LOG. K	SLL	89	RS	R1, D2 (B2)
63	PRZESUŃ W LEWO LOG. D	SLDL	8D	RS	R1, D2 (B2)
64	ZERUJ I DODAJ	ZAP	F8	SS	D1 (L1, B1), D2 (L2, B2)
65	MNÓŻ	M	5C	RX	R1, D2 (X2, B2)
66	MNÓŻ	MR	1C	RR	R1, R2
67	MNÓŻ DZIESIĘTNIE	MP	FC	SS	D1 (L1, B1), D2 (L2, B2)
68	MNÓŻ ZMP D	MD	6C	RX	R1, D2 (X2, R2)
69	MNÓŻ ZMP D	MDR	2C	RR	R1, R2
70	MNÓŻ ZMP K	ME	7C	RX	R1, D2 (X2, X2)
71	MNÓŻ ZMP K	MER	3C	RR	R1, R2
72	MNÓŻ PÓLSŁOWO	MH	4C	RX	R1, D2 (X2, B2)
73	SUMUJ LOGICZNIE	O	56	RX	R1, D2 (X2, B2)
74	SUMUJ LOGICZNIE POŁA	OC	D6	SS	D1 (L, B1), D2 (B2)
75	SUMUJ LOGICZNIE ZNAKI	OI	96	SI	D1 (B1), 12
76	SUMUJ LOGICZNIE	OR	16	RR	R1, R2
77	PAKUJ	PACK	F2	SS	D1 (L1, B1), D2 (L2, B2)
78	PRZESUŃ W PRAWO ALG. K	SRA	8A	RS	R1, D2 (B2)
79	PRZESUŃ W PRAWO ALG. D	SRDA	8E	RS	R1, D2 (B2)
80	PRZESUŃ W PRAWO LOG. K	SRL	88	RS	R1, D2 (B2)
81	PRZESUŃ W PRAWO LOG. D	SRDL	8C	RS	R1, D2 (B2)
82	(PISZ BEZPOŚREDNIO)	WRD	84	SI	D1 (B1), 12
83	USTAW MASKĘ PROGRAMU	SPM	04	RR	R1
84	PISZ KLUCZ PAMIĘCI	SSK	08	RR	R1, R2
85	USTAW MASKĘ SYSTEMU	SSM	80	SI	D1 (B1)
86	PAMIĘTAJ	ST	50	RX	R1, D2 (X2, B2)
87	PAMIĘTAJ ZMP D	STD	60	RX	R1, D2 (X2, B2)
88	PAMIĘTAJ ZMP K	STE	70	RX	R1, D2 (X2, B2)
89	PAMIĘTAJ PÓLSŁOWO	STH	40	RX	R1, D2 (X2, B2)
90	PAMIĘTAJ WIELOKROTNIĘ	STM	90	RS	R1, R3, D2 (B2)
91	PAMIĘTAJ ZNAK	STC	42	RX	R1, D2 (X2, B2)

1	2	3	4	5	6
92	START WE/WY	SIO	9C	SI	D1(B1)
93	ODEJMIJ	S	5B	RX	RX, D2 (X2, B2)
94	ODEJMIJ	SR	1B	RR	R1, R2
95	ODEJMIJ DZIESIĘTNIE	SP	FB	SS	D1(L1, B1),
96	ODEJMIJ I NORM. ZMP D	SD	6B	RX	R1, D2 (X2, B2)
97	ODEJMIJ I NORM. ZMP D	SDR	2B	RR	R1, R2
98	ODEJMIJ I NORM. ZMP K	SE	7B	RX	R1, D2 (X2, B2)
99	ODEJMIJ I NORM. ZMP K	SER	3B	RR	R1, R2
100	ODEJMIJ BEZ NORM. ZMP K	SU	7F	RX	R1, D2 (X2, B2)
101	ODEJMIJ BEZ NORM. ZMP K	SUR	3F	RR	R1, R2
102	ODEJMIJ BEZ NORM. ZMP D	SW	6F	RX	R1, D2 (X2, B2)
103	ODEJMIJ BEZ NORM. ZMP D	SWR	2F	RR	R1, R2
104	ODEJMIJ PÓLSŁOWO	SH	4B	RX	R1, D2 (X2, B2)
105	ODEJMIJ LOGICZNIE	SL	5F	RX	R1, D2 (X2, B2)
106	ODEJMIJ LOGICZNIE	SLR	1F	RR	R1, R2
107	WYWOŁAJ SUPERWIZOR	SVC	0A	RR	I
108	TESTUJ WE/WY	TIO	9D	SI	D1(B1)
109	(TESTUJ KANAŁ)	TCH	9F	SI	D1(B1)
110	TESTUJ I USTAW	TS	93	SI	D1(B1)
111	TESTUJ MASKI	TM	91	SI	D1(B1), 12
112	PRZENIEŚ POLE CYFR	MVN	D1	SS	D1(L, B1), D2(B2)
113	PRZENIEŚ POLE STREF	MVZ	D3	SS	D1(L, B1), D2(B2)
114	PRZENIEŚ PARAMETR	MVI	92	SI	D1(B1), 12
115	PRZESUŃ I PRZENIEŚ	MVO	F1	SS	D1(L1, B1), D2(L2, B2)
116	PRZENIEŚ POLE ZNAKÓW	MVC	D2	SS	D1(L, B1), D2(B2)
117	TŁUMACZ	TR	DC	SS	D1(L, B1), D2(B2)
118	TŁUMACZ I ZNACZ	TRT	DD	SS	D1(L, B1), D2(B2)
119	MNÓŻ LOGICZNIE	N	54	RX	R1, D2 (X2, B2)
120	MNÓŻ LOGICZNIE	NC	D4	SS	D1(L, B1), D2(B2)
121	MNÓŻ LOGICZNIE	NI	94	SI	D1(B1), 12
122	MNÓŻ LOGICZNIE	NR	14	RR	R1, R2
123	PORÓWNAJ	C	59	RX	R1, D2 (X2, B2)
124	PORÓWNAJ	CR	19	RR	R1, R2
125	PORÓWNAJ DZIESIĘTNIE	CP	F9	SS	D1(L1, B1), D2(L2, B2)
126	PORÓWNAJ ZMP D	CD	69	RX	R1, D2 (X2, B2)
127	PORÓWNAJ ZMP D	CDR	29	RR	R1, R2
128	PORÓWNAJ ZMP K	CE	79	RX	R1, D2 (X2, B2)
129	PORÓWNAJ ZMP K	CER	39	RR	R1, R2
130	PORÓWNAJ PÓLSŁOWO	CH	49	RX	R1, D2 (X2, B2)
131	PORÓWNAJ LOGICZNIE	CL	55	RX	R1, D2 (X2, B2)
132	PORÓWNAJ LOGICZNIE POLA	CLC	D5	SS	D1(L, B1), D2(B2)
133	PORÓWNAJ LOGICZNIE ZNAKI	CLI	95	SI	D1(B1), 12
134	PORÓWNAJ LOGICZNIE	CLR	15	RR	R1, R2
135	SKOK PRZY INDEKSIE WIĘKSZYM	IXH	80	RS	R1, R3, D2(B2)

1	2	3	4	5	6
136	SKOK PRZY INDEKSIE MNIEJSZYM LUB RÓWNYM	DXLE	87	RS	R1, R3, D2 (B2)
137	SKOK ZE ŚLADEM	BAL	45	RX	R1, D2 (X2, B2)
138	SKOK ZE ŚLADEM	BALR	05	RR	R1, R2
139	SKOK WG WARUNKU	BC	47	RX	M1, D2 (X2, B2)
140	SKOK WARUNKOWY	BCR	07	RR	M1, R2
141	SKOK LICZNIKA	BCT	46	RX	R1, D2, (X2, B2)
142	SKOK LICZNIKA	BCTR	06	RR	R1, R2

Wg Rechentchnik Datenverarbeitung 1976 nr 3
opracowała Janina Wąsowska

mgr inż. Wacław JAKACKI

Centrum Badawczo-Rozwojowy
Centrum Komputerowych Systemów
Automatyki i Pomiarów "MERA-ELMRO"

TECHNICZNE ŚRODKI KONTROLI I DIAGNOSTYKI USZKODZEŃ ELEKTRONICZNEJ MASZYNY CYFROWEJ R-32

Wysokie wymagania stawiane maszynom Jednolitego Systemu w zakresie wykrywania i szybkiej lokalizacji uszkodzeń sprawiły, że w jednostkę centralną (JC) elektronicznej maszyny cyfrowej R-32 wbudowano środki kontroli i diagnostyki o możliwościach nie spotykanych w dotychczasowych rozwiązaniach krajowych.

W artykule przedstawiono techniczne środki wykrywania uszkodzeń JC R-32, reakcję systemu na wykryty błąd i wreszcie sprzęt i procedury diagnostyczne.

Wymienione środki kontrolno-diagnostyczne są związane w określony sposób z systemem operacyjnym. I tak, sprzęt i mikroprogramy kontroli zapisują stan JC po błędzie, co pozwala odpowiednim programom (SER, MCH, CCH) na próbę powtórzenia przerwanej akcji i uzupełnienie statystyki błędów. Z kolei program EREP wyprowadza w postaci zredagowanej informację o błędach stanowiącą punkt wyjścia dla operacji diagnostycznych. Jeśli system nie może kontynuować pracy po błędzie - informację taką wyprowadza program autonomiczny SEREP. Do lokalizacji uszkodzenia służą testy wykorzystujące sprzęt i mikroprogramy diagnostyczne. Po usunięciu uszkodzenia poprawne działanie całego systemu sprawdza się testami DMES. Testy te stanowią zbiór sekcji pracujących pod kontrolą autonomicznego programu-monitora.

Efektywno środki kontrolno-diagnostyczne w zasadniczy sposób wpływają na odporność systemu na uszkodzenia sprzętu i czas odnowy systemu.

Układy kontroli jednostki centralnej R-32

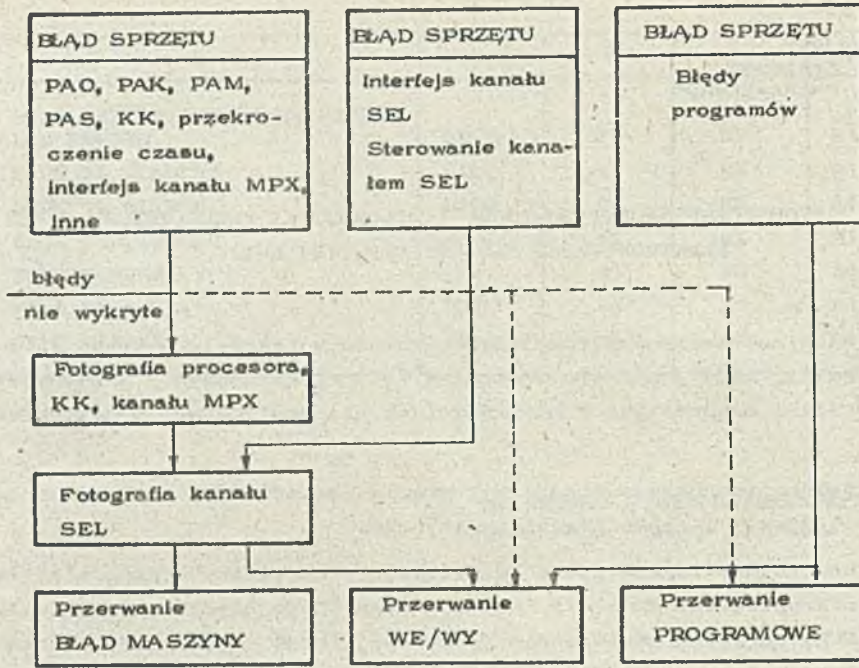
Określono węzły przepływu informacji objęto kontrolą ciągłą tzn. wykonywaną w każdym podstawowym cyklu maszynowym. Ponieważ arytmometr, a w szczególności sumator, są zbudowane z elementów średniej skali integracji i układy kontroli byłyby objętościowo porównywalne z układami sprawdzanymi, zrezygnowano w wypadku tego bloku z kontroli ciągłej, zastępując ją mikroprogramowym testem okresowym procesora sprawdzającym między rozkazami poprawność działania arytmometru.

Struktura JC R-32 z wydzielonymi zasadniczymi blokami funkcjonalnymi: procesorem z pamięcią stałą (PAS), pamięcią operacyjną (PAO) i kanałami jest schematycznie przedstawiona na rys. 1. Na rysunku naniesiono punkty, w których sprawdzana jest poprawność przyjmowanej informacji. Oznaczenia punktów odpowiadają punktom poniższego opisu i pozycjom rejestru błędów, w którym przyezym błędzie jest zapamiętywana (por. rys. 3).

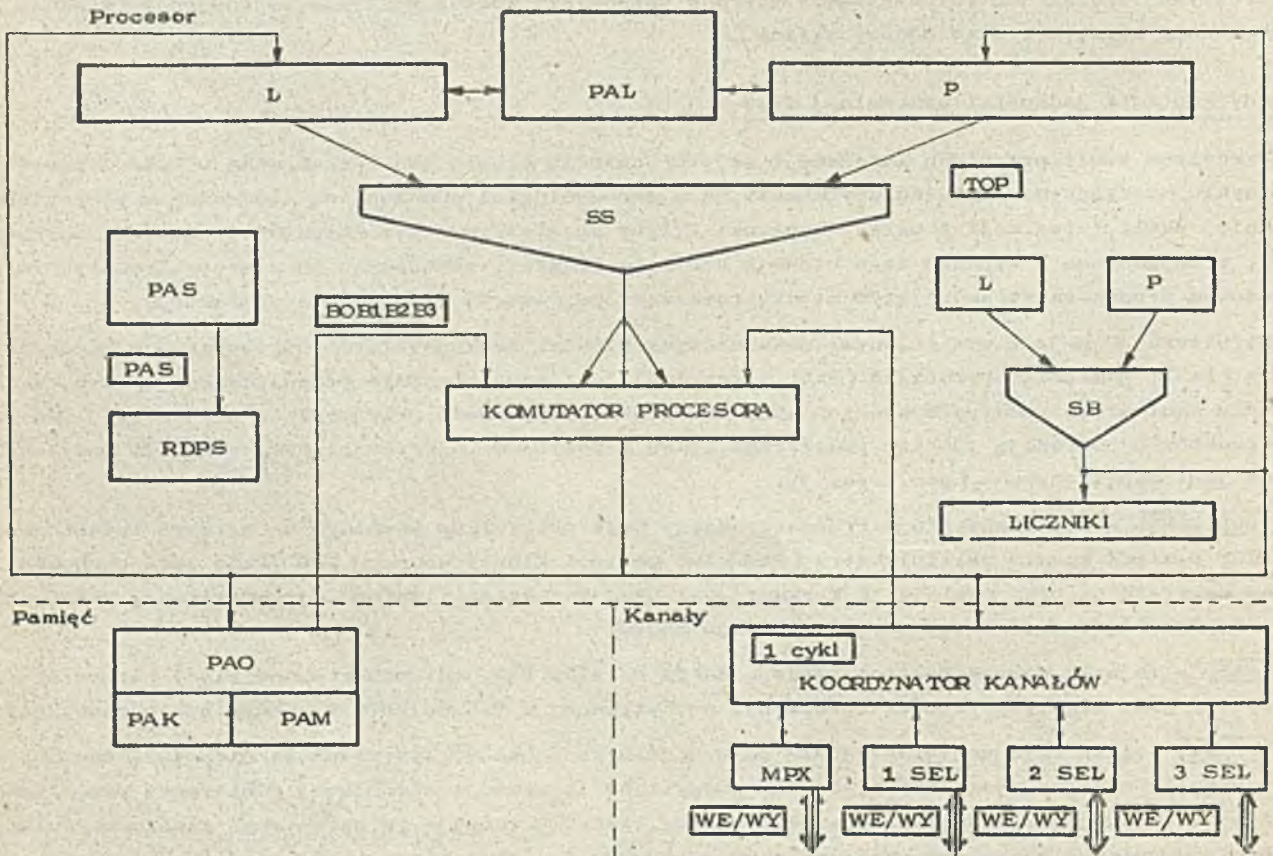
[J011B2B3] - Interfejs PAO - Procesor. Każdy bajt przesyłany wspólnym 4-bajtowym interfejsem do PAO, pamięci kanału multipleksora (PAM) lub pamięci kluczy ochrony PAO (PAK) jest cechowany bitem parzystości przy zapisie. Wykrywany jest błąd parzystości informacji odczytanej z PAO, PAM, PAK. Sygnał błędu jest formowany dla każdego bajtu.

[PAS] - rejestr danych pamięci stałej. Każde ze słów PAS jest nacechowane bitem parzystości. Wykrywany jest błąd parzystości informacji odczytywanej z PAS do RDPS w każdym cyklu maszynowym.

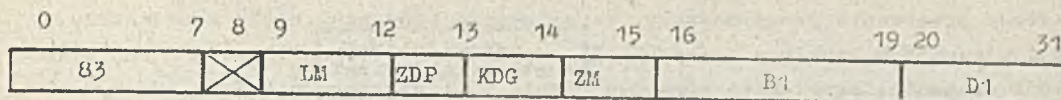
[WE/WY] - interfejs wejścia/wyjścia poszczególnych kanałów. Wykrywane są nieprawidłowe sekwencje sygnałów w interfejsie WE/WY, błędna parzystość informacji sterującej odbieranej przez kanały z urządzeń wejścia/wyjścia. Dodatkowo układy kontroli reagują po określonym czasie na całkowity brak odpowiedzi ze strony urządzenia wejścia/wyjścia (tzw. błąd przekroczenia czasu po około 60 μ s od zainicjowania rozkazu wejścia/wyjścia) a także na fakt odłączenia się urządzenia wejścia/wyjścia od interfejsu w czasie łańcucha komend.



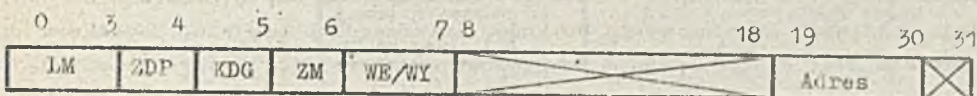
Rys. 1. Blokowa struktura JC R-32



Rys. 2. Reakcja systemu na błąd



DIAGNOZA



Diagnostyczne słowo sterujące

- LM - licznik mikrorozkazów
- ZDP - zezwolenie na dostęp do PAO podczas odliczania mikrorozkazów
- KDG - kontynuuj diagnostykę
- ZM - zliczaj mikrorozkazy
- WE/WY - tryb deszyfracji mikrorozkazu: procesorowy; w rozkazie DIAGNOZA ładowany bitem 8 sumy (B1) + D1
- Adres - Adres pierwszego mikrorozkazu w ciągu; w rozkazie DIAGNOZA ładowany bitami 20 + 31 sumy (B1) + D1

Rys. 3. Format rozkazu DIAGNOZA i diagnostycznego słowa sterującego

1 CYKL - koordynator kanałów (KK). Kanały w EMC r-32 są mikroprogramowane. Mikroprogramy związane z czynnościami kanału są podzielone na krótkie odcinki zwane mikroprogramami obsługi kanałów (MOK). Podział taki pozwala na zachowanie ustalonych priorytetów obsługi zgłoszeń z kanałów z dokładnością do czasu wykonania jednej MOK max. (2,3 μs). Kolejne MOK są wybierane z uwzględnieniem priorytetu przez sprzęt KK. Układy kontroli sprawdzają prawidłowość wyboru MOK badając mikroprogramowy znacznik w każdym pierwszym mikrorozkazie MOK. Brak znacznika sygnalizuje błąd wyboru i wejścia do komórki innej niż pierwsza komórka MOK.

W punktach wyliczonych powyżej kontrola jest prowadzona w sposób ciągły. Ponadto w mikroprogramy JC R-32 jest wbudowany tzw. test okresowy procesora **TOP**. TOP jest uruchamiany co 20 ms, kiedy następuje zmiana stanu rejestru czasomierza. Mikroprogram TOP sprawdza (rys. 1) sumator bajtowy (SB), sumator słowowy (SS), przesłania między rejestrami prawej (P) i lewej (L) strony SS, operacje przesunięć na komutatorze procesora i działania liczników. Sprawdzanie wykonywane jest metodą "bootstrap", tzn. obejmuje coraz szerszy zakres sprzętu procesora. TOP zużywa mniej niż 1 promil czasu maszyny. Ma on również pewne znaczenie diagnostyczne - punkt wyjścia z mikroprogramu TOP, pod wpływem błędu określa w przybliżeniu miejsce uszkodzenia.

