



Patent dodatkowy
do patentu

Zgłoszono: 12.04.74 (P. 170324)

Pierwszeństwo: 13.04.73 Wielka
Brytania

Zgłoszenie ogłoszono: 02.06.75

Opis patentowy opublikowano: 15.06.1982

Int. Cl.² G06F 9/18

CZYTELNIA

Urzędu Patentowego
Polskiej Rzeczypospolitej Ludowej

Twórca wynalazku: _____

Uprawniony z patentu: International Computers Limited, Londyn (Wielka Brytania)

System przetwarzania danych

1

Przedmiotem wynalazku jest system przetwarzania danych, wyposażony w procesor przystosowany do obsługi urządzeń peryferyjnych w oparciu o przypisane im priorytety.

Pojemność układów buforowych wymagana co najmniej dla najszybszych urządzeń peryferyjnych związanych z systemem przetwarzania danych można zmniejszyć na zasadzie przypisania poszczególnym urządzeniom peryferyjnym priorytetów i umożliwienia urządzeniom o priorytetach wyższych przerywania obsługi urządzeń o priorytetach niższych. Procedury obsługi przerwania wymagają przetwarzania odpowiednich mikroprogramów o różnym stopniu skomplikowania, a przetwarzanie to pochłania na ogół znaczną część czasu pracy procesora, co jest zauważalne zwłaszcza przy składaniu i odtwarzaniu mikrorozkazów i danych urządzenia, którego obsługa zostaje przerwana.

Znane są na przykład z opisów patentowych USA nr 3 282 239 i 3 599 162 systemy przetwarzania danych zawierające zespół przetwarzający, który odbiera żądania przerywania, na przykład z urządzeń peryferyjnych. Żądania są przypisane do wielu różnych poziomów priorytetu, tak że żądanie o wyższym poziomie priorytetu ma pierwszeństwo przed żadaniami o niższych poziomach priorytetu. Zespół przetwarzający ma obwód kontroli priorytetu, który wybiera żądanie o wyższym priorytecie i uaktywnia odpowiedni poziom lub stan programu, który obsługuje żądanie.

2

Programy są zazwyczaj przetrzymywane w pamięci, przy czym zwykle stosuje się specjalny rejestr zwany rejestrem adresów programu dla adresowania poszczególnych instrukcji programowych w pamięci. Rejestr ten jest zazwyczaj inkrementowany tak, aby nadążał za programem, to znaczy jedna instrukcja w określonym czasie.

Z opisu patentowego USA nr 3 373 408 znane jest stosowanie wielu rejestrów adresów programu, przy czym każdy odpowiada jednemu poziomowi lub stanowi programu. Tak więc żądany poziom lub stan programu może być uaktywniony w prosty sposób przez wybranie odpowiedniego, jednego z rejestrów dla adresowania pamięci. Wybrany rejestr będzie oczywiście inkrementowany w zwykły sposób, tak aby podążał za programem. Zaopatrzenie w wiele rejestrów adresów programu znacznie ułatwia przełączanie z jednego poziomu lub stanu programu na drugi, gdy występuje przerwanie.

To znane rozwiązanie posiada tę wadę, że chociaż ułatwia obsługę żądań przerywania, lecz nie wspomaga wykonania programów, które nie są związane z żadaniami przerywania. W szczególności, nie ma tu wspomaganie w wykonywaniu zgrupowanych podprogramów, to jest gdy jeden program zmienia się na drugi program lub podprogram, dla którego przewidywany jest powrót.

Celem wynalazku jest pokonać to ograniczenie przez opracowanie systemu ułatwiającego wykonywanie zgrupowanych podprogramów. System prze-

tworzania danych, wyposażony w mikroprogramowo sterowany procesor mający mikroprogramową pamięć, pamięć operacyjną, układ sterowania pamięci operacyjnej i wiele mikroprogramowych rejestrów adresowych wybieranych odpowiednio dla adresowania mikroprogramowej pamięci, wiele urządzeń peryferyjnych oraz układ priorytetu odbierający sygnały żądania przerwania z urządzeń peryferyjnych i wytwarzający sygnał selekcji wskazujący identyczność urządzenia peryferyjnego, z którego pochodzi wybrany sygnał żądania, według wynalazku charakteryzuje się tym, że multiplekser znajdujący się w układzie sterowania pamięci operacyjnej swym pierwszym wejściem połączony jest z układem priorytetu na którego wyjściu występuje sygnał selekcji.

Swym drugim wejściem multiplekser połączony jest z wyjściem włącznika rewersyjnego. Ponadto wyjście multipleksera jest połączone z mikroprogramowym rejestrem adresowym dla adresowania mikroprogramowej pamięci, przy czym na wyjściu multipleksera występuje sygnał z pierwszego wejścia przy obecności sygnału żądania, a przy braku sygnału żądania na wyjściu tego multipleksera występuje sygnał z drugiego jego wejścia.