Poza układami kontroli, wykrywającymi uszkodzenia sprzętu JC, istnieją układy wykrywające błędy programów procesora i błędy programów kanałów - błędy wynikające z niezgodności formalnej tych programów z zasadami działania Jednolitego Systemu EMC. Układy te w pewnych wypadkach mogą zareagować na błąd sprzętu, "wspomagając" w ten sposób podstawowe układy kontroli uszkodzeń (na rys. 2 - linia przerywana).

Reakcja na błąd sprzętu jednostki centralnej

Wszelka informacja związana z wykonywanym programem (licznik rozkazów, kod warunków, klucz programu a także maski zezwalające na przyjmowanie przerwania BŁĄD MASZYNY, wejścia/wyjścia, zewnętrznych i programowych) zapamiętana jest w słowie stanu programu (PSW). Wykonanie przerwania polega na wymianie "starego" na "nowe" PSW, a więc na wymianie licznika rozkazów i warunków (nowe maski), w jakich będzie pracował nowy program. W sytuacji przerwania wejścia/wyjścia dodatkowa informacja o jego przyczynach jest zapamiętywana w tzw. słowie stanu kanału (CSW). Reakcja na błąd zależy od stanu PSW i pownych kluczy pulpitu technicznego (PT). W normalnym stanie systemu maski przerwania BŁĄD MASZYNY i wejścia/wyjścia zezwalają na te przerwanie, a klucze PT nie blokują reakcji na błąd. W tej sytuacji wykryte uszkodzenia sprzętu procesora, pamięci, koordynatora kanałów, kanału multipleksorowego a także błędy typu przekroczenie czasu w którymkolwiek z kana-

low powodują natychmiastowe przerwanie pracy maszyny, zapamiętanie przyczyny błędu w rejestrze błędów, przerwanie akcji w kanałach współpracujących z KK, wykonanie "fotografii" stanu rejestrów JC na ustalone pole PA0 i wreszcie "przerwanie właściwe" czyli wymianę starego PSW na PSW wyznaczające adres programu obsługi błędu maszyny. Przez czas wykonywania mikroprogramu "fotografii" reakcja na błąd jest zablokowana a kanały "zamrożone". Nowe PSW ma zazwyczaj maskę przerwania BŁĄD MASZINY zamkniętą, co daje możliwość przetworzenia do końca informacji o błędzie. Dodatkowa informacja o stanie kanałów zawarta jest w niedostępnym programowo obszarze PAL, który bezpośrednio po przerwaniu jest odczytywany rozkazem DIAGNOZA.

Po przerwaniu błąd maszyny sterowanie jest zwykle przekazywane programowi przetwarzającemu informację zawartą w "fotografii". Następnie system operacyjny może próbować kontynuować przerwany program lub dokonać jego restartu.

Błędy sprzętu kanałów selektorowych powodują ustawienie wskaźników "błąd sterowania kanału" i/lub "błąd sterowania interfejsu" w CSW, zapis "fotografii" stanu kanału w ustalone miejsce PA0 i wreszcie wykonanie przerwania WE/WY.

Błędy sprzętu wykryte przez układy sprawdzające formalną poprawność programów prowadzą do przerwania programowych lub WE/WY (rys. 2). Analiza zapamiętanego PSW pozwala określić, czy przyczyna leży rzeczywiście w programie, czy też jest to uszkodzenie sprzętu JC.

Określone ustawienie masek w PSW oraz kluczy na PT może ograniczyć lub całkowicie zlikwidować reakcję systemu na błąd.

Sprzęt diagnostyki jednostki centralnej

Na sprzęt diagnostyki JC składają się układy logiczne związane z testowaniem kanałów, układy sterujące wykonaniem rozkazu DIAGNOZA oraz specjalne mikroprogramowe procedury diagnostyczne realizujące niektóre funkcje tego rozkazu.

Rozkaz DIAGNOZA jest rozkazem uprzywilejowanym, wykonywanym tylko w stanie supervisor. Aby zapobiec niepowołanemu użyciu tego rozkazu w programach pisanych w języku ASSEMBLER nie przyporządkowano mu żadnego kodu mnemonicznego. W związku z tym rozkaz DIAGNOZA musi być deklarowany w ciągu zwykłych rozkazów maszynowych jako stała. Rozkaz ma format SI i poszczególne jego pola sterujące określonymi funkcjami (rys. 3 i 4).

Funkcję identyczną z rozkazem DIAGNOZA pełni diagnostyczne słowo sterujące (DSS) używane wówczas, gdy pojedynczy rozkaz nie wystarczy do zakończenia operacji diagnostycznej. Wtedy rozkaz DIAGNOZA, wraz z następującymi po nim polami danych i DSS-ami tworzy jeden krok testu diagnostycznego, traktowany przez sterowanie jako jeden rozkaz maszynowy.

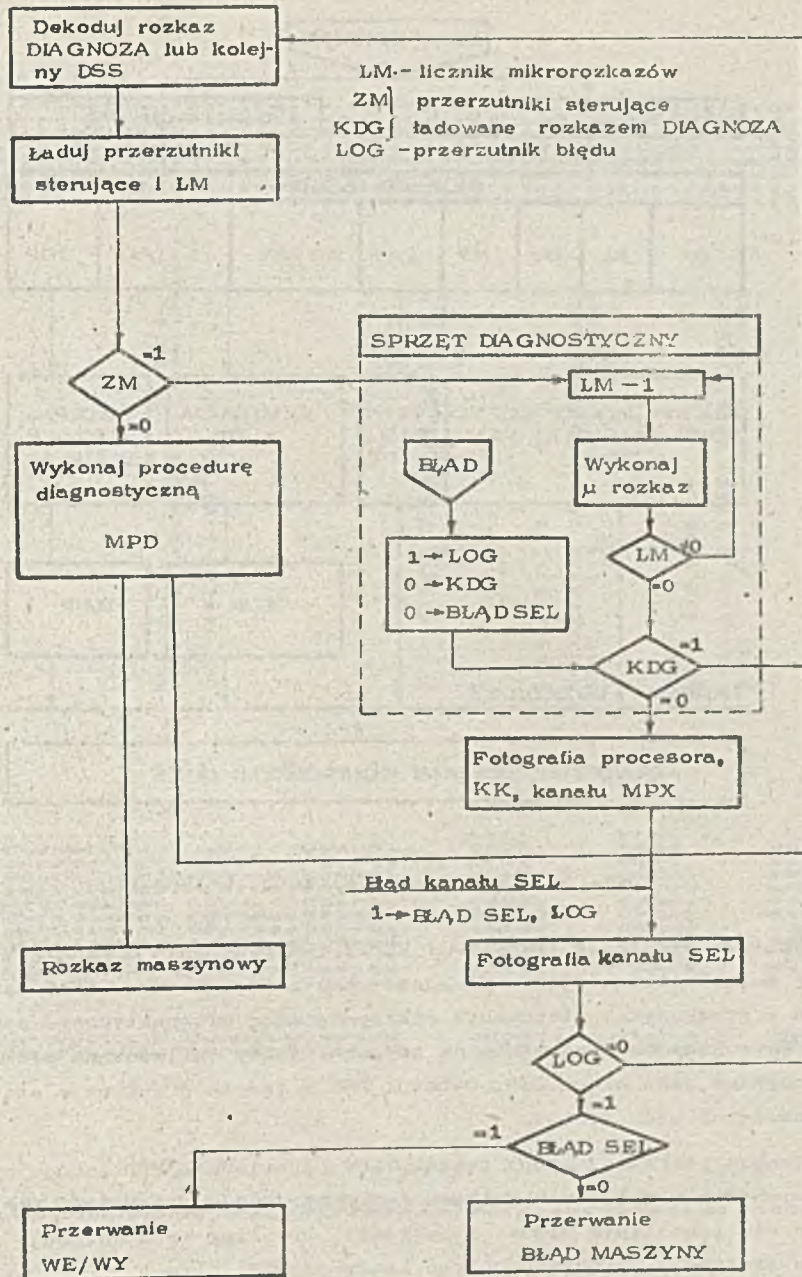
Rozkaz DIAGNOZA pozwala na wykonanie operacji (rys. 4) niemożliwych w wypadku standardowych instrukcji maszynowych:

- wejścia w dowolny ciąg mikrorozkazów, wykonania zadanej liczby mikrorozkazów i wyjścia do następnego DSS,
- wejścia w jedną z mikroprogramowych procedur diagnostycznych (MPD), które m.in. ładują rejestry arytmometru zadaną przez programistę informacją, kontaktują się z niedostępnymi programowo rejestrami PAL i PAM, ładują rejestr symulujący sygnały z interfejsu WE/WY oraz rejestr sterujący diagnostyką kanałów,
- zapisu stanu JC w dowolnym momencie operacji diagnostycznej (mikroprogram wspólny dla rozkazu DIAGNOZA i przerwania BŁĄD MASZINY, rys. 4),
- wejścia do "normalnego" rozkazu maszynowego.

O wykonywanej funkcji diagnostycznej decydują: ustawienie przerzutników sterujących i liczniki mikrorozkazów ładowanych z pól rozkazu DIAGNOZA lub kolejnego DSS przed rozpoczęciem akcji.

Diagnostyka uszkodzeń jednostki centralnej

Taktykę lokalizacji uszkodzeń przedstawia rys. 5. Odpowiednio zrodagowany wydruk "fotografii" lub sygnalizacja na PT dostarczają informacji o stanie maszyny w momencie wystąpienia błędu. Naj-

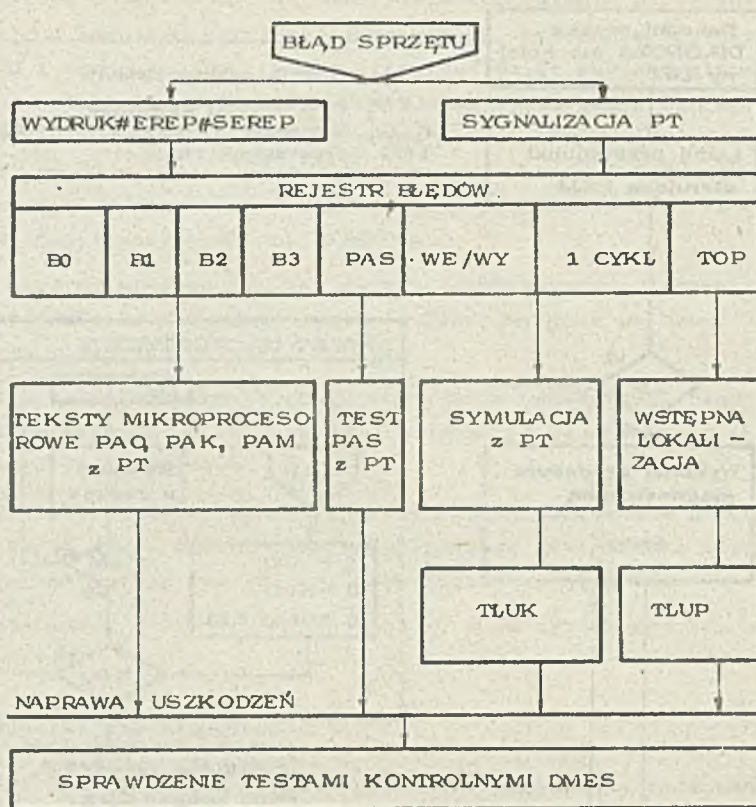


Rys. 4. Działanie rozkazu DIAGNOZA

prostsze funkcje diagnostyczne wykonuje operator z pulpitu. Bogaty zestaw pulpitowych testów pomocy PAO, PAK, PAL, PAM i PAS pozwala na umiejscowienie uszkodzeń w wymienionych blokach z dokładnością do pakietu. Test TOP jest także narzędziem diagnostycznym, ponieważ adres błędnie wykonanego mikrorozkazu w TOP określa w przybliżeniu przyczynę błędu. Dodatkowo istnieje możliwość symulacji z PT sygnałów interfejsu WE/WY w każdym z kanałów, co pozwala sprawdzić kanał i koordynator kanałów bez wykorzystywania urządzeń wejścia/wyjścia.

Podstawowym środkiem lokalizacji uszkodzeń JC są diagnostyczne testy lokalizujące uszkodzenia (DTLU) dzielące się na testy lokalizujące uszkodzenia procesora (TLUP) i kanałów (TLUK).

Testy TLUP są wprowadzane z taśmy przez sprawny kanał selektorowy specjalną procedurą początkowego ładowania programu diagnostycznego (IPLD). W wypadku "porażki" należy naprawić kanał przez wymianę identycznych pakietów z dowolnego z dwu pozostałych kanałów (kanał ma trzy pakiety). Pulpit



Rys. 5. Procedura lokalizacji uszkodzeń

pit techniczny umożliwia symulację transmisji WE/WY i sprawdzenie w kanale sprzętu działającego w czasie IPLD. Aby zminimalizować jądro diagnostyczne zaprojektowano IPLD tak, aby zużywał minimum sprzętu kanału i przekazywał sterowanie mikroprogramom diagnostycznym operującym tylko na DSS. W testach TLUP rozkazy maszynowe nie są używane. Testy są ładowane blokami do pierwszych 4 kbajtów PAQ i wykonywane jako ciągi DSS. Ostatni DSS w teście przekazuje sterowanie procedurze IPLD wczytującej następną grupę testów.

W czasie wykonywania pierwszej grupy testów TLUP odpowiednia MPD ładuje określoną zawartość rejestrów procesora, które w kolejnych krokach testu sprawdzane są bit po bicie. Po sprawdzeniu rejestrów następuje przesłanie między rejestrami. Przesłania tych dokonuje się za pomocą wybranych mikrorozkazów procesora wykonywanych pod kontrolą DSS. Trzeci i czwarty etap TLUP obejmuje sprawdzenie modyfikacji adresów PAS i różnych sposobów adresacji PAL.

Sygnalizacja na PT określa w wypadku błędu numer i krok testu a zatem bit rejestru i oczekiwaną zmianę zawartości (1-0 lub 0-1). Suma informacji o uszkodzeniu, dostarczona przez wszystkie testy, pozwala określić miejsce uszkodzenia z dokładnością co najmniej do pakietu. Dokładność lokalizacji zależy oczywiście od położenia elementu w strukturze logicznej. Odpowiednie ustawienie kluczy na PT pozwala na wykonywanie błędnego testu w pętli.

Dokumentacja testów TLUP składa się z wydruku testów w języku ASSEMBLER wraz z podaną w każdym kroku lokalizacją sprawdzanego elementu, jego wejść i wyjść oraz z oczekiwanym wynikiem testu. Każdemu krokowi testu jest przypisany adres wyświetlany na PT w wypadku błędu. Fragment testu TLUP przedstawia rys. 6.

Ponieważ przyczyną wielu błędów była pamięć stała, opracowano specjalny test pozwalający na precyzyjnie określenie pola strobowego PAS i lokalizację elementów uszkodzonych lub o gorszych parametrach.

Test za pomocą rozkazu DIAGNOZA czyta i wykonuje wszystkie mikrorozkazy i pary mikrorozkazów

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	START	START DC	HPG
000000	00000002			3	START	X'00000002'		HPG00001
000000	00000002			4	DC			HPG00002
000004	2E001F62			5			DC	HPG00003
000008	70001F60			6			DC	HPG00004
00000C	75A178C0			7			DC	HPG00005
000010	DA074265			8			DC	HPG00006
000014	F79438A3			9			DC	HPG00007
000018	783098B4			10			DC	HPG00008
00001C	ASAS45B2			11			DC	HPG00009
000020	700A4F0C			12			DC	HPG00010
000024	00009000			13			DC	HPG00011
000028	00FF37B5			14			DC	HPG00012
00002C	113378FC			15			DC	HPG00013
000030	00000000			16			DC	HPG00014
000034	00000000			17			DC	HPG00015
000038	00011E82			18			DC	HPG00016
00003C	00001F60			19			DC	HPG00017
000040	75AF78C0			20			DC	HPG00018
000044	DA074265			21			DC	HPG00019
000048	F79438A3			22			DC	HPG00020
00004C	783098B4			23			DC	HPG00021
000050	ASAS45B2			24			DC	HPG00022
000054	700A4F0C			25			DC	HPG00023
000058	00009000			26			DC	HPG00024
00005C	00FF37B5			27			DC	HPG00025
000060	113378FC			28			DC	HPG00026
000064	00000000			29			DC	HPG00027
000068	FFFFFEFF			30			DC	HPG00028
000070	00003000			31			DC	HPG00029
000074	80000000			32			DC	HPG00030
000078	80000000			33			DC	HPG00031
00007C	80000000			34			DC	HPG00032
000080	00001696			35			DC	HPG00033
000084	00003000			36			DC	HPG00034
000088	40000000			37			DC	HPG00035
00008C	40000000			38			DC	HPG00036
000090	00000000			39			DC	HPG00037
000094	00001696			40			DC	HPG00038
000098	00003000			41			DC	HPG00039
00009C	20000000			42			DC	HPG00040
0000A0	20000000			43			DC	HPG00041
0000A4	00000000			44			DC	HPG00042
0000A8	00001696			45			DC	HPG00043
0000AC	00003000			46			DC	HPG00044
0000B0	10000000			47			DC	HPG00045
0000B4	10000000			48			DC	HPG00046
0000B8	00000000			49			DC	HPG00047
0000BC	00000000			50			DC	HPG00048
0000C0	00000000			51			DC	HPG00049
0000C4	00000000			52			DC	HPG00050
0000C8	00000000			53			DC	HPG00051
0000CC	00000000			54			DC	HPG00052
0000D0	00000000			55			DC	HPG00053
0000D4	00000000			56			DC	HPG00054
0000D8	00000000			57			DC	HPG00055
0000DC	00000000			58			DC	HPG00056
0000E0	00000000			59			DC	HPG00057
0000E4	00000000			60			DC	HPG00058
0000E8	00000000			61			DC	HPG00059
0000EC	00000000			62			DC	HPG00060
0000F0	00000000			63			DC	HPG00061
0000F4	00000000			64			DC	HPG00062
0000F8	00000000			65			DC	HPG00063
0000FC	00000000			66			DC	HPG00064
000000	00000000			67			DC	HPG00065
000004	00000000			68			DC	HPG00066
000008	00000000			69			DC	HPG00067
00000C	00000000			70			DC	HPG00068
000010	00000000			71			DC	HPG00069
000014	00000000			72			DC	HPG00070
000018	00000000			73			DC	HPG00071
00001C	00000000			74			DC	HPG00072
000020	00000000			75			DC	HPG00073
000024	00000000			76			DC	HPG00074

Rys. 6. Fragment testu ILUP sprawdzającego pierwsze cztery bity rejestru H przelączonego z zer na jednici.

(licznik w DSS równy 1 lub 2). W ten sposób większość "naturalnych" przejść między komórkami PAS jest sprawdzana. Test reaguje tylko na błąd PAS i sam używa mniej niż 10% mikrorozkazów PAS. Zmieniając położenie strobu w czasie pracy testu łatwo lokalizuje się najczęstsze przyczyny wąskiego pola strobowego:

- uszkodzony rdzeń,
- za długi czas przełączania tranzystorów matrycy,
- nieprawidłowo ułożoną nitkę,

Test ten znacznie skrócił i ułatwił proces uruchamiania bloku PAS. Jest również nieoceniony w czasie eksploatacji systemu.

Testy TLUK są wykonywane przy założeniu poprawnego działania procesora, PAO, PAL i PAS. Używane są rozkazy maszynowe a w szczególności rozkaz DIAGNOZA. Testy TLUK są wczytywane przez sprawny kanał selktorowy i uruchamiane przez specjalny program sterujący. Program ten ponadto sprawdza zgodność wyniku testu z wzorcami a także sygnalizuje na PT wykryte niezgodności. Wzorce otrzymano przez zapis stanów sprawnej maszyny.

Schemat testu pokazuje rys. 7. Procedura polega na wykonywaniu w kanale fragmentu operacji WE/WY wydłużonego w kolejnych krokach testu. Impulsy sterujące kanałem są zwalniane tylko na okres wykonywania fragmentu operacji WE/WY. Po każdym kroku wykonuje się automatycznie fotografia



Rys. 7. Schemat diagnostycznego testu kanału TLUK

procesora i "zamrożonego" kanału. Na czas operacji diagnostycznych kanał jest logicznie oddzielony od interfejsu WE/WY - konieczne sygnały wejściowe są symulowane przez specjalny rejestr. Ładowniom tego rejestru, tak jak odliczaniem mikrorozkazów i wykonaniem fotografii, steruje rozkaz DIAGNOZA wraz z następującym po nim ciągiem DSS. Przed każdym krokiem testu kanał jest ustawiany w określony stan początkowy. Fotografia, wykonywana po takim kroku jest porównywana z wzorcem przez program sterujący testami. Sprawdzenie w kanale praktycznie wszystkich operacji WE/WY, podzielonych na dużą liczbę kroków, zapewnia dużą efektywność testów.

Dokumentacja testów TLUK składa się z graficznych obrazów testów przedstawiających sprawdzany ciąg mikrorozkazów z zaznaczonymi numerami kroków testu i ważniejszymi sygnałami w kanale oraz z wydruków testów w języku ASSEMBLER (rys. 8 i 9).

Objętość programu wynikowego DTLU wynosi około 1500 kbajtów.

Po zakończeniu procedur lokalizacyjnych i naprawie uszkodzenia pozostaje sprawdzenie systemu testami kontrolnymi DMES (rys. 5) wchodzącymi w skład programowych środków kontroli poprawnej pracy maszyny.

Testy DTLU znajdują się obecnie w początkowym stadium eksploatacji i z pewnością uwagi użytkowników wpłyną na zwiększenie ich efektywności. Testy DTLU powinny być bowiem uzupełniane testami, których projektant nie przewidział. Takie sprzężenie projektant - użytkownik będzie gwarantowało, że testy diagnostyczne nie będą odłożone na półkę.

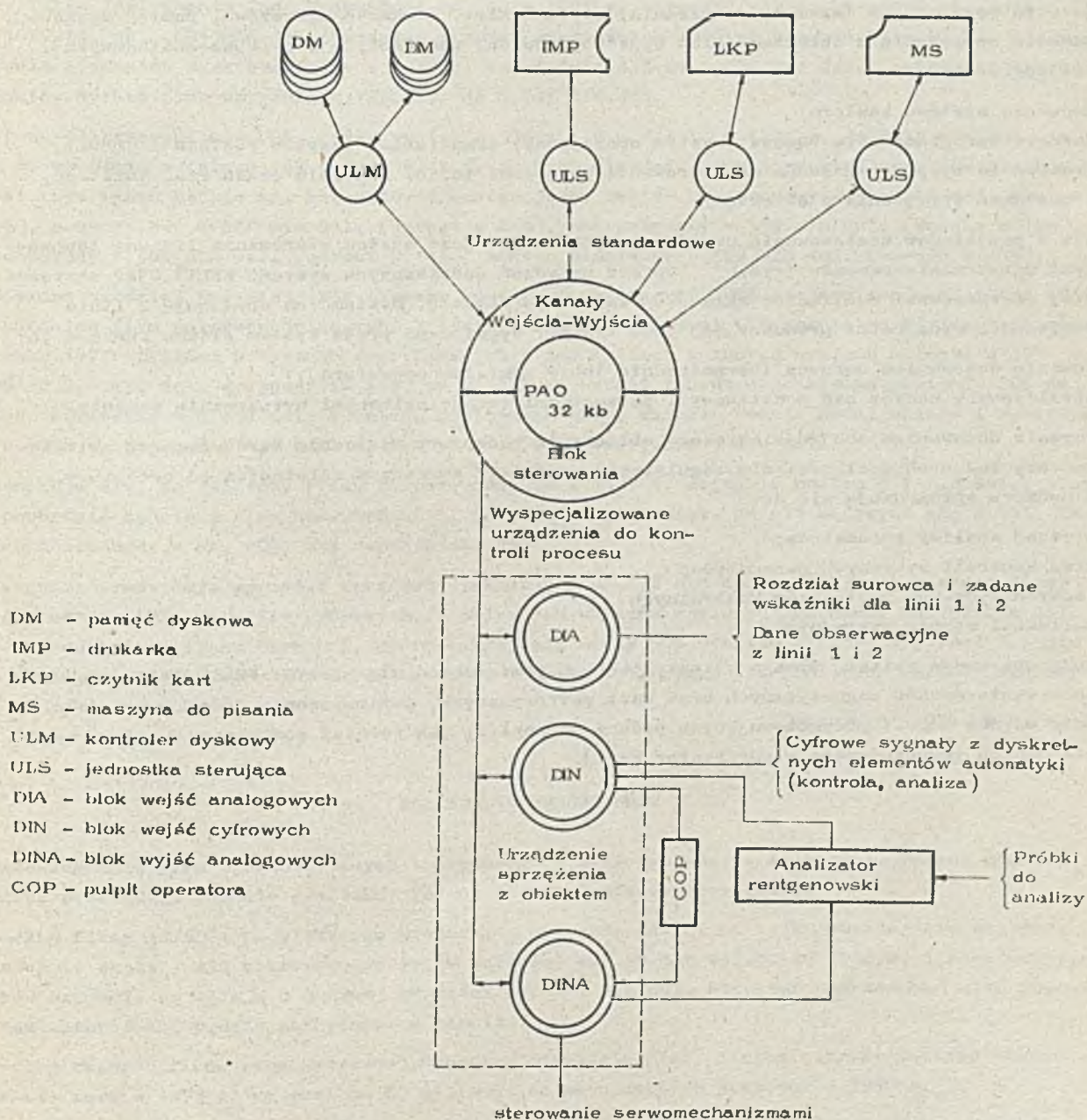
* Przedstawione na rys. 6, 8, 9 wydruki są fragmentami dokumentacji testów DTLU dostarczanej użytkownikowi.

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	PAGE
003480	05F0			283	BALR 15,0	0116
003481	05F0			284	USING 14,0	0117
003482	05F0			285	BALR 14,0	0118
003483	05F0			286	DC C,X(8)KAM,200(000)	0119
003484	03A00F41			287	RESSEL	0120
				288	DC 12E001308, 984-ML027;L=1,ZLM,ZDP,KDG	0121
				289	DC 0C0001300, 9C8-ML201;L=0,ZDP,KDG	0122
				290	DC X,X(8)KAM,200(000)	0123
				291	DC X,X(8)KAM,200(000)	0124
				292	DC X,X(8)KAM,200(000)	0125
				293	DC X,X(8)KAM,200(000)	0126
				294	DC X,X(8)KAM,200(000)	0127
				295	DC X,X(8)KAM,200(000)	0128
				296	DC X,X(8)KAM,200(000)	0129
				297	DC X,X(8)KAM,200(000)	0130
				298	DC X,X(8)KAM,200(000)	0131
				299	DC X,X(8)KAM,200(000)	0132
				300	DC X,X(8)KAM,200(000)	0133
				301	DC X,X(8)KAM,200(000)	0134
				302	DC X,X(8)KAM,200(000)	0135
				303	DC X,X(8)KAM,200(000)	0136
				304	DC X,X(8)KAM,200(000)	0137
				305	DC X,X(8)KAM,200(000)	0138
				306	DC X,X(8)KAM,200(000)	0139
				307	DC X,X(8)KAM,200(000)	0140
				308	DC X,X(8)KAM,200(000)	0141
				309	DC X,X(8)KAM,200(000)	0142
				310	DC X,X(8)KAM,200(000)	0143
				311	DC X,X(8)KAM,200(000)	0144
				312	DC X,X(8)KAM,200(000)	0145
				313	DC X,X(8)KAM,200(000)	0146
				314	DC X,X(8)KAM,200(000)	0147
				315	DC X,X(8)KAM,200(000)	0148
				316	DC X,X(8)KAM,200(000)	0149
				317	DC X,X(8)KAM,200(000)	0150
				318	DC X,X(8)KAM,200(000)	0151
				319	DC X,X(8)KAM,200(000)	0152
				320	DC X,X(8)KAM,200(000)	0153
				321	DC X,X(8)KAM,200(000)	0154
				322	DC X,X(8)KAM,200(000)	0155
				323	DC X,X(8)KAM,200(000)	0156
				324	DC X,X(8)KAM,200(000)	0157
				325	DC X,X(8)KAM,200(000)	0158
				326	DC X,X(8)KAM,200(000)	0159
				327	DC X,X(8)KAM,200(000)	0160
				328	DC X,X(8)KAM,200(000)	0161
				329	DC X,X(8)KAM,200(000)	0162
				330	DC X,X(8)KAM,200(000)	0163
				331	DC X,X(8)KAM,200(000)	0164
				332	DC X,X(8)KAM,200(000)	0165
				333	DC X,X(8)KAM,200(000)	0166
				334	DC X,X(8)KAM,200(000)	0167
				335	DC X,X(8)KAM,200(000)	0168
				336	DC X,X(8)KAM,200(000)	0169
				337	DC X,X(8)KAM,200(000)	0170
				338	DC X,X(8)KAM,200(000)	0171
				339	DC X,X(8)KAM,200(000)	0172
				340	DC X,X(8)KAM,200(000)	0173
				341	DC X,X(8)KAM,200(000)	0174
				342	DC X,X(8)KAM,200(000)	0175
				343	DC X,X(8)KAM,200(000)	0176
				344	DC X,X(8)KAM,200(000)	0177
				345	DC X,X(8)KAM,200(000)	0178
				346	DC X,X(8)KAM,200(000)	0179
				347	DC X,X(8)KAM,200(000)	0180
				348	DC X,X(8)KAM,200(000)	0181
				349	DC X,X(8)KAM,200(000)	0182
				350	DC X,X(8)KAM,200(000)	0183
				351	DC X,X(8)KAM,200(000)	0184
				352	DC X,X(8)KAM,200(000)	0185
				353	DC X,X(8)KAM,200(000)	0186
				354	DC X,X(8)KAM,200(000)	0187
				355	DC X,X(8)KAM,200(000)	0188
				356	DC X,X(8)KAM,200(000)	0189
				357	DC X,X(8)KAM,200(000)	0190
				358	DC X,X(8)KAM,200(000)	0191
				359	DC X,X(8)KAM,200(000)	0192
				360	DC X,X(8)KAM,200(000)	0193
				361	DC X,X(8)KAM,200(000)	0194
				362	DC X,X(8)KAM,200(000)	0195
				363	DC X,X(8)KAM,200(000)	0196
				364	DC X,X(8)KAM,200(000)	0197
				365	DC X,X(8)KAM,200(000)	0198
				366	DC X,X(8)KAM,200(000)	0199
				367	DC X,X(8)KAM,200(000)	0200
				368	DC X,X(8)KAM,200(000)	0201
				369	DC X,X(8)KAM,200(000)	0202
				370	DC X,X(8)KAM,200(000)	0203
				371	DC X,X(8)KAM,200(000)	0204
				372	DC X,X(8)KAM,200(000)	0205
				373	DC X,X(8)KAM,200(000)	0206
				374	DC X,X(8)KAM,200(000)	0207
				375	DC X,X(8)KAM,200(000)	0208
				376	DC X,X(8)KAM,200(000)	0209
				377	DC X,X(8)KAM,200(000)	0210
				378	DC X,X(8)KAM,200(000)	0211
				379	DC X,X(8)KAM,200(000)	0212
				380	DC X,X(8)KAM,200(000)	0213
				381	DC X,X(8)KAM,200(000)	0214
				382	DC X,X(8)KAM,200(000)	0215
				383	DC X,X(8)KAM,200(000)	0216
				384	DC X,X(8)KAM,200(000)	0217
				385	DC X,X(8)KAM,200(000)	0218
				386	DC X,X(8)KAM,200(000)	0219
				387	DC X,X(8)KAM,200(000)	0220
				388	DC X,X(8)KAM,200(000)	0221
				389	DC X,X(8)KAM,200(000)	0222
				390	DC X,X(8)KAM,200(000)	0223
				391	DC X,X(8)KAM,200(000)	0224
				392	DC X,X(8)KAM,200(000)	0225
				393	DC X,X(8)KAM,200(000)	0226
				394	DC X,X(8)KAM,200(000)	0227
				395	DC X,X(8)KAM,200(000)	0228
				396	DC X,X(8)KAM,200(000)	0229
				397	DC X,X(8)KAM,200(000)	0230
				398	DC X,X(8)KAM,200(000)	0231
				399	DC X,X(8)KAM,200(000)	0232
				400	DC X,X(8)KAM,200(000)	0233
				401	DC X,X(8)KAM,200(000)	0234
				402	DC X,X(8)KAM,200(000)	0235
				403	DC X,X(8)KAM,200(000)	0236
				404	DC X,X(8)KAM,200(000)	0237
				405	DC X,X(8)KAM,200(000)	0238
				406	DC X,X(8)KAM,200(000)	0239
				407	DC X,X(8)KAM,200(000)	0240
				408	DC X,X(8)KAM,200(000)	0241
				409	DC X,X(8)KAM,200(000)	0242
				410	DC X,X(8)KAM,200(000)	0243
				411	DC X,X(8)KAM,200(000)	0244
				412	DC X,X(8)KAM,200(000)	0245
				413	DC X,X(8)KAM,200(000)	0246
				414	DC X,X(8)KAM,200(000)	0247
				415	DC X,X(8)KAM,200(000)	0248
				416	DC X,X(8)KAM,200(000)	0249
				417	DC X,X(8)KAM,200(000)	0250
				418	DC X,X(8)KAM,200(000)	0251
				419	DC X,X(8)KAM,200(000)	0252
				420	DC X,X(8)KAM,200(000)	0253
				421	DC X,X(8)KAM,200(000)	0254
				422	DC X,X(8)KAM,200(000)	0255
				423	DC X,X(8)KAM,200(000)	0256
				424	DC X,X(8)KAM,200(000)	0257
				425	DC X,X(8)KAM,200(000)	0258
				426	DC X,X(8)KAM,200(000)	0259
				427	DC X,X(8)KAM,200(000)	0260
				428	DC X,X(8)KAM,200(000)	0261
				429	DC X,X(8)KAM,200(000)	0262
				430	DC X,X(8)KAM,200(000)	0263
				431	DC X,X(8)KAM,200(000)	0264
				432	DC X,X(8)KAM,200(000)	0265
				433	DC X,X(8)KAM,200(000)	0266
				434	DC X,X(8)KAM,200(000)	0267
				435	DC X,X(8)KAM,200(000)	0268
				436	DC X,X(8)KAM,200(000)	0269
				437	DC X,X(8)KAM,200(000)	0270
				438	DC X,X(8)KAM,200(000)	0271
				439	DC X,X(8)KAM,200(000)	0272
				440	DC X,X(8)KAM,200(000)	0273
				441	DC X,X(8)KAM,200(000)	0274
				442	DC X,X(8)KAM,200(000)	0275
				443	DC X,X(8)KAM,200(000)	0276
				444	DC X,X(8)KAM,200(000)	0277
				445	DC X,X(8)KAM,200(000)	0278
				446	DC X,X(8)KAM,200(000)	0279
				447	DC X,X(8)KAM,200(000)	0280
				448	DC X,X(8)KAM,200(000)	0281
				449	DC X,X(8)KAM,200(000)	0282
				450	DC X,X(8)KAM,200(000)	0283
				451	DC X,X(8)KAM,200(000)	0284
				452	DC X,X(8)KAM,200(000)	0285
				453	DC X,X(8)KAM,200(000)	0286
				454	DC X,X(8)KAM,200(000)	0287
				455	DC X,X(8)KAM,200(000)	0288
				456	DC X,X(8)KAM,200(000)	0289
				457	DC X,X(8)KAM,200(000)	0290
				458	DC X,X(8)KAM,200(000)	0291
				459	DC X,X(8)KAM,200(000)	0292
				460	DC X,X(8)KAM,200(000)	0293
				461	DC X,X(8)KAM,200(000)	0294
	</					

mgr inż. Roman CZAJKOWSKI
Instytut Maszyn Matematycznych

SYSTEM FELIX C32P

Rumuński przemysł komputerowy produkuje rodzinę urządzeń komputerowych FELIX. Asortyment tych urządzeń obejmuje minikalkulatory, maszyny księgująco-fakturowe sprowadzane do Polski oraz



Rys. Konfiguracja systemu sterowania linią technologiczną wytwarzania cementu

maszyny cyfrowe średniej mocy obliczeniowej. Opracowana w ostatnich latach maszyna FELIX C32P jest przystosowana do sterowania procesami technologicznymi w przemyśle chemicznym, energetyce, przemyśle maszynowym i przemyśle metalurgicznym, może również znaleźć zastosowanie w sterowaniu transportem kolejowym lub eksperymentach naukowych.

System sterowania procesami technologicznymi na bazie urządzeń FELIX C32P jest wynikiem współpracy Instytutu Badawczego Techniki Obliczeniowej (ITC - Institutul de Cercetari pentru Tehnica de Calcul), w którym opracowano jednostkę centralną i oprogramowanie bazowe, Instytutu Badawczo-Projektowego Automatyzacji (IPA), który opracował urządzenie sprzężenia z obiektem oraz systemu oprogramowania sterującego.

W skład podstawowej konfiguracji do sterowania wchodzi:

- jednostka centralna (pamięć 32 kśłów - 32-bitowych, wielodostęp; 32 poziomy przerwań);
- urządzenia peryferyjne (maszyna do pisania, czytnik kart, drukarka wierszowa, pamięć dyskowa);
- urządzenie sprzężenia z obiektem (blok wyjść cyfrowych, analogowych i cyfrowo-analogowych),
- pulpit operatora.

Oprogramowanie systemu zawiera:

- standardowe oprogramowanie bazowe (system operacyjny, translatory języków FORTRAN i COBOL);
- oprogramowanie wyspecjalizowane do sterowania procesami (m.in. kontrola czasu rzeczywistego, testy urządzeń sprzężenia z obiektem).

Jednym z przykładów zastosowania urządzenia FELIX C32P jest system sterowania liniami technologicznymi wytwarzania cementu (rys.) Oprócz urządzeń podstawowych systemu FELIX C32P zastosowano tutaj rentgenowski analizator składu wytwarzanego cementu. System ten obsługuje 2 linie technologiczne o wydajności produkcyjnej 3000 t/dobę. Wypełniane przez system głównie funkcje to:

- sterowanie dozowaniem surowca (bezpośrednio lub z udziałem operatora);
- scentralizowany nadzór nad podstawowymi parametrami technologicznymi wytwarzania cementu.

Sterowanie dozowaniem obejmuje okresowe obliczenia receptury mieszania surowców oraz obliczenia i pomiary zadanych wielkości dla regulatorów rozdziału surowca w zależności od receptury.

Funkcje nadzoru prowadzą się do:

- cyklicznej analizy parametrów;
- ciągłej kontroli wybranych parametrów;
- okresowych wydruków protokółów kontrolnych;
- sygnalizacji stanów awaryjnych.

Instalacja opisanego systemu wymaga klimatyzowanych pomieszczeń dla maszyny FELIX C32P (ok. 80 m²) i przechowywania dysków magnetycznych oraz kart perforowanych, pomieszczenia dodatkowego (około 15 m²) dla młynów i pras przygotowujących próbki do analizy jak również pomieszczenia do przygotowywania danych (wyposażonego w perforator kart).

INFORMACJE Z KRAJU I ZE ŚWIATA

ZE ŚWIATA

SYTUACJA NA RYNKU UKŁADÓW PÓLPRZEWODNIKOWYCH

Światowy rynek półprzewodników wzrósł z 4,2 mld dol. w 1975 r. do 5,6 mld dol. w 1976 r., t.j. o 32%, ale najlepsze ma jeszcze nadzieję - mówi Charles Clough, dyrektor ds. rynku półprzewodników firmy TEXAS Instruments Inc. Przewiduje on, że w 1977 r. sprzedaż półprzewodników wzrośnie o dalsze 18%, tj. do 6,625 mld dol. Sprzedaż układów scalonych w 1976 r. przewyższyła po raz pierwszy sprzedaż elementów dyskretnych, a w 1977 r. wzrośnie o 24% do 3,580 mld dol., natomiast sprzedaż elementów dyskretnych wzrośnie o 12%, tj. do 3,045 mld dol.

Rynek półprzewodników USA, który wzrósł o 33% z 1,8 mld dol. w 1975 r. do 2,39 mld dol. w 1976 r. wzrośnie o dalsze 18% w 1977 r. t.j. do 2,83 mld dol., a w poszczególnych sektorach wzrost tego rynku będzie się kształtował następująco: sektor komputerowy - 540 mln dol. (wzrost o 26%), przemysłowy - 620 mln dol. (wzrost o 24%), energetyczny - 550 mln dol. (wzrost o 15%), konsumpcyjny - 500 mln dol. (wzrost o 14%), sektor państwowy - 500 mln dol. (wzrost o 14%).

Jeszcze bardziej optymistyczne akcenty zawiera raport grupy WEMA stowarzyszającej ponad 40 amerykańskich firm półprzewodnikowych. Zgodnie z danymi zawartymi w raporcie po bardzo niekorzystnym roku 1975 sprzedaż w firmach amerykańskich i ich filiach w innych krajach wzrosła w 1976 r. o 30% do 3,4 mld dol. Przypuszcza się, że w 1977 r. wzrost ten utrzyma się na poziomie 20% dając wartość sprzedaży 4,6 mld dol. W latach następnych tempo wzrostu będzie nadal malało i wyniesie 12% w 1978 r. (sprzedaż 4,6 mld dol.) i 9% w 1979 r. (sprzedaż 5 mld dol.).

Oczekuje się, że światowy rynek półprzewodników w 1977 r. osiągnie poziom 6,1 mld dol., w tym 2/3 pochodzących będzie z firm amerykańskich. Oczekuje się również, że 67% zatrzyma się w USA, Europa Zachodnia zużyje ok. 20%, zaś Japonia ok. 5%.

Prognoza przewiduje sprzedaż urządzeń dyskretnych na 2,8 mld dol. w 1977 r., z czego firmy USA sprzedadzą 57%. Sprzedaż cyfrowych, bipolarnych układów scalonych wzrośnie do 952 mln dol. przy 80% udziale w rynku firm USA, które także będą miały 80% udziału w rynku wartości 1,5 mld dol. dla cyfrowych układów MOS i 70% wartości 744 mln dol. rynku dla liniowych układów scalonych.

Electronics 1977 nr 9 s. 25

ROZWÓJ MIKROPROCESORÓW

Przewidywania firm prognostycznych i producentów mikroprocesorów są zgodne. Podany poniżej przegląd przewiduje zgodnie gwałtowny rozwój techniki mikroprocesorowej.

Według firmy INTEL - największego producenta mikroprocesorów na rynku europejskim znajduje się obecnie około 1 mln mikroprocesorów (o wartości ok. 50 mln dol.). Do 1980 r. liczba mikroprocesorów wzrośnie do 15 mln o łącznej wartości ok. 180 mln dol. Struktura zastosowań mikroprocesorów kształtować się będzie jak podano w tabeli.

Według raportu firmy prognostycznej Creative Strategies Inc. liczba mikrokomputerów wzrośnie od prawie zera w 1973 r. do poziomu 60 milionów zainstalowanych systemów w 1980 r.

Mikroprocesory - podstawowe układy mikrokomputerów - będą wzrastać z bardzo dużą szybkością 110% rocznie do 1980 r. Oczekuje się, że ten wzrost w sprzedaży jednostkowej będzie bardziej przyspieszony po roku 1980.

Sprzedaż jednostkowa mikroprocesorów prawdopodobnie zwiększy się z 0,6 mln sztuk w 1975 r. do

Europejski rynek mikroprocesorów w 1980 r.

Wyszczególnienie	Ilościowo	Wartościowo
Urządzenia kontrolno-pomiarowe	7%	10%
Układy sterowania	8%	15%
Technika przetwarzania danych	11%	16%
Komputery i urządzenia zewnętrzne	10%	19%
Samochody	19%	4%
Artykuły konsumpcyjne	28%	6%
Pozostałe	10%	10%
Zastosowania wojskowe	1%	4%
Transmisja danych	6%	16%

26 - 35 mln sztuk w 1980 r. Sprzedaż w 1975 r. wynosiła 62 mln dol. i może wzrosnąć do 507 - 567 mln dol. w 1980 r. przy rocznym tempie wzrostu o 52 - 55%. Małe tempo wzrostu wartościowego jest wynikiem dramatycznego spadku cen, który może spowodować, że koszt pewnych uniwersalnych pakietów mikrokomputerowych wyniesie poniżej 100 dol.

Zanalizowano szczegółowo dziesięć sektorów rynku użytkowników obejmujących handel i szkolnictwo, telekomunikację, rynek urządzeń konsumpcyjnych, komputerowych urządzeń końcowych i zewnętrznych.

Sektor urządzeń konsumpcyjnych będzie pochłaniać do 1980 r. największą ilościowo sprzedaż, chociaż niekoniecznie największą wartościowo. Segment komputerowych urządzeń końcowych i zewnętrznych będzie stanowił największą wartościowo pozycję rynku w 1980 r. (140 mln dol.).

Firma Frost and Sullivan Inc., specjalizująca się w badaniach rynku i prognozowaniu, w obszernej analizie podaje podobną ocenę rozwoju mikroprocesorów. Według analizy sprzedaż mikrokomputerów, która w 1975 r. osiągnęła wielkość 45 mln dol. osiągnie wielkość 1,2 mld dol. w 1985 r.

Sprzedaż samych mikroprocesorów (22 mln dol. w 1975 r.) osiągnie wielkość 405 mln dol. w roku 1985. Spodziewać się należy, że mikroprocesory segmentowe staną się artykułem rozwojowym, wzrastając od pięciu procent rynku procesorów centralnych w 1977 r. i do 20% w 1985 r. Spodziewane jest zmniejszenie się udziału mikroprocesorów 8-bitowych z poziomu 60% w 1975 r. do 35% rynku w 1985 r. Prognoza dla 12-bitowych i 4-bitowych mikroprocesorów jest także niekorzystna. Spodziewany jest wzrost 16-bitowych mikroprocesorów z 10% rynku w 1975 r. do 45% w 1985 r. Oczekuje się, że przemysł chemiczny będzie stanowić największy rynek mikrokomputerowy w 1985 r. Przewiduje się, że urządzenia oparte na mikrokomputerach w przemyśle naftowym osiągną 110 mln dol. w 1985 r.

Elektronik 1977 nr 11

Electronics Weekly 1977 nr 864

Control Engineering 1977 nr 2

USA ZWIĘKSZA EKSPORT SYSTEMÓW I URZĄDZEŃ KOMPUTEROWYCH

Według danych Departamentu Handlu USA eksport amerykańskich komputerów i urządzeń komputerowych wzrósł w 1976 r. o 16% do 2,587 mln dolarów. Import osiągnął w 1976 r. wielkość 234,2 mln dol. Bilans netto wyniósł 2.352,8 mln dol., co stanowi 12% wzrostu w stosunku do roku ubiegłego. Sprzedaż komputerów i części zamiennych wyniosła 1,339 mln dol. (wzrost o 15%), podczas gdy urządzenia wejścia/wyjścia o 20% do 645 mln dol. Dostawy pamięci wewnętrznych i innych urządzeń pamięci wyniosły ogółem 363 mln dol., do 4%. Pozostałe 197 mln dol. przy 8% wzroście przypada na inne urządzenia związane z komputerami.

Kanada w dalszym ciągu pozostaje największym użytkownikiem amerykańskich komputerów przy zaku-

pie sprzętu o wartości 408 mln dol., po niej Wielka Brytania (305 mln dol.), RFN (303 mln dol.), Francja (237 mln dol.) i Japonia (239 mln dol.).

Electronics 1977 nr 7

FIRMY KOMPUTEROWE W OCENIE SZWAJCARSKICH BANKIERÓW

Bank Kantonalny w Zurychu podał ceny akcji giełdowych producentów sprzętu komputerowego. Podano również uzasadnienia tych ocen. Do pierwszej grupy godnych lokaty kapitału zaliczono te akcje, które nadają się na lokaty średnio- i długoterminowe. Są to papiery wartościowe niżej wymienionych firm:

- IBM - firma z przeszło 50-procentowym udziałem rynkowym, największy producent sprzętu komputerowego. Oprócz tego produkuje ona elektryczne maszyny do pisania, systemy do przetwarzania tekstu oraz urządzenia kopiujące, w którym to zakresie zajmuje trzecie miejsce w świecie. Tę dominację rynkową IBM zawdzięcza bardzo dobrej organizacji sprzedaży i osiągnięciom w zakresie usług. Poza tym IBM postawiła nie tyle na sprzedaż, ile na najem urządzeń. W roku bilansowym 1976 wpływ z usług i najmu wyniosły ok. 64% całkowitych obrotów koncernu. Ocenia się, że trwające obecnie dochodzenie antytrustowe przeciw IBM nie powinno stanowić bezpośredniego zagrożenia dla koncernu.
- SPERRY UNIVAC - zajmuje trzecie miejsce w światowej produkcji komputerów. SPERRY RAND jest znanym producentem sprzętu rolniczego, SPERRY VICKERS-pomp hybrydowych i systemów transmisyjnych, urządzeń nawigacyjnych i radarowych. Firma przeznaczona duże sumy na badania rozwojowe. Koncern ten produkuje w zakresie dostaw sprzętu komputerowego dla linii lotniczych i uniwersytetów.
- SIEMENS - drugi po firmie PHILIPS największy koncern europejski w zakresie elektroniki i największy prywatny pracodawca w RFN. Przedmiotem jego zainteresowań są urządzenia energetyczne oraz systemy informacyjne. Współpraca w zakresie produkcji komputerów z firmą UNIDATA nie udało się, stąd SIEMENS pracuje w tej dziedzinie sam i trzeba trochę poczekać, aż dziedzina ta będzie przynosić zyski firmie.
- BURROUGHS - druga co do wielkości firma produkuje dziś ok. 200 różnych urządzeń: komputerów dużych, średnich i małych. Firma ta może liczyć na bogatych klientów: banki i zakłady ubezpieczeniowe.
- DIGITAL EQUIPMENT - największy producent minikomputerów. W ostatnich latach notuje ona wzrost obrotów średnio rocznie o 39%. Uwagę zwraca fakt, że wzrost ten przypada również na lata recesji. Osiągnięto go jednak ze względu na stale rosnący popyt na komputery do sterowania procesami i minikomputerowe systemy problemowo zorientowane.

Computer Zeitung 4.05.77

RYNEK PRZETWARZANIA SŁÓW OSIĄGNIĘ 2.500 MLN FUNTÓW SZTERLINGÓW

Przetwarzanie słów jest wyrażeniem, które weszło w użycie w ciągu ostatnich kilku lat i dotyczy, jak wyjaśnia nowa analiza "Word Processing in Europe" firmy FROST and SULLIVAN, automatyzacji danych tekstowych, podobnie jak przetwarzanie danych dotyczy automatyzacji danych liczbowych. Autorzy analizy przewidują potrojenie się europejskiego rynku przetwarzania słów w następnym dziesięcioleciu do 2.500 mln funtów szterlingów.

Według analizy wyrażenie "przetwarzanie słów" (word processing) obejmuje tworzenie tekstu odgłosem, na dyktafon lub dyktowanie stenotypistce; przekształcenie tekstu do postaci maszynowej za pomocą urządzenia z klawiaturą, czy też optycznego czytnika znaków lub kresek; wytwarzanie tekstów lub drukowanie; reprodukcję tekstów przez kopiowanie, powielanie lub drukowanie; przechowywanie tekstów lub dokumentów: papierowych, w postaci mikrofilmów i mikrofisz, czy też komputerowych pamięci magnetycznych; dystrybucję tekstów pocztą, telexem lub transmisją telekopiową. Głów-

ny sektor rynku: sprzęt i usługi w zakresie przetwarzania słów wzrosło pięciokrotnie, t.j. od 55 mln funtów szt. w 1976 r. do 355 mln funtów szt. w 1985 r. i osiągnie w tym dziesięcioleciu łączną wartość 1.728 mln funtów szt. RFN z udziałem 40% będzie dominującym rynkiem wśród 17 państw zachodnioeuropejskich. Udział Wielkiej Brytanii i Francji wyniesie po 16%, a Holandii, Szwajcarii i Szwecji po ok. 4%.

Inny sektor europejskiego rynku przetwarzania słów, tj. urządzenia dyktujące, zwiększy się średnio o 7%. Mając wartość 61 mln funtów szt. w 1976 r. osiągnie 102 mln funtów szt. w 1985 r. i łącznie w tym dziesięcioleciu 792 mln funtów. W tym zakresie tylko Wielka Brytania i RFN mają znaczny udział. Holandia, Szwecja i Szwajcaria są raczej umiarkowanymi użytkownikami. Urządzenia dyktujące nie są popularne we Francji i we Włoszech oraz w kilku mniejszych państwach europejskich. Mg analizy, biurowość stanowi znaczną dziedzinę nie tkniętą przez automatyzację. "Praca papierkowa" wzrasta i pochłania coraz większą część siły roboczej biur i urzędów. Potrzebna jest zasadnicza zmiana głęboko zakorzenionych wzorów pracy. Istnieje zwłaszcza potrzeba przełamania tradycyjnego stosunku kierownik - sekretarka. Wymaga to wprowadzenia systematycznego podejścia do korespondencji i komunikowania oraz zastosowania znormalizowanych metod.

Przy średnim koszcie sporządzania listu do 6 funtów szt. w Europie i wydajności europejskich stenotypistek wynoszącej zaledwie jedną czwartą teoretycznej możliwości, w analizie wykazano, że nowe metody i urządzenia prawdopodobnie zyskają dobre przyjęcie, jeśli przedsiębiorstwo będzie próbowało sprawować większą kontrolę nad kosztami i wydajnością pracy w biurach. "Wielu producentów urządzeń komputerowych, telekomunikacyjnych i biurowych doszło do wniosku, że rynek przetwarzania słów oferuje istotny potencjał przyszłościowy", stwierdza analiza. "Firmy przekształcają się w pewnym sensie w to, co można nazwać zupełnie nowym przemysłem, w którym typowe i tradycyjne urządzenia i metody ulegną całkowitej przemianie w przeciągu kilku lat". Będą to firmy rozbudowane tzn. z licznymi użytkownikami, szerokimi liniami produktów i bogatymi zasobami finansowymi dla zapewnienia zyskowej sprzedaży, o szerokich programach badawczo-rozwojowych oraz korzystnych dla użytkownika zasadach dzierżawy urządzeń.

W analizie podzielono urządzenia do przetwarzania tekstów na pięć zasadniczych kategorii.

- Mechaniczne urządzenia do redagowania tekstów. Są one obecnie najbardziej rozpowszechnione na rynku europejskim. Sprzedaż ich będzie zwiększać się prawie aż do lat osiemdziesiątych, ale pod wpływem urządzeń ekranowych znacznie powoli zmniejszać się i całkowita wartość sprzedaży w dziesięcioleciu osiągnie 891 mln funtów szt.
- Wolnostojące procesory tekstów z monitorami ekranowymi. Są one nowością w Europie. Wzrost ich sprzedaży będzie szybki i osiągnie łącznie 529 mln funtów szt. w dziesięcioleciu 1976 - 1985.
- Wielostanowiskowe problemowo zorientowane systemy przetwarzania tekstów są w trakcie wprowadzania w Europie przez wyspecjalizowanych producentów lub działają jako adaptacje istniejących już komputerów biurowych, minikomputerów lub małych systemów komputerowych. Ze względu na duży koszt początkowy i konieczność wprowadzania zmian systemy te będą wprowadzane początkowo tylko do dużych i dobrze zorganizowanych firm. Przewiduje się, że sprzedaż tych systemów wyniesie łącznie 144 mln funtów szt. w ciągu dziesięciolecia.
- Usługi przetwarzania tekstów świadczone w reżimie podziału czasu przez ośrodki usługowe nie będą miały dużego powodzenia w Europie. Ocenia się, że w ciągu dziesięciolecia całkowity dochód z nich nie przekroczy 20 mln funtów szt.
- Komputery do przetwarzania danych dostarczą środków do przetwarzania tekstów związanych ze złożonymi systemami przechowywania i wyszukiwania. Wielu dostawców małych systemów biurowych umożliwi także przetwarzanie tekstów i danych na swych urządzeniach. W tej grupie sprzętu ocenia się, że wydatki w ciągu dziesięciolecia osiągną łączną wartość 144 mln funtów szt.

INTELLIGENTNE URZĄDZENIA KOŃCOWE NA RYNKU EUROPEJSKIM

Rynek inteligentnych urządzeń końcowych wzrasta jak żadna inna branża. Ocenia się, że w 1980 r. w Europie będzie zainstalowanych 480.000 urządzeń końcowych, z których 40% będzie urządzeniami "inteligentnymi". Są to urządzenia, umożliwiające wprowadzanie i wyprowadzanie danych, posiadające interfejs człowiek-maszyna, oddalone od komputera centralnego, z którym połączone są linie transmisji danych. Cechą charakterystyczną inteligentnego urządzenia końcowego jest realizacja tych procesów na bazie własnego procesora, którego zbiór instrukcji jest podzbiorem instrukcji komputera centralnego.

Inteligentne urządzenia końcowe są podstawowym stanowiskiem pracy w przetwarzaniu rozproszonym, a dzięki wbudowanej "inteligencji" mogą odciążać komputer centralny, rozwiązując samodzielnie niektóre zadania cząstkowe. Pierwszymi takimi urządzeniami były teledalokopisy i częściowo same minikomputery.

Producenci minikomputerów weszli więc łatwo na rynek z inteligentnymi urządzeniami końcowymi, podobnie producenci urządzeń zewnętrznych i maszyn biurowych. Podczas gdy producenci branży minikomputerowej dysponują obszernym oprogramowaniem systemowym i komunikacyjnym, producenci komputerów biurowych oferują wiele programów zastosowaniowych. Do tego dochodzą jeszcze producenci klasycznych urządzeń rejestrowania danych. Użytkownik poszukujący inteligentnego urządzenia końcowego powinien po dokładnym określeniu jego zadań jak najdokładniej zbadać "inteligentną" część urządzenia, tzn.: procesor, jednostkę sterującą, pamięć, wejście/wyjście i zdolność komunikacji. Istotnym zagadnieniem jest również sprawa jego kompatybilności. Powinna ona umożliwiać współpracę z komputerami różnych producentów tak, aby w razie ewentualnej zmiany komputera centralnego nie ponosić zbyt dużych kosztów przy zmianie oprogramowania.

Również duże znaczenie dla wyboru urządzenia ma oprogramowanie, konserwacja, repertuar urządzeń zewnętrznych oraz system przerwania urządzenia końcowego. Oprócz powyższych parametrów należy zastanowić się nad kosztami i wyborem dostawcy. Dla oceny jakości dostawcy trzeba stosować takie kryteria jak czas dostawy, późniejsza konserwacja, odległość od najbliższej stacji serwisowej, czas między zgłoszeniem uszkodzenia a przybyciem fachowca, możliwość pomocy dostawcy w zakresie programowania zastosowaniowego, liczba sprzedanych systemów, jakość dokumentacji sprzętu i oprogramowania oraz pomoc przy instalacji i wdrażaniu systemu.

W omawianej dziedzinie sprzętowej daje się zauważyć staranie o dużą wydajność urządzeń, opracowanie systemów bogato oprogramowanych z licznymi kompilatorami, o opracowanie tanich i o zróżnicowanej wydajności urządzeń zewnętrznych.

Jeśli prawdziwo są przypuszczenia obserwatorów firmy IBM, to na przykład pod słowem ORBIT ukrywa się surogat systemu komunikacyjnego IBM 3790, który według wszystkich danych bazuje na komponentach minikomputera System-1 i będzie miał o wiele większą elastyczność niż on sam.

ADL NACHRICHTEN ONLINE 1977 nr 9

PAKIETY PROGRAMÓW W EUROPIE ZACHODNIEJ

Nie ulega wątpliwości, że z punktu widzenia przemysłowej produkcji oprogramowania wystandaryzowane pakiety programów dla typowych zastosowań należy dziś stawiać ponad rozwiązaniem indywidualnymi. Problem ten w klarowny sposób przedstawia raport "ISIS-SOFTWARE", opracowany w Monachium. Według danych zawartych w raporcie w RFN, Austrii i Szwajcarii 252 firmy oferują 1.211 pakietów programowych o łącznej wartości ok. 35 mln DM. 646 pakietów dotyczy zarządzania, 232 pakiety przewidziane są dla zastosowań naukowo-technicznych, a 27,5% (333 programy) stanowią tzw. "systemy/programy usługowe" (w tym systemy operacyjne), które można zaliczyć do oprogramowania podstawowego komputerów. Ceny programów standardowych dla zarządzania wynoszą od 780 do 250,000 DM; w zakresie zastosowań naukowo-technicznych można nabyć programy od 200 do 260,000 DM, w zakresie systemów/programów usługowych od 350 do 393,000 DM. Większość programów standardowych

jest sprzedawana; tylko 1/3 jest oddawana w dzierżawę. Zgodnie z regułą programy to są rozpowszechniane z "nieograniczonym czasowo prawem użytkowania"; kupujący je są więc użytkownikami, a nie ich właścicielami. 50% programów z zakresu zarządzania, 1,3% programów naukowo-technicznych i 24,3% systemów/programów użytkowych jest napisana w języku COBOL, natomiast 19% programów z zakresu zarządzania, 8,6% programów naukowo-technicznych i 60% systemów/programów usługowych napisanych jest w języku ASSEMBLER. Większość programów standardowych opracowano z myślą o komputerach firmy IBM, mianowicie 88,3% programów z zakresu zarządzania, 44,8% programów naukowo-technicznych i 84,6% systemów/programów usługowych.

Częstotliwość zastosowania programów w %

Z a k r e s	Liczba instalacji					
	1	2-5	6-10	11-20	21-50	ponad 50
Programy z zakresu zarządzania	31,4	33,9	18,3	9,4	5,1	1,9
Programy naukowo-techniczne	53,0	34,4	6,0	2,2	2,2	2,2
Systemy/programy usługowe	19,3	23,7	14,7	10,2	7,2	

Po raz pierwszy w katalogu ISIS znalazły się programy komputerów biurowych, systemów zbierania danych i mikrokomputerów.

Biurotechnika 1977 nr 2

PRZEWIDYWANIE SPADKU DOMINACJI USA

Ostatnie sprawozdanie "Przetwarzanie informacji w Stanach Zjednoczonych - streszczenie ilościowe" opublikowane przez AFIPS (American Federation of Information Processing Societies) - przewiduje zanik dominacji USA w zakresie uniwersalnych systemów komputerowych zainstalowanych na świecie w przeciągu najbliższych pięciu lat.

W okresie najbliższych pięciu lat USA może spodziewać się zmniejszenia swej dominacji bazy uniwersalnych systemów komputerowych zainstalowanych na świecie, a nawet jeszcze większego zmniejszenia się przewagi na rynkach minikomputerów i problemowo zorientowanych systemów komputerowych. Przyczyni się do tego agresywna polityka japońskich producentów komputerów.

Według opublikowanych danych wartość komputerów uniwersalnych produkcji USA wynosiła 87% wartości bazy zainstalowanej na świecie w 1976 r. (tabela 1). Przewiduje się zmniejszenie dominacji USA na tym rynku do 81% w 1981 r. Przewiduje się, że udział USA w bazie komputerów zainstalowanych za granicą zmniejszy się nawet z 75% w 1976 r. do 67% w 1981 r. Ze względu na to, że baza zainstalowana jest wielkością kumulatywną, średnią ważoną w stosunku do wartości sprzedaży, oczekuje się, że ta tendencja spadkowa wywoła niepokój wśród sprzedawców amerykańskich.

Dominacja rynkowa jest szczególnie ważna dla amerykańskich producentów dużych komputerów, wynosi 50% ich wszystkich dochodów ze sprzedaży zagranicznych. Eksport amerykańskiego sprzętu komputerowego przewyższa import w stosunku 15 do 1, a w 1977 r. osiągnie nadwyżkę netto w wysokości 2,8 mld dol. Technika komputerowa jest sprawą żywotną dla USA jako środek wyrównawczy kosztownego importu energii.

Japonia stanowi silne zagrożenie konkurencyjne dla przemysłu USA, przede wszystkim ze względu na zintegrowaną, nastawioną na eksport gospodarkę tego kraju. Firmy japońskie oferują pełne linie uniwersalnych systemów komputerowych i przeznaczają znaczne sumy na prace badawczo-rozwojowe.

Biorąc 1974 r. za rok wyjściowy, Ministerstwo Handlu porównało USA z wybranymi państwami obliczając udział bazy zainstalowanych komputerów w produkcji globalnym brutto. Dane wskazują na to,

że USA miały w 1974 r. bazę 168.000 zainstalowanych komputerów, stanowiącą 2,3% produktu globalnego brutto (tabela 2).

Tabela 1.

Zainstalowana baza systemów komputerowych na świecie (bez minikomputerów i systemów problemowo zorientowanych)

Obszar zainstalowanej bazy	% wartości zainstalowanej bazy dostarczonej przez firmy USA	
	ocena 1976	prognoza 1981 r.
Świat	87	81
USA	100	97
Kanada	95	92
Europa Zachodnia	86	81
Europa Wschodnia	5	10
Japonia	44	40
inne	85	80

Tabela 2.

Kraj	Produkt globalny brutto mld dol.	Wartość zainstalowanej bazy mld dol.	Zainstalowana baza jako % PGB	Ludność w mln.	Wielkość zainstalowanej bazy (systemy)	Liczba systemów na milion ludności
USA	1.413,2	33,6	2,3	211,9	168.800	797
Japonia	450,0	5,4	1,2	109,7	30.095	274
RFN	367,6	4,4	1,2	62,0	20.860	337
W. Brytania	173,1	3,5	2,0	56,2	15.520	276
Francja	282,7	3,5	1,2	52,5	13.064	249
ZSRR	710,0	4,5	0,6	252,1	12.500	50
ZSRR i 6 państw wschodnio-europejskich	955,2	5,6	0,6	356,9	15.435	43

Dla tego samego roku odpowiednie dane wykazały, że Japonia miała 30,095 systemów, równoważących 1,2% swego produktu globalnego brutto. W. Brytania miała 15,520 komputerów czyli 2% swego PGB, zaś RFN - 20,860 maszyn, tj. 1,2% swego PGB.

Największą część sprawozdania AFIPS poświęcona jest trzem zagadnieniom: dostawcom urządzeń i usług obliczeniowych, instytucjom użytkowym komputerów oraz usługom i personelowi stanowiącemu siłę roboczą komputerów.

AFIPS wskazują na to, że ilość pieniędzy wydanych przez użytkowników przypadająca na głowę, podwoiła się w okresie od 1970 r. do 1975 r.; tendencja ta ma utrzymać się do 1985 r.

Podobnie dochody producentów komputerów i sprzedawców usług podwoiły się w okresie pięcioletnim kończącym się w 1976 r. i mają się podwoić ponownie do 1981 r. Według danych AFIPS rynek minikomputerów i systemów problemowo zorientowanych wzrósł czterokrotnie od 1971 r. do 1976 r. i tak samo wzrośnie do 1981 r. Podczas gdy stosunek koszt/wydajność sprzętu poprawiał się 100-krotnie w każdym dziesięcioleciu od 1955 r. i powinien to czynić w dalszym ciągu przynajmniej w okresie najbliższego dziesięciolecia, problemy ulepszenia oprogramowania nie zostały rozwiązane. Produktowność programisty poprawia się tylko o 3% rocznie.

Problem ten mógłby być rozwiązany, utrzymuje sprawozdanie, gdyby nowa technika programowania mogła zmniejszyć potrzebę szkolenia programistów i zastąpić szkolenie użytkowników w zakresie użytkowania programowanych urządzeń końcowych. Problem oprogramowania, według sprawozdania, jest krytyczną ścieżką przetwarzania danych w następnym dziesięcioleciu.

Computer Weekly 1977 nr 578

PONAD 125,000 KOMPUTERÓW ZAINSTALOWANO W RFN

W dniu 1.07.1977 na terenie RFN było zainstalowanych 125.804 różnej wielkości komputerów o wartości 31,28 mld DM. Dane te podane w najnowszym wydaniu "Diebold Management Reports". Od dnia 1.1.1977 stan komputerów powiększył się o 11.080 systemów. Z ogólnej liczby zainstalowanych komputerów 16,701 (23,4 mld DM) przypada na komputery uniwersalne, 16,530 (2,7 mld DM) - na komputery do sterowania procesami i minikomputery, 92,573 (5,2 mld DM) - na małe systemy komputerowe i systemy zbierania danych. Największy wzrost zaobserwowano w instalacji małych systemów komputerowych (+7.668) po nich następują komputery do sterowania procesami (+2.242) i komputery uniwersalne (+1.170). Jak z tego wynika największa dynamika wzrostu utrzymuje się na rynku małych systemów komputerowych i urządzeń zbierania danych. Uniwersalne systemy komputerowe stanowią jak i poprzednio 75% ogólnej liczby instalacji. Interesujące jest to, że tylko 46,8% wszystkich komputerów do sterowania procesami i minikomputerów zainstalowano w przemyśle. Inne dziedziny zastosowań: instytucje i zakłady naukowo-badawcze i dydaktyczne 23,4%, komunikacja i usługi 13,1%, zarządzanie 6,3%, banki i firmy ubezpieczeniowe 3,8%, pozostałe 6,6%.

Elektronik 1977 nr 11

KOMPUTERY Z PRAGI DLA CAŁEJ CZECHOSŁOWACJI

Czechosłowackie zakłady "Tesla" wyprodukowały już 130 komputerów "Tesla 200" i "Tesla 300". Pracują one w przedsiębiorstwach i instytucjach 44 miast; w samej Pradze jest ich 26, w Brnie 9 a 7 w Bratysławie. Jednym z głównych użytkowników tych urządzeń jest Czechosłowacki Urząd ds. Obliczeń i Organizacji; jeszcze w tym roku zakupi on dla swych ośrodków 24 komputery. Duże znaczenie mają systemy do sterowania procesami szczególnie komputer "Tesla RPP-16". Czechosłowacka gospodarka narodowa została wyposażona w 1976 r. w 85 systemów do sterowania o wartości 650 mln. koron.

Rechentechnik Datenverarbeitung 1977 nr 4

FIRMA IBM WPROWADZA SYSTEM/34

IBM General System Division wprowadza tani system rozproszonego przetwarzania danych - System/34. Podstawowa konfiguracja systemu zawiera jednostkę centralną, pamięć dyskową o pojemności 8,6 megabajtów i jednostkę pamięci na dyskach elastycznych. Jednostka centralna ma pamięć MOS o pojemności 32.768 słów z czasem dostępu 600 ns i urządzenie sterujące pozwalające na przyłączenie do ośmiu monitorów ekranowych lub drukarek stolikowych. Kompatybilny z dawniejszym wolnostojącym małym komputerem System/32, System/34 kosztuje 34.700 dol. Pierwsze dostawy zaplanowane są na styczeń 1978 r.

Electronics 1977 nr 9 s. 36

SZYBKA DRUKARKA ELEKTROSTATYCZNA

Elektrostatyczna drukarka wierszowa Model 8200 firmy Houston Instrument (USA) drukuje z szybkością 2,400 wierszy/min. linie o długości 80 znaków oraz z prędkością 1,400 wierszy/min. 132-

znakowe linie. Urządzenie przenosi obraz drukowanego wiersza na papier za pomocą grafionów ustalanych w dwóch rzędach. Grafiony są formowane przez wytrawienie płytki drukowanej z zastosowaniem innych procesów mechanicznych stosowanych w produkcji drukarek elektrostatacznych. Metoda ta umożliwia 10-krotne zmniejszenie kosztów produkcji tego komponentu w porównaniu z innymi drukarkami liniowymi.

Znaki mają postać 7-bitową ASCII (7 bitów danych + 1 bit parzystości). Wprowadzane do bufora drukarki są następnie przenoszone na papier w postaci ładunków elektrostatacznych. Do tworzenia obrazu kropkowego wykorzystuje się pamięć ROM. Obraz wiersza umieszczony jest na matrycach o wymiarach 7 x 9 dla każdego znaku. Przy długości linii równej 132 znaki 924 grafiony wymagają 132 bitów do pokrycia całej szerokości.

Bit wejściowy podawany jest do grupy napędowej, z których każda napędza kilka grafionów. Obraz powstaje wtedy, gdy papier przechodzi obok walca pigmentującego (toner supply bar). Napęd papieru jest prosty - szybki silnik krokowy (stepper) jest bezpośrednio sprzężony z częścią ruchomą. Urządzenie nie jest wyposażone w żadne koła zębate, światłoczułe koła ani w silniki prądu stałego wyposażone w szczotki. Papier można otrzymać od firmy Houston Instrument lub od innych dostawców.

Cena urządzenia 80-kolumnowego o prędkości 2,400 wierszy/min. wynosi 3,000 dol. i 3,785 za urządzenie o prędkości 1,400 wierszy/min.

Control Engineering 1977 nr 4

STÓL KREŚLARSKI X-Y W 4 KOLORACH FIRMY HEWLETT-PACKARD

Stół kreślarski HP 9872, sterowany mikroprocesorem o wymiarach A3, (27,9 x 43,2 cm) w sposób automatyczny pod kontrolą programu, wybiera dowolne z czterech piór o różnych kolorach. Zaprojektowane do współpracy z kalkulatorem HP 9825 i HP 9831 zapewnia kreślenie w czterech kolorach siedmiu wzorów linii przerywanych oraz pięć wbudowanych wzorów znaków. Znaki definiowane przez użytkownika oraz określanie symboli umożliwiają tworzenie wyraźnych i łatwych do odczytania wykresów wtedy, gdy krzywe i wykresy są trudne do rozróżnienia. Wbudowanych w mikrokomputer 38 instrukcji zapewnia zmianę punktu na cyfrę, etykiotowanie, sortowanie znaków wg wielkości bezpośrednio przez układ pośredniczący HP-1B. Silnik krokowy napędu zapewnia adresowane ruchy co 0,0025 mm. Zdolność rozdzielcza wynosi 0,008 mm. Szybkość pisania może być zmieniana od 10 mm/s do 360 mm/s w przyrostach 10 mm/s.

Computer Design 1977 nr 4

KOMPUTER STOŁOWY DLA ŚREDNIEJ TECHNIKI PRZETWARZANIA DANYCH

Firma Wang opracowała nowy komputer stołowy PCS-II w cenie sprzedaży poniżej 20,000 DM. W cenie zawarty jest koszt jednostki pamięci na dysku elastycznym. Komputer ten różni się od wszystkich modeli konkurencyjnych możliwością bezpośredniego dostępu do danych i programów. PCS-II jest wyposażony w monitor ekranowy o pojemności ekranu 1.024 znaki i pamięć 8 kbajtów, którą można rozbudować do 32 kbajtów.

Elektronik 1977 nr 6

KOMPUTER BASIC DLA HOBBYSTY

Atrakcyjnie niska cena, oto najjaśniejsze określenie urządzenia, które będzie sprzedawane w Europie przez firmę Commodore. Firma ta oferuje za 2,500 DM tzw. PET-komputer z wbudowanym magnetofonem kasetowym i monitorem, zaprogramowany w wersji 8kb rozpowszechnionego języka wyższego rzędu BASIC. Translator języka BASIC jest zapamiętany w pamięci ROM i nie trzeba go po włączeniu ładować na nowo. Użytkownik ma do dyspozycji 4kb, a na życzenie nawet 8k bajtów pamięci RAM

dla programu użytkownika. Dodatkowy moduł pamięci RAM o pojemności 4 K bajtów zawiera system operacyjny urządzenia. Na monitorze może się ukazywać 25 wierszy po 40 znaków, zbudowanych na matrycy 8 x 8 punktów. Oprócz 64 znaków ASCII można przedstawić taką samą liczbę symboli graficznych, co jest szczególnie atrakcyjne w zastosowaniach hobbystycznych, ale może być również wygodne w rozwiązywaniu problemów technicznych. Przy programowaniu istnieją różnorodne możliwości, np. wstawianie lub wymazywanie znaków. Napisane programy są magazynowane na kasetach magnetofonowych i mogą być w każdej chwili powtórnie odczytane.

Można nabyć także wiele kaset z nagranyymi gotowymi programami, między innymi z symulacją krajoobrazu księżycowego i grą w karty "Czarny Piotruś".

Mimo że niska cena urządzenia wskazuje na hobbystę jako potencjalnego nabywcę komputera, komputer ten ma na wskroś "profesjonalne" właściwości. Dowodzi tego wyprowadzone na zewnątrz złącze, zgodne z normą IEEE 488 (zastępujące magistralę IEC) umożliwiające przyłączenie licznych urządzeń zewnętrznych.

Inną cechą, która zainteresuje profesjonalistę jest dostęp do języka maszyny: we wnętrzu znajduje się mikrokomputer KIM-1 z mikroprocesorem 6502 firmy MOS Technology Inc. (od niedawna filia firmy Commodore).

Elektronik 1977 nr 6

OBNIŻKA KOSZTÓW NOWYCH MODUŁÓW PAMIĘCI

W 1990 r. lub wcześniej nowe serie pamięci szeregowych staną się tańsze od dysków elastycznych, taśmowych pamięci kasetowych lub pamięci dyskowych z ruchomymi głowicami. Według przewidywań firmy McKintosh Consultants Co. w połowie lat osiemdziesiątych wejdą na rynki amerykański i zachodnioeuropejski m.in. pamięci ze sprzężeniem ładunków (CCD), pamięci domenowe.

Koszt wytworzenia jednego bitu nowych pamięci będzie spadać, jak przewiduje analiza, w miarę zwiększenia pojemności mikromodułów i opanowania technologii produkcji. Koszt np. obecnie wytwarzanych mikromodułów 64-kilobitowej pamięci na magnetycznych domenach cylindrycznych kształtuje się na poziomie 0,1 centa za bit, ale przewiduje się obniżenie do 0,001 centa za bit dla mikromodułów o pojemności 1 miliona bitów, które powinny znaleźć się na rynku w 1985 r. Także zanosi się na dużą obniżkę dla urządzeń ze sprzężeniem ładunków - z 0,04 centa za bit dla obecnych 64kb wersji do 0,003 centów za bit dla mikromodułów 265kb w 1985 r.

Electronics 1977 nr 7

FIRMA NATIONAL SEMICONDUCTOR CORP. PRZYGOTOWUJE PAMIĘĆ STAŁĄ O POJEMNOŚCI 65kb, PRZEWIDUJE PAMIĘĆ 262k W KOŃCU 1978 R.

Firma National Semiconductor Corp. wprowadza na rynek próbki wielkiej nowej pamięci stałej, planowanej do produkcji pod koniec lata 1977 r. Z pamięcią o pojemności 65.536 bitów wyprzedza ona pamięci stałe 32.768-bitowe ostatnio wprowadzone przez firmy: Electronics Arrays, General Instrument, Mostek i Texas Instruments.

W chwili obecnej dla firmy National jest to dopiero początek. W końcowej fazie projektowania znajduje się 131.072-bitowa pamięć stała, którą firma spodziewa się przygotować w ilościach próbnych pod koniec 1977 r. Pamięć o pojemności 262.144 bitów jest w przygotowaniu pod koniec 1978 r.

Dla pamięci stałej o pojemności 65kb firma National stosuje nowo opracowany proces samowyrównawczy bramki metalowej n-kanalowej MOS. Zawierając równoważnik 75 000 do 80 000 tranzystorów, 65kb pamięć stała jest wciśnięta w mikromoduł o powierzchni zaledwie 39 000 mcali^2 - tylko o 5 - 10% większej od powierzchni pamięci stałych o połowę mniejszej gęstości (moduł 32kb firmy GI ma powierzchnię 36 000 mcali^2). Ale gęstość - to połowa historii. Czas dostępu całkowicie statycznego mikromodułu 28-szpilkowego mieści się w zakresie 300 - 450 ns (prawie równym czasowi modułów pamięci 16.384-bitowych) i 350-500 ns (krótszy niż dla większości całkowicie statycznych 32kb pa-

mięci stałych). Pamięć ta wymaga zasilania 5V, a jej pobór mocy wynosi ok. 6 uW/bit, ok. 1/5 poboru mocy wymaganego dla pamięci stałych o pojemności 8.192 bity.

Aby uzyskać tę kombinację małej mocy, dużej szybkości i gęstości, zastosowano bramki metalowe, a nie krzemowe ze względu na ich zalety połączeń wzajemnych. Technika ta zmniejszyła średnią wielkość komórki w układzie pamięci stałej z ok. 0,5 mcala² do ok. 0,25-0,29 mcala², dając w wyniku zmniejszenie mikromodułu o ok. 20 - 30%.

Aby uzyskać dalsze zmniejszenie powierzchni mikromodułu o 10-20%, przyjęto niektóre rozwiązania projektowe w zakresie układów zewnętrznych opracowane dla pamięci z dostępem swobodnym, m.in. zastosowano różniczkowy wzmacniacz odczytu, który jest bardziej czuły niż prosty przekształtnik w starszych pamięciach stałych.

Wahania napięcia w układzie nie są tak duże jak w starszych pamięciach stałych (tylko 0,5 V wobec 4-5 V).

Electronics 1977 nr 8

FIRMA FAIRCHILD URUCHAMIA PRÓBNĄ PRODUKCJĘ PAMIĘCI ZE SPRZĘŻENIEM ŁADUNKÓW O POJEMNOŚCI 65536 BITÓW

Firma Fairchild zaanonsowała wypuszczenie na rynek próbnej serii pamięci ze sprzężeniem ładunków (CCD) o pojemności 65.536 bitów. Początkowym terminem był pierwszy kwartał 1977 r., ale opóźnienie zostało spowodowane postanowieniem firmy Fairchild umieszczenia pamięci w pakiecie o szerokości 300 mcala - mówi Robert Dwyer, dyrektor ds. marketingu dla urządzeń typu MOS i ze sprzężeniem ładunków. Cena w liczbie 100 sztuk wyniesie 96,61 dol. za moduł. Również firma Texas Instruments ogłosiła już wcześniej, że w kwietniu przedstawi tej samej wielkości próbki pamięci ze sprzężeniem ładunków.

Electronics 1977 nr 7

"DYSK ELASTYCZNY" W TECHNICIE CCD

CCDISC-Memory-System amerykańskiej firmy Alpha Data Inc. funkcjonuje podobnie jak dysk elastyczny; jest jednak znacznie szybszy i nie zawiera części poruszanych mechanicznie. Jest to rozbudowywalna pamięć CCD, o pojemności równej pamięci elektromechanicznej, ze średnim czasem dostępu wynoszącym 250 ns i w związku z tym 40 razy szybsza. System można rozbudowywać stopniowo, po 128 kbajtów, aż do osiągnięcia maksymalnej pojemności pamięci 1024 kbajty; posiada on poza tym (podobnie jak na dysku) 16 do 128 "ścieżek", z których każda zawiera 4096 słów 16-bitowych. Pamięć CCD umieszczona jest w 19-calowej obudowie (18x48x51 cm). System CCDISC spełnia wysokie wymagania dotyczące szybkości przekazywania danych: wartość wynosi maksymalnie 4 Ml/s. Sterowanie można dostosować do wielu typów minikomputerów za pomocą odpowiednich układów interfejsów, przy czym wyprowadzone wyjścia dostosowane są do techniki TTL. Podsumowując, jest to produkt mogący stać się poważnym konkurentem dla dysków elastycznych w różnych dziedzinach zastosowań.

Elektronik 1977 nr 1

PAMIĘĆ DOMENOWA FIRMY ROCKWELL O POJEMNOŚCI 1 Mbit

Firma Rockwell International opracowała największą ze znanych płytkę pamięci domenowej o pojemności 1 Mbit wytworzoną metodą fotolitograficzną. Wymiary płytki wynoszą 400x400 mcala, przy średniej wielkości komórki 8 µm oraz domen 18 µm. Adresowanie danych jest blokowo przy minimalnej wielkości bloku 512 bitów.

Omawiana płytka pamięciowa przedstawia znaczny wzrost gęstości zapisu informacji w stosunku do poprzednich opracowań produkowanych przez firmę Rockwell dla NASA. Może to być głównym czynnikiem

ktem zwiększającym ekonomiczność technologii domenowych, ponieważ znaczną część kosztów urządzenia stanowi podłoże opitaksjalnego granatu. Zgodnie z informacją firmy, nowa konstrukcja układu pozwoliła na czterokrotne zwiększenie gęstości osiągnięte za pomocą konwencjonalnego procesu fotolitograficznego o zdolności rozdzielczej $1 \mu\text{m}$. Jest to czynnik bardzo ważny, ponieważ takie technologie jak litografia strumienia elektronów czy rentgenowska są kosztowne.

Electronic Engineering 1977 nr 594

KASETOWA PAMIĘĆ DOMENOWA FIRMY FUJITSU

Firma Fujitsu opracowała pamięć domenową w rozwiązaniu kasetowym. Pamięć ta zawiera cztery płytki o pojemności po 73 kilobitów i jest zrealizowana w konstrukcji pozwalającej na zamienność ze sprzętem innych producentów. Pojemność maksymalna pamięci wynosi 294.912 bitów. Średni czas dostępu dla pętli obejmującej pojedynczą płytkę domenową, wynosi 290 milisekund lub 36 milisekund przy połączeniu na mniejszą pętlę. Pobór mocy w czasie pracy kasety wynosi 8,5 W. Wymiary kasety wynoszą 5,2 cm długość, 5,9 cm szerokość i 1,7 cm wysokość, a ciężar 210 gram. Kaseta może być wyjmowana ponad 20.000 razy i może być stosowana podobnie jak dysk elastyczny, taśma papierowa, czy też kasetowa pamięć taśmowa w inteligentnych urządzeniach końcowych i systemach mikrokomputerowych.

Electronics 1977 nr 13

RDZENIOWA PAMIĘĆ STAŁA W ZASTOSOWANIACH PRZEMYSŁOWYCH

Wzrastające zastosowanie mikroprocesorów w sterowaniu produkcją stwarza potrzebę posiadania pamięci, której zawartość nie podlegałaby zniszczeniu na skutek przerw w zasilaniu oraz zapewniającej niezawodną pracę przy wysokim poziomie zakłóceń elektrycznych. Dla tych zastosowań firma Controllex Corp., (USA) wprowadziła 4 kbitową pamięć stałą na rdzeniach ferrytowych z elektryczną wymianą informacji (EAROM). Pamięć ta jest przystosowana do 8-bitowego systemu szyn przesyłających mikroprocesorów. Pamięć ta oznaczona CM 203, wykorzystuje konwencjonalne rdzenie, ale stosuje opatentowaną metodę uzyskiwania nieniszczącego odczytu (NDRO). Wykorzystując odczyt NDRO, w przeciwieństwie do klasycznej pamięci rdzeniowej, gdzie dane muszą być zregenerowane po odczycie, w pamięci EAROM można selektywnie czytać lub pisać pod dowolnym adresem, bez naruszania informacji przechowywanej w innych miejscach. Pamięć CM 203 w ten sposób dodaje swój nieniszczący odczyt do uznanej własności pamięci rdzeniowych nie tracenia informacji, po wyłączeniu zasilania oraz jej odporności na zakłócenia elektryczne. Wszystkie te cechy razem, pozwalają pamięci CM 203 pracować jak prawdziwa pamięć nieniszcząca operacyjna z prostym przełącznikiem przekształcającym ją w pamięć, która może być pamięcią operacyjną (RAM) lub pamięcią stałą z elektryczną wymianą informacji. Omawiana pamięć ma przewagę nad pamięciami półprzewodnikowymi dla zastosowań przemysłowych, gdyż dynamiczne pamięci operacyjne muszą mieć odnawianą informację i zarówno one jak i pamięci statyczne tracą dane po wyłączeniu zasilania. Omawiana pamięć o czasie dostępu 350 ns jest konkurencyjna z punktu widzenia szybkości z pamięcią półprzewodnikową. Inne dane pamięci to: czas cyklu 1 ns, zarówno przy odczycie jak i przy zapisie, wejścia i wyjścia na poziomach TTL, napięcia zasilania +5 oraz ± 120 V, zakres temperatury pracy 0 i 70°C . Pamięć jest całkowicie zmontowana na pakiecie z obwodem drukowanym o wymiarach 8,5 x 12 cali. Cena pamięci CM 203 wynosi 500 dol. przy zakupach w dużych liczbach.

Electronics 1977 nr 3

MIKROMODUŁ NA PLYTCE

Firma National Semiconductor wystawiła na Targach w Hannoverze 8-bitowy mikroprocesor SC, MP na jednym mikromodule oraz płytki pamięci w formacie europejskim. Jest to nowa rodzina kompatybilnych mikroprocesorów.

Zestaw obejmuje: płytkę jednostki centralnej z modułem zastosowaniowym ISP-8C/100 i mikroprocesorem 8C/MP, układy impulsów taktowych i sterowania, 8-bitową pamięć zapis/odczyt, podłączenie dla 512-bajtowej pamięci PROM lub kompatybilną w podłączeniu pamięć ROM oraz układy buforowe. Płytki tych układów ma rozmiary 100 x 160 mm i ma jeszcze 8-bitową trzystanową magistralę danych i sterowania oraz 16-bitową magistralę adresów. Tak rozszerzona magistrala adresów umożliwia wybór 16 pamięci i także określonych komórek wewnątrz jednej pamięci.

Na płycie ISP 8C/100 E umieszczono dwa niezależne układy wejścia/wyjścia do szeregowego przetwarzania danych. Trzy sterowalne programowo impulsy sterowania na wyjściu i dwa programowo sterowane sygnały odczytu umożliwiają zwykle sterowanie urządzeniami zewnętrznymi przez przewód asymetryczny. Mikroprocesor 8C/MP ma repertuar rozkazów obejmujący 46 rozkazów jedno- i dwubajtowych. Procesor ma 5 różnych pamięci i rodzajów adresowania urządzeń zewnętrznych do programowania pamięci. Cztery 16-bitowe rejestry adresowe redukują długość części adresowej, a także umożliwiają wybór podprogramów. Płytki ISP-8C/004 E-RAM ma statyczną pamięć zapis/odczyt o pojemności 4 k słów ośmiobitowych z układami sterowania do kodowania adresów i do obustronnej transmisji danych przez trzystanową magistralę. Dwie kolejne płytki pamięci służą do rozszerzenia pojemności pamięci PROM i ROM/PROM. Są to: ISP-8C/004 E i ISP 8C/004BE-ROM/PROM. Obie płytki mają kompletne sterowanie, logikę wyboru modułu oraz układy wejścia/wyjścia.

Computer Zeitung z dn. 20.04.1977

NOWY MIKROKOMPUTER 8085 FIRMY INTEL

Firma INTEL rozpoczyna pierwsze dostawy systemu mikrokomputerowego 8085. Będąc rozszerzeniem systemu 8080, nowy system wykonuje programy systemu 8080 bez modyfikacji i jest wymienny szynowo z elementami systemu 8080. Wg firmy INTEL, system 8085 ma o 50% wyższy współczynnik koszt-wydajność od systemu 8080. Typowy 3-mikromodulowy system 8085 może zastąpić do 10 lub więcej elementów systemu 8080. Szybkość pracy przy pamięci o standardowej szybkości wynosi 3 MHz zamiast 2 MHz; nowe rozwiązania zwiększają wydajność więcej niż o 50% w pewnych zastosowaniach. Wykorzystuje on pojedynczy zasilacz 5 V, wspólny dla wszystkich części składowych wraz z programowanymi pamięciami stałymi, podczas gdy system 8080 wymaga trzech zasilaczy dla pamięci stałej (+5, +12, -5 V). Nowy system charakteryzuje się prostszym projektowaniem - każda część składowa 8085 może zastąpić grupę trzech do pięciu części 8080, pamięci buforowe wymagane są tylko w systemach największych, marginesy synchronizacji są znacznie szersze.

Nowy system zawiera liczne elementy systemu mikrokomputerowego MCS-85, wymienne z urządzeniami MCS-80 oraz wiele urządzeń pomocniczych. Urządzenia pomocnicze są nowymi wersjami urządzeń stosowanych przy projektowaniu oprogramowania i sprzętu systemu 8080. Podzespół MCS-85 są urządzeniami typu MOS/LSI pozwalającymi na zbudowanie całego systemu z trzech tylko urządzeń: jednostki centralnej 8085, pamięci z dostępem swobodnym 8155 o pojemności 2 K i układem wejścia/wyjścia i zegara. Może być również użyta pamięć stała 8355 o pojemności 16 K wraz z urządzeniem wejścia/wyjścia lub wymienna kasowalna programowalna pamięć stała 8755 o pojemności 16 K wraz z układem wejścia/wyjścia.

Te trzy urządzenia zapewniają system 5-woltowy z jednostką centralną, 256 bajtami pamięci odczytu-zapisu, z 2-k bajtami pamięci programów, 38 programowalnymi, równoległymi liniami wejścia/wyjścia, szeregowymi bramkami wejścia/wyjścia, zegarem systemu, systemem, urządzeniem sterującym, sterowaniem wielopoziomowym, przerwaniami wektorowymi oraz programowalnym zegarem i licznikiem zdarzeń. Typowym czasem cyklu rozkazów jest 1,3 μ s. Wydajność systemu pokrywa się częściowo z wydajnością układów logicznych TTL pozwalając na wykorzystywanie systemu do zastosowań o dużej wydajności w zakresie od urządzeń sterujących do 8-bitowych mikrokomputerów. Projektowanie systemu jest wspomagane przez system projektowania mikrokomputerów Intellec.

Computer 1977 nr 2

ADVANCED MICRO DEVICES (AMD) ROZPOCZYNA PRODUKCJĘ MIKROKOMPUTERÓW MCS-85

Firma Advanced Micro Devices (AMD) producent mikrokomputerów z Santa Clara (USA) będzie drugim źródłem dostaw układów MCS-85. Firma AMD, obecnie drugie główne źródło dostaw mikroprocesorów 8080, rozpocznie dostawę jednostki centralnej 8085 oraz innych części podstawowego zestawu. Aby pokonać początkowy opór firmy AMD przed wydaniem funduszy potrzebnych na skopiowanie 5-woltowej rodziny MCS-85, firma INTEL zgodziła się udzielić pomocy technologicznej i konstrukcyjnej: AMD otrzymała zestaw masek technologicznych dla MCS-85 oraz taśmy testowe w zamian za przekazanie firmie INTEL niewymienionych konstrukcji na 8-bitowe układy peryferyjne. Jednocześnie uzgodniono dostawę z drugiego źródła, z wymianą niektórych szczegółów technologii, z firmą SIEMENS AG na rozprowadzanie rodziny mikrokomputerów MCS-85. Również japońska firma NIPPON Electric Corp. będzie dostawcą mikrokomputerów MCS-85, ale bez porozumienia co do wymiany technologii. Pierwsze dostawy mikrokomputerów pojawiają się w roku 1978.

Electronics 1977 nr 13

UNIVERSALNY ASSEMBLER DLA MIKROPROCESORÓW

W centrum Obliczeniowym Instytutu Techniki Regulacji (Berlin NRD) opracowano uniwersalny assembler dla mikroprocesorów, który nazwano MICRO/MACRO-CROSS-ASSEMBLER KRS 4200. Assembler składa się ze słownika mnemotechnicznych symboli kodów rozkazowych, dostosowanego do wszystkich mikroprocesorów o długości słowa od 14 do 16 bitów. Oprócz tego uniwersalny translator umożliwia uzgodnienie makrorozkazów systemu.

Dotychczas założono słownik mnemotechniczny dla następujących mikroprocesorów:

- INTEL 8008 CROSS-ASSEMBLER KRS 4201
- INTEL 8080 CROSS-ASSEMBLER KRS 4201
- U 808 CROSS-ASSEMBLER KRS 4201
- BM/NSTB 716 CROSS-ASSEMBLER KRS 4201.

Assemblery pracują na mikrokomputerze KRS 4200/4201. Wymagają one pamięci ferrytowej o pojemności przynajmniej 8 K słów, czytnika taśm dziurkowanych i dziurkarki taśm. Planuje się w przyszłości podłączenie drukarki szeregowej, czytnika kart dziurkowanych i jednostek taśmy magnetycznej.

Rechentechnik Datenverarbeitung 1977 nr 4

URZĄDZENIE KONTROLUJĄCE I TESTUJĄCE DLA MIKROPROCESORÓW

Firma Fluke-Trendar przedstawiła cyfrowy system testujący typ T-3040-A umieszczający dokładnie zdefiniowany wzorec danych w przedmiocie badany (mikroprocesor na uzbrojonej płycie przewodzącej) z częstotliwością do 1,5 MHz; częstotliwość ta jest wystarczająca nawet dla szybszych mikroprocesorów przyszłości. Równocześnie urządzenie daje możliwość wprowadzania do obiektu badanego nie tylko stałych wzorców bitowych, lecz również automatycznie wytwarzanych sekwencji bitów z maksymalną częstotliwością 5 MHz. Przyrząd testujący ma poza tym nowe możliwości programowania: można za jego pomocą tworzyć programy, dotychczas tworzone tylko za pomocą systemów testowych wspomnianych komputerowo. W tym celu informacje programowe wprowadza się bezpośrednio do pamięci z klawiatury. Należy dodać, że obsługa programowania jest bardzo prosta, samo programowanie łatwe do opanowania, a ładowanie sekwencji rozkazów odbywa się z dysku elastycznego. Urządzenie "do zatrzymywania danych" pozwala na magazynowanie logicznych faz podczas przebiegu testu i wskazywanie błędów. System jest oferowany z 128 lub 240 połączeniami kontrolnymi; cena wynosi, zależnie od opcji, od 180 000 do 240 000 DM.

Elektronik 1977 nr 11

TESTER DO SPRAWDZANIA MODUŁÓW I PAKIETÓW

Wobec coraz szerszego pojawiania się na rynku takiego sprzętu cyfrowego jak urządzenia końcowe dla punktów sprzedaży lub punktów wymiany pieniędzy oraz rozpowszechniania się gier elektronicznych powstało zapotrzebowanie na tani przyrząd serwisowy, który pozwalałby na miejscu sprawdzić moduły cyfrowe, pakiety z układami hybrydowymi, układy scalone oraz płytki dużej skali integracji. Oddział Systemów Pomiarowych firmy Bendix Corp. uważa, że spełni to zapotrzebowanie przenośnym automatycznym testem modułów, który jest sterowany mikroprocesorem. Tester ten jest pierwszą rynkową pozycją Oddziału, który do tej pory produkował komputerowe systemy kontrolno-pomiarowe o wartości od 100 000 do 300 000 dolarów dla wojska i lotnictwa cywilnego. Omawiany tester wykorzystuje mikroprocesor oraz wymazywalną programowalną pamięć stałą (EPROM), co w efekcie dało mały (45 x 45 x 20 cm), lekki (15 kg), uniwersalny przyrząd, który pozwala na przeprowadzenie kontroli na stanowisku pracy badanego sprzętu. Przyrząd pomiarowy o nazwie "Herbie" wykorzystuje mikroprocesor 8080 firmy INTEL, w którym przebiega realizacja systemu operacyjnego, dokonuje się sterowanie kolejnością działań i za pomocą którego dokonywana jest automatyczna kontrola własnej pracy systemu po każdym włączeniu zasilania. Programy testujące dla systemu dostarczane są przez firmę BENDIX lub też wytwarzane przez użytkownika za pomocą biblioteki automatycznych generatorów testów (ATG). Biblioteka ta zawiera między innymi programy dla zmiany wyprowadzeń, stanów i poziomów. Programy są zapisane na pakietach pamięci stałej, aby uzyskiwać niezmiennie dane do wielokrotnego użycia. Niemniej jednak zastosowana jest pamięć stała 8192 bitów typ 2708 firmy INTEL, co pozwala na zmianę programów przez kasowanie światłem ultrafioletowym i ponowne zaprogramowanie na drodze elektrycznej w zwykły sposób. Pakiet z pamięcią stałą 5120 słów można podwoić przez zastosowanie pakietu z drugą pamięcią 2708. Wielkość pamięci może ulec dalszemu podwojeniu przez zastosowanie kostek pamięci 2708 o podwójnej pojemności. Dla bardzo dużych programów wymagających większej pamięci przewidziano również możliwość podłączenia pamięci kasetowej na taśmie magnetycznej.

Podstawowy tester modułów wyceniony na 10 000 dolarów ma wejście na 64 końcówki (kanały), które można rozbudować do 256 końcówek. Każdy z kanałów może współpracować z układami typu C-MOS, DTL, TTL oraz innymi układami logicznymi o napięciu 5 V. Tester może być używany również do badań diagnostycznych pakietów, modułów i urządzeń cyfrowych z pewnym ograniczonym zakresem wykrywania uszkodzeń. Wyposażony jest w odczyt cyfrowy 16-elementowy w kodzie znaków ASCII. Druga pamięć stała 2708 z wymazywaniem przewidziana jest do lokalizacji uszkodzeń. W szerszych badaniach nad uruchamianiem należy użyć jeszcze jednej pamięci 2708, co pozwala operatorowi wejść w kontakt konwersacyjny z testerem i wypracowywać programy testujące w czasie kontroli. Początkowe testery przewidziane są tylko do badania układów cyfrowych, ale będą miały przewidziane możliwości rozbudowy w kierunku testerów hybrydowych przez włożenie pakietów dla kontroli napięcia, zliczania częstotliwości i innych pomiarów analogowych.

Electronics 1977 nr 3

AUTORADAR Z MIKROPROCESOREM OSTRZEGA KIEROWCÓW

Firma Standard Elektrik Lorenz AG, niemiecka filia firmy ITT, przeprowadza badania antykolidyjnego systemu radarowego dla samochodów, opartego na mikroprocesorze. Promień 2° z radaru umieszczonego na przodzie samochodu wykrywa przedmioty na jego drodze. W centrum systemu znajduje się mikroprocesor zaprogramowany do rozróżniania obiektów ruchomych, jak np. ludzie, od nieruchomych, jak znaki drogowe. Tylko potencjalnie niebezpieczne przedmioty powodują alarm, który ostrzega kierowcę i wskazuje kierunek zagrożenia. Program zapewnia, że tylko echa znaczące są przetwarzane, zaś echa "nieznaczące" nie powodują alarmu ostrzegającego kierowcę.

Electronics 1976 nr 24

SYSTEM ZABEZPIECZENIA SAMOCHODÓW OPARTY NA MIKROPROCESORZE,
PRZYGOTOWYWANY NA ROK 1985

Samochód przyszłości sterowany elektronicznie zamierza opracować firma Minicars Inc. (USA). Firma przewiduje opracowanie deski rozdzielczej i układu hamulcowego z wykorzystaniem mikroprocesorów. Po całkowitym opracowaniu i wyprodukowaniu tego systemu (około 1985 r.) koszt samochodu wzrośnie tylko o 200 dolarów.

Deska rozdzielcza stanowi całkowity system informacyjny, który kontroluje wszystkie warunki operacyjne, a nawet może zahamować samochód w nagłym wypadku. Większość urządzeń deski rozdzielczej pochodzi z firmy RCA. Samochód przyszłości, budowany przez firmę Minicars, wykorzystuje dwa mikroprocesory COSMAC firmy RCA - jeden do elektronicznej deski rozdzielczej, drugi zaś do układu hamulcowego sprzężonego z radarem.

Zarówno informacja standardowa jak też i nowe dane dotyczące szybkości zużycia paliwa, szybkości obrotów/min. silnika, czas i długość przebytej drogi są przekazywane do mikroprocesorowej deski rozdzielczej. Kadłowe działanie powoduje, że na desce rozdzielczej zapala się migające światło ostrzegawcze przez 3 s co 30 s do chwili usunięcia wady. Monitorem ekranowym jest płyta plazmowa z 32 znakami. W celu uniknięcia zderzeń system radarowy sterowany przez mikroprocesor daje akustyczne sygnały ostrzegawcze w potencjalnie niebezpiecznych sytuacjach. Ale gdy nie można uniknąć poważnego zderzenia, system ten może automatycznie uruchomić specjalny układ hamulcowy w celu minimalizowania szkody.

Electronic Design 1977 nr 1

KOMPILATOR CORAL DLA MIKROPROCESORÓW F 100-L FIRMY FERRANTI

Firma Ferranti przygotowuje dla 16-bitowego mikroprocesora F 100-L kompilator - CORAL 66. Będzie on generował kody assemblera F 100. Firma planuje dwie wersje tego kompilatora: kompilator skrośny i kompilator rezydujący. Język CORAL opracowano dla problemów czasu rzeczywistego z zakresu sterowania procesami; rozpowszechniony jest szczególnie w Wielkiej Brytanii. Wielu producentów komputerów do sterowania procesami oferuje dzisiaj kompilator dla tego języka: Data General, Digital Equipment, Honeywell i Interdata.

Nadaniem należy, że CORAL 66 (Computer Outline Realtime Application Language), FORTRAN, LISP i PASCAL są obecnie rozważane przez ISO, jako propozycje opracowania standardowego języka programowania do sterowania procesami technicznymi.

Elektronische Rechenanlagen 1977 nr 4

PRZEMYSŁ OBRABIARKOWY I NARZĘDZIOWY ZSRR

Ministerstwo Przemysłu Obrabiarkowego ZSRR zatwierdziło podstawowe zadania Wszechzwiązkowego Zjednoczenia Przemysłowego "Sojuztjazzstankprom" na dziesięcioletnią pięcioletkę. Jako podstawowe zadania planu w 1980 r. przewiduje się produkcję towarową na sumę 739 mln rubli, co oznacza wzrost w stosunku do 1975 r. o 46%. Średnioroczne tempo wzrostu produkcji wyniesie 7,9%. Wydajność pracy w ciągu pięcioletki zwiększy się o 42%, co umownie jest równoważne pracy 30 tys. pracowników.

W latach 1976 - 1980 zwiększenie wydajności pracy spowoduje wzrost produkcji towarowej o 94% w porównaniu z 91,7% w dziesięcioletniej pięcioletce. Cały przyrost produkcji w działających przedsiębiorstwach będzie osiągnięty bez zwiększenia zatrudnienia. Przewiduje się wzrost liczby pracowników o 1955 ludzi. Będą oni kierowani do kompletowania nowych wydziałów i stanowisk pracy. W porównaniu z 1975 r. produkcja obrabiarek będzie zwiększona o 7,5% a wartość tej produkcji o 50,2%.

Istotne zmiany są zaplanowane w strukturze produkcji obrabiarek do skrawania metali. Wzrośnie produkcja dużych, ciężkich i unikalnych obrabiarek, obrabiarek ze sterowaniem programowym, w tym obrabiarek z magazynem narzędzi oraz centrów obróbkowych, automatycznych linii budowy maszyn ze

zwiększonym oprzyrządowaniem i innych nowoczesnych urządzeń. Do 1980 r. w porównaniu z 1975 r. produkcja obrabiarek do skrawania metali zwiększy się o 7,5%, w tym produkcja specjalnych obrabiarek o 12,7%, obrabiarek ze sterowaniem programowym o 65,3%, wielooperacyjnych obrabiarek z magazynem narzędziowym o 12,2% i obrabiarek z uproszczonym i cyklowym sterowaniem programowym ze wskaźnikami cyfrowymi o 7,2%; produkcja automatycznych i półautomatycznych linii dla przemysłu maszynowego będzie większa o 13,6%. Produkcja obrabiarek ze sterowaniem programowym w 1980 r. będzie stanowiła 30,8% ogólnej produkcji obrabiarek w porównaniu z 19,9% w 1975 r. Wartość tej produkcji w ciągu pięcioletki zwiększy się 2,3 raza. W 1980 r. powinno być wyprodukowanych o 21,3% więcej obrabiarek ciężkich i unikalnych.

W ciągu pięcioletki zaplanowano zwiększenie produkcji z państwowym znakiem jakości o 8,3 raza i osiągnięcie udziału tej produkcji o wysokiej jakości w ogólnej produkcji w 1980 r. do 48% w porównaniu z 8,4% w 1975 r. W bieżącej pięcioletce przewiduje się dalsze zwiększenie eksportu obrabiarek do skrawania metali oraz oprzyrządowania i części zamiennych. Dostawy eksportowe obrabiarek do skrawania metali powinny wzrosnąć w pięcioletce ilościowo o 21,1% a wartościowo o 22,3%. Zacznie się eksport seryjnie produkowanych obrabiarek ze sterowaniem programowym, w tym centrów obróbczych, obrabiarek ze sterowaniem programowym i magazynem narzędzi.

Jednak wzrost ilościowych wskaźników eksportu obrabiarek do skrawania metali, szczególnie obrabiarek ciężkich i unikalnych w bieżącej pięcioletce będzie ograniczony w związku ze wzrastającym zapotrzebowaniem tych obrabiarek w kraju. W podstawowych kierunkach rozwoju gospodarki narodowej ZSRR w latach 1976-1980 przewiduje się bowiem przyspieszony rozwój, m.in. metalurgii żelaza i stali, budowy maszyn ciężkich, budowy urządzeń energetycznych i innych.

W wyniku tego w dziedzinie przemysłu budowy obrabiarek przewiduje się rozwój produkcji obrabiarek ze sterowaniem programowym oraz rozwój produkcji obrabiarek ciężkich i unikalnych oraz pras. Przedsiębiorstwa Zjednoczenia do roku 1980 będą produkowały około 3 tys. obrabiarek ze sterowaniem programowym, w tym około tysiąca z magazynem narzędzi i typu centrum obróbczego.

Podstawowym kierunkiem rozwoju przemysłu obrabiarek ze sterowaniem programowym jest produkcja unifikowanych obrabiarek o szerokim technologicznym przeznaczeniu o różnym stopniu automatyzacji i z różnymi sposobami zamiany narzędzi i półwyrobów. Obrabiarki ze sterowaniem programowym będą wbudowane w zautomatyzowane systemy ze sterowaniem komputerowym. Wiele przemysłowych zjednoczeń ma za zadanie opracowanie i wyprodukowanie zautomatyzowanych systemów do obróbki korpusów i detali tocznych. Systemy te będą sterowane za pomocą komputerów.

Opracowanie układów zautomatyzowanych systemów w jednym ze zjednoczeń przemysłu budowy obrabiarek na bazie centrów obróbczych IR-500 i IR-800 wykazało celowość utworzenia modułów obróbczych na bazie tych centrów wyposażonych w magazyny półwyrobów. System sterowania zapewnia ciągłą pracę modułu obróbczego w czasie jednej lub kilku zmian. W miarę potrzeby moduły obróbcze mogą być łączone w linie sterowane za pomocą komputerów. Takie rozwiązanie zapewni jakościowo opracowanie konstrukcji obrabiarek, urządzeń i mechanizmów ładujących oraz możliwość komponowania zautomatyzowanych systemów z takiej liczby modułów obróbczych, która jest niezbędna dla danego programu produkcyjnego. Otrzymanie zautomatyzowanych systemów sterowanych za pomocą komputera zapewni znaczne (3-5 razy) zwiększenie wydajności, zmniejszenie liczby pracowników oraz stworzy przesłanki do przejścia na produkcję zautomatyzowaną.

Wymieniony wysokowydajny sprzęt ze sterowaniem programowym powinien mieć wysokie parametry techniczne: dokładność pozycjonowania od $\pm 0,001$ do $\pm 0,005$ mm; czas rozbiegu i hamowania 0,2 - 0,5 s; prędkość szybkich przesunięć do 10 - 12 m/min.; czas wymiany narzędzia 10 - 15 s.

Tak wysokie parametry techniczne obrabiarek zapewni zastosowanie nowoczesnych systemów sterowania programowego, silników z wysokimi momentami w napędach posuwów, silników prądu stałego w głównym ruchu, dokładnych czujników, niezawodnej hydrauliki oraz elementów konstrukcyjnych, kulowych par śrubowych, materiałów polimerowych i hydrostatyki na prowadnicach, nakładek, prowadnic hartowanych, stosowanie nowych materiałów konstrukcyjnych i in. Tradycyjne napędy posuwów i głównego ruchu mają skrzynie przekładni zębatych i ograniczony zakres regulacji. Elektryczne silniki o wysokim momencie umożliwią wyeliminowanie skrzyń przekładni zębatych i zapewnią szeroki zakres regulacji, znaczną szybkość działania i zwiększenie prędkości biegu jałowego do 10 -

12 m/min. Napędy prądu stałego zamieniają wielostopniowe skrzynie przekładniowe, pozostaną tylko zestawy z kilku kół zębatych.

Takie rozwiązania konstrukcyjne wymagają nowej bazy elementowej (wysokomomentowe silniki z blokiem zasilania i silniki prądu stałego z przetwornikami tyrystorowymi).

W Instytucie Elektrotechnicznym w Nowosybirsku opracowano podzespoły ruchu postępowego i obrotowego. Urządzenia te jak i wysokomomentowe silniki elektryczne znacznie upraszczają kinematykę obrabiarek, zmniejszają zużycie metalu i pracochłonność przygotowania oraz polepszają parametry techniczne obrabiarek. Przewiduje się połączenie produkcji podzespołów elektromechanicznych z napędami elektrycznymi.

Podstawowe zadania zakładów i biur konstrukcyjnych Zjednoczenia w dziesiętej pięcioletce są następujące:

- opanowanie produkcji uniwersalnych ciężkich i unikalnych obrabiarek z obwodowymi i kombinowanymi systemami sterowania programowego ze zmechanizowaną lub zautomatyzowaną wymianą narzędzi,
- wyprodukowanie prototypów ciężkich obrabiarek z minikomputerowymi systemami sterowania,
- wyprodukowanie dla głównych gałęzi przemysłu wydajnych specjalnych i wyspecjalizowanych obrabiarek, przeważnie ze sterowaniem programowym,
- wyprodukowanie sterowanych komputerowo centrów złożonych z ciężkich i unikalnych obrabiarek dla kompleksowego procesu obróbki.

Podstawowe zadania zapowiadające realizację powyższego programu to:

- przeprowadzenie prac naukowo-badawczych i doświadczalnie-konstrukcyjnych w zakresie głównych problemów budowy obrabiarek, np. badanie statycznej i dynamicznej sztywności obrabiarek, napędów, części, obudów o masie do 80 ton (spawanych i odlewanych), niemetalowych przewodniczących anty-frykcyjnych, hydrostatycznych przewodniczących ruchu prostoliniowego i przekładni, szybkich napędów elektrycznych o mocy do 18 kW;
- opanowanie urządzeń i rozwiązań opracowanych przez współpracujące branże: minikomputerowych urządzeń sterujących o dużej niezawodności, niezawodnej bezstykowej aparatury elektrycznej dla obwodów sterowania, wysokoczęstotliwościowych łożysk tocznych o średnicy od 2000 mm, w szczególności łożysk krzyżowo-wałeczkowych i dwurzędowych wałeczkowych; optycznych, dokładnych czujników kołowych, wskaźników przesunięć liniowych, urządzeń do chłodzenia oleju smarowego i innych.

W dziesiętej pięcioletce będzie wyprodukowanych 385 typów nowych obrabiarek, w tym 72 typy obrabiarek ze sterowaniem programowym, 264 serii przemysłowych, w tym 61 typów obrabiarek ze sterowaniem programowym, natomiast wycofanych z produkcji będzie 56 typów obrabiarek o przestarzałej konstrukcji.

Mechanizacija i Avtomatizacija Proizvodstva 1977 nr 5

KOMPUTEROWO STEROWANY SYSTEM OBRABIAREK "CONCEMA"

Coraz to większego znaczenia nabiera zastosowanie komputerów sterujących procesami do dyskretnego sterowania obrabiarkami. Do systemu CONCEMA-DNC, prezentowanego przez NRD na Targach Lipskich 1977 dołączone były następujące obrabiarki sterowane komputerowo: obrabiarka wielooperacyjna C-BKOZ 900/5, VEB Mikromat Dresden, obrabiarka wielooperacyjna C-BET 90/4, VEB WEMA UNION Karl-Marx Stadt, obrabiarka wielooperacyjna C-FKR5 250, VEB WEMA Auerbach.

Powyższe obrabiarki wielooperacyjne były sterowane komputerem KRS 4200/16K firmy VEB Robotron.

Komputer służył do:

- zarejestrowania programów NC w pamięci bębnowej (96 k słów odpowiada mniej więcej 700 programom NC) i ich transmisji do podłączonych obrabiarek,
- rejestrowania i wydruku danych operacyjnych obrabiarek, jak np. liczba gotowych części w każ-

do obrabiarek uporządkowanych wg numerów części programu, czas obróbki, czas nieefektywny wraz z podaniem przyczyn,

- maszynowego programowania obrabiarek za pomocą systemu programowania AUTOTECH/SYMAP, BOFR 3 do wiercenia i frezowania, DR4 i DR5 do toczenia.

W porównaniu z rokiem minionym dokonano następujących usprawnień:

- urządzenia NCP i DNC-monitory systemu CONCEMA-CNC 500 były po raz pierwszy przedstawione jako osiągnięcia firmy VEB Starkstom-Anlagenbau Karl-Marx-Stadt,
- przy monitorach, dotąd używane w rejestrowaniu danych operacyjnych przełączniki dokadowo zastąpiono klawiszami,
- do komputera KRS 4200 podłączono drukarkę mozaikową do wydruku instrukcji roboczych i programów sterujących,
- system AUTOTECH/SYMAP-BOFR3 rozszerzono o sterowanie kształtowe. Oprócz tego rozszerzono możliwości systemu o dalsze postprocesory. Oferuje się obecnie ponad 10 postprocesorów. W r. 1977 opracowano dalsze postprocesory do sterowanych kształtowo obrabiarek.

Do ekonomicznych efektów sterowania komputerowego DNC należy zaliczyć:

- zwiększenie wykorzystania czasu podstawowego maszyn OSN o 10% przez skrócenie technicznie uwarunkowanych czasów przestoju, jak również przez polepszenie sterowania i organizacji produkcji przy zastosowaniu komputera do rejestrowania danych operacyjnych,
- skrócenie czasów przygotowania i zakończenia o ok. 15% na skutek szybszej dyspozycyjności programu, korekty i doglądania programu przez komputer oraz eliminacji taśm dziurkowanych,
- skrócenie czasu programowania o ok. 60% przy użyciu komputera DNC do maszynowego programowania z BOFR3.

Messen, Steuern, Regeln 1977 nr 9

FIRMA PRIME ZAINTERESOWANA STEROWANIEM NUMERYCZNYM

Firma PRIME Computer przystosowuje swoje komputery serii PRIME 300 do programowania maszyn sterowanych numerycznie. Minikomputer PRIME 100 używany był dotychczas do graficznego przedstawienia taśm sterujących. Firma zajęła się programowaniem dla sterowania numerycznego na skutek ścisłej współpracy z producentami maszyn. Opracowano system programowania przystosowujący rozwiązania z dużych komputerów wyposażonych w język APT na małe komputery PRIME 300 i PRIME 400 umożliwiające wielu użytkownikom równoczesny dostęp do systemu komputerowego o pojemności pamięci 128 Kb.

Firma PRIME Computer stawia do dyspozycji pakiet programów PRIMAPT, za pomocą którego można przeanalizować przebiegi pracy maszyn sterowanych numerycznie nawet 2,5-osiowych. Język ten wyróżnia się uniwersalnością zastosowań i łatwą przyswajalnością. Zastosowanie małego komputera RHEE zmniejsza znacznie koszty. Równocześnie utrzymywane są funkcje dużego komputera.

Konfiguracja dla 3 użytkowników kosztuje wraz z językiem programowania PRIMAPT ok. 180 tys. DM. Dotychczasowe rozwiązania z minikomputerami umożliwiały pracę jedynie z jednym użytkownikiem i kosztowały od 70 000 do 150 000 DM zależnie od konfiguracji.

Elektronische Rechenanlagen 1977 nr 4

PROGRAM ROZWOJU KOMPUTEROWO WSPOMAGANEGO WYTWARZANIA

Firma Softech Inc. (USA) zawarła kontrakt z Laboratorium Materiałów Sił Powietrznych USA w ramach programu badawczego, dotyczący opracowania systemu zintegrowanego, komputerowo wspomaganego wytwarzania samolotów (Integrated Computer Aided Manufacturing) .ICAM. Program ten początkowo będzie rozwijany w zakresie części samolotów, które wymagają obróbki metali, ale obejmie on w przyszłości elektronikę podzespołów i montaż. System ICAM ma być skomputeryzowanym systemem, organi-

zującym każdy krok produkcji od projektowania części aż do jej spedycji. Oczekuje się, że system ten będzie nadawał się do produkcji jednostkowej, małoseryjnej i średnioseryjnej. System będzie miał możliwość optymalizacji kosztu.

W pierwszym etapie tego dwufazowego programu zamierza się przedstawić celowość systemu ICAM z punktu widzenia znacznego zmniejszenia kosztu produkcji części samolotów. Faza II obejmuje zdemonstrowanie zasad działania systemu ICSM na przykładzie technologii produkcji blach.

Podwykonawcami firmy Softech są producenci konstrukcji i silników samolotowych oraz uniwersytety. Działającymi w zakresie doradczym tego programu są producenci obrabiarek m.in. firmy Cincinnati Milacron oraz Kearny i Trecker.

Control Engineering 1977 nr 2

FIRMA HONEYWELL UZUPEŁNIA SYSTEM TDC O NOWE URZĄDZENIA

Oddział Sterowania Procesami firmy Honeywell uzupełnia swój rozproszony system sterowania procesem o dwa nowe urządzenia TDC 4500 i TDC 7100.

TDC 4500 jest komputerem do sterowania procesami zbudowanym na kilku mikroprocesorach. Zbudowany jest podobnie jak minikomputery level 6, z tą jednak różnicą, że minikomputer Level 6 ma słowo 16-bitowe, natomiast TDC 4500 pracuje na słowie o długości 24-bitów; oprogramowanie jest kompatybilne z poprzednim typem komputera firmy Honeywell TDC HS4400 i komputerami serii 4000.

TDC 4500 posiada czas cyklu pamięci równy 600 ns oraz czas indeksowania 250 ns. Półprzewodnikowa pamięć o dostępie swobodnym może mieć pojemność do 256 K; wyposażona jest w automatyczne urządzenia wykrywania i korygowania błędów, które zwiększają niezawodność pamięci operacyjnej pięciokrotnie w stosunku do poprzednich systemów. W celu zwiększenia szybkości, uwzględnione jest również specjalnie sprzętowe rozwiązanie zmiennego przecinka. Przetwarzanie testowe wykazuje szybkość większą od HS4400 o 1,7 razy.

TDC 7100 jest nową jednostką łączącą proces z systemem, zbudowaną na mikroprocesorze. TDC 7100 umożliwia przygotowanie sygnalizacji, wielokrotnianie i konwersję analogowo-cyfrową do 512 wejść analogowych lub 1 000 wejść cyfrowych rozmieszczonych w sioci o łącznej długości do 2.000 m. Zdalny multiplexer połączony z TDC 7100 przez parę kabli, pozwala wydłużyć szynę danych o następne 2000 m. TDC 7100 może zapożyczyć wszystkie wejścia dla urządzeń sterujących, mikroprocesora TDC 2000, dla sterowania zamkniętą pętlą przy bezpośrednim dostępie komputera w systemie komputerowym w celu centralnego rejestrowania danych, księgowania i monitorowania.

Control Engineering 1977 nr 4

FIRMA DEC URUCHAMIA SPRZĘT KOMPUTEROWY W FABRYCE

Firma Digital Equipment uruchomiła pakiet sprzętu i system sterujący, który wprowadza przetwarzanie rozproszone do zarządzania fabryką. Rozproszony system zarządzania fabryką oferuje zintegrowaną metodę zarówno do gromadzenia danych fabryki jak i sterowania procesami.

Decentralizowane wokół popularnych procesorów PDP 11 do sterowania procesami, systemy te zawierają nową serię urządzeń końcowych do gromadzenia danych fabrycznych, podsystemy miejscowej i zdalnej kontroli i sterowania procesami oraz nową szynę szeregową, zwaną "DEC Dataway", która jest wspólną drogą danych między komputerem centralnym i stacjami zdalnie położonymi. System ten będzie dostarczany przez firmę DEC dużym przedsiębiorstwom, które mogą opracować oprogramowanie we własnym zakresie.

Istotą systemu jest szyna szeregową DEC Dataway. Jest to skręcona para przewodów, mająca długość do 5 km od jednostki centralnej, która na swojej długości może być połączona z 63 punktami wejściowymi i wyjściowymi.

Szyna danych jest sterowana przez mikrokomputer LSI/11, który działa jak łącze między szyną

"Unibus" PDP 11 i szyną szeregową. Pierwsza faza dostaw rynkowych obejmuje systemy centralne DPM 60 i DPM 80 oparte na PDP 11/34 i PDP 11/70. Wyposażone są w system operacyjny RSX-11M oraz kompilatory języków programowania. W pierwszej partii produktów znajdują się trzy urządzenia końcowe do gromadzenia danych fabrycznych; będą one stosowane do rejestrowania danych dotyczących personelu lub produktów.

W ciągu roku wprowadzony będzie koncentrator końcowy do łączenia monitora ekranowego i urządzeń końcowych z szyną DBC Dataway. W przyszłości wprowadzone będą podsystemy oparte na programowanym mikrokomputerze z cyfrowymi i analogowymi urządzeniami wejścia/wyjścia dla zdalnej kontroli i sterowania procesami. Dostępne będzie nowe również oprogramowanie do przetwarzania transakcyjnego w celu szybkiego opracowywania danych do zarządzania fabryką.

Firma DEC przewiduje duży popyt na swój nowy system, np. w przemyśle maszynowym system ten może gromadzić dane podstawowe dla sprawozdawczości pracy, gospodarki magazynowej i dostępności siły roboczej. Systemy wejścia/wyjścia mogą kontrolować prasy, obrabiarki i transportery w czasie rzeczywistym, umożliwiając kierownikom otrzymanie dokładnego obrazu postępu prac, dostępności maszyn, dostaw materiałów, czasów przestoju.

W przemyśle elektronicznym firma DEC sugeruje, że systemy DPM mogą być stosowane do rozwiązywania problemów związanych z przechowywaniem materiałów i ich badaniem, przechowywaniem i transportem części na montaż, podsystemami sterującymi i kontrolującymi połączonymi szyną szeregową z tym samym komputerem centralnym, mogącym wykonywać automatycznie testowanie modułów i przesyłanie danych testowych do centralnej kartoteki.

Pierwsza instalacja tego nowego systemu to odlewnia firmy International Harvester, produkująca odlewy do maszyn rolniczych. System steruje piecem odlewni, układami wagowymi, magazynem i podsystemem zbierania danych produkcyjnych.

Firma DEC nie zapewnia dostaw gotowych systemów zarządzania produkcją. Klienci muszą pisać programy zastosowaniowe i instalować własne okablowanie. Wówczas mogą oni "podwiesić" różne urządzenia końcowe na szynie "dostawcy".

Typowa fabryka może stosować kilka rodzajów urządzeń: przetwornik analogowo-cyfrowy do przyjmowania danych z czujników, takich jak chromatografy lub termopary, trzy rodzaje stanowisk roboczych sterowanych za pomocą mikroprocesora, multipleksor danych, podukład wejścia/wyjścia złożony z mikrokomputera LSI-11 w obudowie przemysłowej i komputer główny fabryki PDP11/34 lub 11/70. Ilość danych 7,5 do 28 Mbajtów jest umieszczona w pamięci komputera głównego.

Stanowiska roboczo przyjmują standardową kartę identyfikacyjną na wejściu do systemu. Każde stanowisko ma także monitor ekranowy o 32 znakach alfanumerycznych, wskazujący pracownikowi, co zrobić przy każdym kroku pracy. Stanowiska robocze i multipleksor podłączone są do szyny "dostawcy" przez oprogramowanie techniczne mikroprocesora, zaś cały system posługuje się językami wysokiego poziomu, jak CODOL, FORTRAN i BASIC.

System typowy, obejmujący komputer główny PDP-11/34, podukład wejścia/wyjścia oparty na mikrokomputerze LSI-11, 10 podstawowych stanowisk roboczych, trzy stacje chronometrażu i nadzoru, trzy stanowiska robocze terenowe, drukarkę wierszową i dwa nadzorcze urządzenia końcowe z lampami oscyloskopowymi kosztuje ok. 200.000 dol.

Electronics Weekly 1977 nr 364

MIĘDZYNARODOWE SEMINARIUM NA TEMAT PRZEMYSŁOWYCH SYSTEMÓW KOMPUTEROWYCH

Czwarte doroczne Międzynarodowe Seminarium na temat przemysłowych systemów komputerowych odbyło się na Uniwersytecie Purdue w dniach 8 - 11 listopada 1976 r. (seminarium obejmuje odrębne spotkania prowadzone przez Instrument Society of America - ISA i Uniwersytet w Purdue). Różnorodna działalność komitetu jest ukierunkowana na opracowanie standardów i wytycznych dotyczących stosowania centralnej rejestracji danych i systemów sterowania w czasie rzeczywistym.

Sesje obejmowały prezentację problemów technicznych i oprogramowania, dyskusję na temat projektowania oraz wymianę doświadczeń eksploatacyjnych.

Komitet Techniczny TC 1 ds Przemysłowego Fortranu Czasu Rzeczywistego był wyróżniony przez ISA za opracowanie standardu ISA, zatytułowanego "Procedury języka FORTRAN w przemysłowym systemie komputerowym dla funkcji wykonawczych procesów wejścia/wyjścia i manipulacji bitowej". Jest to zrewidowana wersja standardu wydanego w 1972 r. Komitet ds FORTRANu kontynuował prace nad propozycją standardu do sterowania działaniami zbieżnymi. Punktem szczególnego zainteresowania były procedury manipulacji semaforami i obszary krytyczne. Komitet TC-1 opracował również projekt zbioru procedur dla FORTRANU ANSI X3J3/76.3 w celu włączenia typów danych bitowych i operacji do FORTRANU ANSI.

Europejski Podkomitet Komitetu Technicznego TC-2 ds Przemysłowego Basica Czasu Rzeczywistego zakończył badania porównawcze języka i sformułował projekt rozszerzenia języka Basic. W innych dodatkowych badaniach uwypuklone będą wymagania użytkownika oraz znaczenie projektu dla systemów rozproszonych i systemów mikroprocesorowych. Jako wynik końcowy, spodziewane jest opracowanie modułu ujednoczonego, minimalnego podzbioru języka Basic ANSI/ECLA, który jest na etapie zatwierdzenia.

Komitet ds Języków Proceduralnych (TC-3) zbadał wcześniejsze oświadczenie dotyczące wymagań funkcjonalnych i specyfikacji języka Tinman zaproponowanego przez amerykańskie Ministerstwo Obrony. W ocenie Komitetu projekt języka wyższego rzędu Tinman powinien być punktem szczególnego zainteresowania dla zastosowań w czasie rzeczywistym, pod warunkiem rozbudowania procedur sterujących procesami równoległymi.

Komitet ds Języków Zorientowanych Problemowo (TC-4) jest w trakcie sporządzania katalogu dostępnych języków zorientowanych problemowo (problem oriented languages - POL). Komitet bada również problem zastosowania POL w mikroprocesorach i opracowuje uogólniony assembler (generalized assembly language - GAL) dla mikroprocesorów. Przypuszcza się, że GAL znacznie uprości generowanie i przenoszenie oprogramowania z jednego systemu do drugiego.

Komitet TC-5 ds Interfejsu i Transmisji Danych pracuje nad wymaganiami funkcjonalnymi dla schematów szeregowego podziału linii transmisji danych między podsystemami. Amerykański podkomitet regionalny został mianowany Komitetem ISA SP 72 ds Interfejsów Komputerów Przemysłowych i Metod Transmisji Danych oraz prowadzi działalność jako Amerykańska Techniczna Grupa Doradcza w zakresie standardu ISO/TC97, SC13/WG1 - "Opis interfejsu między obliczeniowym systemem przetwarzania i procesem technicznym", a w zakresie IEC/SC65A/WG6 - "Transmisja wewnątrzsystemowa komputerów przemysłowych".

Komitet ds Komunikacji Człowiek-Maszyna (TC-6) kontynuował prace nad poszerzeniem i uaktualnieniem publikacji instruktażowej "Projekt interfejsów człowiek-maszyna do sterowania procesem". Opracowywana jest definicja języka niezależnego od urządzenia oraz procedury oszacowania kosztów komunikacji człowiek-maszyna. Komitet przygotowuje zbiór przykładów pozytywnych i negatywnych doświadczeń w zakresie komunikacji człowiek-maszyna.

Komitet ds Niezawodności, Bezpieczeństwa i Zabezpieczenia (TC-7) opracował zakres i program pracy obejmującej ogólne wymagania dotyczące systemu. Zidentyfikowano kilka określonych zagadnień związanych z instrukcją dotyczącą projektowania sprzętu i oprogramowania.

Komitet TC-8 ds Systemów Operacyjnych w Czasie Rzeczywistym miał 6 spotkań w Europie w 1976 r. i osiągnął znaczny postęp w opisie istotnego zagadnienia systemu operacyjnego w czasie rzeczywistym. W planach nowo utworzonych podkomitetów - amerykańskiego i japońskiego - znajduje się opracowanie specyfikacji dotyczących działania systemu operacyjnego w systemach jedno- i wieloprocesorowych oraz w systemach rozproszonych.

Komitet TC-9 ds Terminologii określił zakres tematyczny dziedziny, dla której zbierane są terminy. Zespół zamierza opracować drugie wydanie słownika terminologii przemysłowych komputerów cyfrowych ISA.

SIKOS - PAKIET PROGRAMÓW DO SYMULACJI SYSTEMÓW CIĄGLYCH

Firma SIEMENS opracowała pakiet programów SIKOS do symulacji procesów ciągłych umożliwiającym opis modelu w notacji podobnej do języka FORTRAN. System symulacyjny SIKOS składa się z translatora i symulatora. Translator wczytuje model i dane, testuje składnię, logikę i rozkłady prawdopodobieństwa oraz tworzy przy bezbłędym wprowadzeniu specjalnie strukturywany program w języku FORTRAN. Program ten jest następnie przetłumaczony i z wieloma modułami standardowymi z biblioteki modułów SIKOS podłączany do symulatora.

Dla każdego modelu generowany jest własny symulator, który zależnie od potrzeby, może otrzymać parametry w różnorodny sposób przez nieformatowane dane wejściowe. Jeśli translator wykryje błędy, w modelu symulacyjnym następuje przerwanie jeszcze przed właściwym przebiegiem symulacji.

W przebiegu symulacji oblicza się wszystkie zmienne zależne modelu matematycznego łącznie z dyskretnymi.

Generowane symulatory można zarejestrować w pamięci oraz reaktywować jako odnowione danymi aktualnymi.

Programu SIKOS można używać z komputerami serii 4004 i 7000 firmy SIEMENS.

Elektronische Rechenanlagen 1977 nr 4

KOMPUTER DLA LABORATORIUM

Firma Digital Equipment Corporation zapowiedziała ostatnio nowy zautomatyzowany system laboratoryjny służący do wyszukiwania i analizowania danych jednocześnie z 16 różnych przyrządów analitycznych wytwarzających wyjścia danych o wartościach szczytowych.

System Peak-11 zbudowany jest na systemie PDP-11/34T z dyskiem elastycznym, lub jego wersji PDP-11/34F. Obydwa wymienione systemy wykorzystują procesor PDP-11/34 z pamięcią rdzeniową lub półprzewodnikową o pojemności 32 K słów. PDP-11/34T zawiera podwójny napęd dyskowy RX05, podczas gdy PDP-11/34F wykorzystuje podwójny napęd dyskowy RX11 (dysk elastyczny). System Peak-11 może pracować jako komputer czołowo/końcowy systemu operacyjnego RT-11 i wykorzystuje wersję wielodostępnej języka Basic. Według opinii firmy DEC system RT jest systemem działającym w czasie rzeczywistym, odpowiednim do wyszukiwania i analizowania danych przez oprogramowanie systemu. Posiada funkcjonalny podział oprogramowania na części: czołową i końcową. W części czołowej następuje gromadzenie danych i przetwarzanie szczytowe. Następnie wyniki są gromadzone na dysku. Część końcowa jest wykorzystana do analizy nagromadzonych informacji i tworzenia sprawozdań. Cena systemu Peak-11 wynosi 33,905 \$.

Computer 1976 nr 11

TRANSMISJA DANYCH Z WĘGERSKIEGO KOMPUTERA TPA-1 DO KOMPUTERA JEDNOLITEGO SYSTEMU

Przy zastosowaniu zautomatyzowanych metod w diagnostyce, w laboratorium przetwarzania danych Akademii Medycznej w Erfurcie wykorzystano węgierski komputer do sterowania procesami TPA-1 wraz z kasetą CAMAC. Służy on do przetwarzania analogowo-cyfrowego i do przetwarzania danych pomiarowych.

Do większych zadań z zakresu przetwarzania danych pomiarowych np. do zautomatyzowanej analizy elektrokardiogramu trzeba zastosować uniwersalny komputer Jednolitego Systemu. W takich wypadkach do rejestrowania danych pomiarowych i ich wstępnego opracowania stosuje się komputer TPA-1, a właściwej analizie dokonuje komputer uniwersalny. Najbardziej skutecznym sposobem transmisji danych jest krótkotrwała ich rejestracja na taśmach magnetycznych.

Opracowano programy do dostosowania formatu danych. Programy te mogą być wywołane w języku

FORTTRAN-TPA-1 lub w FORTRAN-DOS/JS i zapewniają elastyczność pracy. Od roku dokonuje się transmisji danych do dalszego opracowania danych z pomiaru EKG do komputerów EC 1020 i Robotron 21.

Dane te można również przysyłać do komputerów CDC 3300, BESM i ODRA.

Rechentechnik Datenverarbeitung 1977 nr 4

TECHNIKA MANIPULACYJNA - ROBOTY PRZEMYSŁOWE

Pod takim tytułem odbyło się sympozjum na Politechnice w Karl-Marx-Stadt. Urządzenia manipulacyjne wyposażone w chwytaki i sterowanie wykonują operacje według ustalonego programu. Zastępują one człowieka na stanowiskach pracy, które są ciężkie, monotonne i niebezpieczne dla zdrowia, ze względu np. na nadmiar ciepła, pyłu i gazów. Zależnie od stopnia programowania sterowania rozróżnia się: zwykle samopodawacze jako manipulatory z programem stałym, roboty przemysłowe jako dowolnie programowane i uniwersalne automaty manipulacyjne i manipulatory synchronizowane, których ramiona i chwytaki są sterowane na wzór pracownika obsługującego daną część w procesie obróbki.

Zastosowanie manipulatorów synchronizowanych jest konieczne tam, gdzie niemożliwa jest obecność człowieka np. w pomieszczeniach o dużej ilości gazu trującego, na głębokich wodach lub w "gorących miejscach" reaktorów jądrowych. Z punktu widzenia racjonalizacji produkcji przemysłowej - efekty znacznej automatyzacji osiąga się stosując zwykły manipulator, jak np. samopodawacze, urządzenia wymiany urządzeń itp.

Roboty przemysłowe znajdują zastosowanie w operacjach załadunkowych, wsadowych, montażowych, spalniczych, natryskowych farbą i w zadaniach transportowych.

Aktualny stan badań nad robotami przemysłowymi cechuje to, że obiekty, które są przedmiotem manipulacji są identyfikowane za pomocą sensorów, podtrzymywane i przenoszone na stanowisko pracy zgodnie z zadaniem programem.

Rozwój badań w tej dziedzinie pójdzie w kierunku zwiększenia tzw. sztucznej inteligencji tych maszyn. Roboty przemysłowe będą mogły rozpoznawać nieprzewidziane sytuacje, szybko reagować na nie według przewidzianych wariantów, aby w ten sposób uniknąć kolizji i przestojów w produkcji. Badania te rodzą nowe zadania dla techniki pomiarowej, sterowania i regulacji: potrzeba nowych systemów rozpoznawania obiektów, sterowań i sensorów. Ważną rolę odegrają również w tych poszukiwaniach mikroprocesory i mikrokomputery. Uniwersalność i cena tych urządzeń będą decydowały o stopniu ich wyposażenia. Tak jak dziś roboty przemysłowe są dostosowane do obsługi maszyn istniejących, tak w przyszłości gdy maszyny i systemy maszynowe będą zaplanowane i przystosowane do współpracy z robotami, wyłoni się z pewnością technologia robotów.

W celu wymiany doświadczeń i rozwoju badań założono specjalny zespół roboczy pod nazwą "Technika manipulacyjna" przy Izbie ds Techniki w Karl Marx-Stadt.

Messer Steuern Regeln 1977 nr 231

RAMIĘ ROBOTA

Roboty powoli torują sobie drogę z laboratoriów badawczych do zastosowań przemysłowych. Firma VICAM Inc. posiada instalacje pilotowe robota w firmie General Motors, National Bureau of Standards i dużym przedsiębiorstwie półprzewodnikowym, gdzie jest stosowany do montażu elektroniki kalkulatorów. Znany jako "Standard Manipulator System" (SMS), robot ten składa się z ramienia z ręką z dwoma palcami, centralnej rejestracji danych i jednostki interfejsu sterowania z 32 kanałami konwersji analogowo-cyfrowej, mikrokomputera DEC LSI-11 z 20 K słów pamięci i urządzenia końcowego z klawiaturą. Ramię ma serwosterowane położenie, szybkość i moment skręcający we wszystkich punktach. Ma ono sześć stopni swobody - wykonuje wszystkie ruchy, jakie można wykonać barkiem, ramieniem i przegubem ręki. Jednostka ta jest zaprogramowana w języku wysokiego poziomu dla PDP-11, będącym własnością firmy PDP-11. Zdolność obciążenia manipulatora wynosi 4 kg, zaś objętość robocza jest kulą o średnicy 2 m. Cena SMS wynosi 48 tys. dol.

Datamation 1977 nr 2

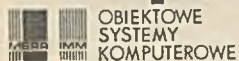
BRANŻOWY OŚRODEK INFORMACJI NAUKOWEJ TECHNICZNEJ I EKONOMICZNEJ
INSTYTUTU MASZYN MATEMATYCZNYCH
02-078 Warszawa, ul. Krzywickiego 34, tel. 21-84-41 w. 391

BOINTE udziela informacji

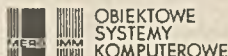
z zakresu techniki komputerowej

BOINTE wydaje

informacja ekspresowa

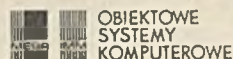


przegląd dokumentacyjny



Materiały konferencyjne, szkoleniowe, prospekty

biuletyn informacyjny



BOINTE gromadzi

wydawnictwa zwarte, czasopisma krajowe i zagraniczne, katalogi i prospekty, sprawozdania z prac naukowo-badawczych oraz inne materiały informacyjne

wykonuje usługi reprodukcyjne i poligraficzne
mikrofilmy, kserokopie z zakresu posiadanych zbiorów

P3057 / 77

WARUNKI PRENUMERATY

Prenumeratę na kraj przyjmują Oddziały RSW "Prasa-Książka-Ruch" oraz urzędy pocztowe i doręczyciele w terminie do 25 listopada na rok następny.

Cena prenumeraty rocznej zł 840.

Jednostki gospodarki uspołecznionej, instytucje i organizacje społeczno-polityczne składają zamówienia w miejscowych Oddziałach RSW "Prasa-Książka-Ruch".

Zakłady pracy i instytucje w miejscowościach, w których nie ma Oddziałów RSW oraz prenumeratorzy indywidualni, zamawiają prenumeratę w urzędach pocztowych lub u doręczycieli.

Prenumeratę ze zleceniem wysyłki za granicę, która jest o 50% droższa od prenumeraty krajowej, przyjmuje RSW "Prasa-Książka-Ruch", Centrala Kolportażu Prasy i Wydawnictw, ul. Towarowa 28, 00-958 Warszawa, konto PKO nr 1531-71 - w terminie podanym dla prenumeraty krajowej.