Multiplekser jest połączony z mikroprogramowym rejestrem adresowym poprzez rejestr identyfikacji poziomu, którego wyjście jest połączone z komparatorem połączonym jednocześnie z pierwszym wejściem multipleksera dla porównania zawartości rejestru identyfikacji poziomu z wartością sygnału selekcji z układu priorytetu i wytworzenia sygnału przerwania jeśli zawartości rejestru są większe niż wartość sygnału selekcji.

Do mikroprogramowego rejestru adresowego dołączony jest poprzez drugi multiplekser tor wieloliniowy dołączony do wszystkich urządzeń peryferyjnych. Ten drugi multiplekser jest sterowany sygnałem przerwania aby mógł spowodować przekazywanie startowych adresów mikroprogramowych z urządzeń peryferyjnych wytwarzających przerwanie poprzez tor wieloliniowy do aktualnie wybranego mikroprogramowego rejestru adresowego.

Przedmiot wynalazku jest przedstawiony w przykładzie wykonania na rysunku, na którym fig. 1 przedstawia ogólny schemat blokowy systemu przetwarzania danych, fig. 2 — schemat blokowy części systemu przedstawionego na fig. 1, która odpowiada zmianom poziomu adresowego pamięci mikroprogramowej wynikającymi z żądania wysyłanego przez urządzenie peryferyjne, fig. 3 — schemat blokowy części systemu z fig. 1, która odpowiada za zmiany poziomu mikroprogramowego, a fig. 4 przedstawia schemat blokowy części systemu z fig. 1 związanej z głównymi operacjami pamięciowymi.

Na figurze 1 przedstawiono układ pamięci 10 obejmujący pamięć mikroprogramową 11 i pamięć operacyjną lub pamięć danych 12. Obie pamięci 11 i 12 przedstawiono jako adresowane niezależnie. Pamięci te są przedstawione w postaci bloków o różnych rozmiarach. Ma to na celu pokazanie, że mają one słowa o różnych długościach, a również różne pojemności informacyjne. Zaleca się,

aby pamięci te były budowane z podobnych modułów półprzewodnikowych w celu otrzymania pojedynczej jednostki o oddzielnych układach adresowych. Układ sterowania 13 pamięci mikroprogramowej jest przedstawiony łącznie z torem adresowym 14 pamięci i torem rozkazowym 15, który jest torem dwukierunkowym umożliwiającym zapisywanie materiału mikroprogramowego, jak również odczytywanie go do celów związanych ze sterowaniem.

W podobny sposób przedstawiono układ sterowania 16 pamięci operacyjnej wyposażony w tor adresowy 17 i dwukierunkową szynę danych 18 połączoną z pamięcią operacyjną 12.

Dekoder mikrorozkazów 19 dekoduje mikrorozkazy pobrane z pamięci 11 przez układ sterujący 13 i przesyłane linią 20.

Dekoder 19 generuje sygnały sterujące przeznaczone dla głównych układów logicznych i jednostki arytmetycznej lub układu obróbki danych 21. Dekoder 19 przetwarza również sygnały sterujące skokami mikroprogramowymi dla układu sterowania 13 pamięci mikroprogramowej 11.

Przedstawiono dwukierunkowe połączenia 22 i 23 pomiędzy układem obróbki danych 21, a dekodерem mikrorozkazów 19 i układem sterującym 16 pamięci operacyjnej. Podobne dwukierunkowe połączenia 24, 25, 26, 27 znajdują się pomiędzy układem obróbki danych 21, a odpowiednio: pamięcią roboczą 28, rejestrami roboczymi 29, pamięcią obrazową 30 i peryferyjnym systemem sterującym 31. Pamięć robocza 28 jest bardzo szybką pamięcią notatnikową o organizacji słowowej, wykorzystywaną w zwykły sposób przez układ obróbki danych 21.

Zaleca się, aby wszystkie adresy pamięci roboczej były dostępne dla poziomów 8—15 aktywności procesora, natomiast aby tylko część adresów — a mianowicie połowa — była dostępna dla poziomów mikroprogramowych związanych z urządzeniami peryferyjnymi. W obu półówkach tej pamięci zarezerwowano komórki przewidziane dla znaczników. Pozostałe komórki są przypisane rejestrom zakresowym, informacjom związanym z tłumaczeniem (translation slaves) stałym, słowom sterującym urządzeń peryferyjnych i innym rejestrom związanym z żadaniami dostępu mikroprogramu. Pamięć robocza 28 jest bardzo szybką pamięcią notatnikową o pojemności 256 słów 32-bitowych, która umożliwia wykonanie dwu odczytów i jednego zapisu w jednym mikrorozkazie.

Rejestry robocze 29 zostaną opisane bardziej szczegółowo w połączeniu z układami mikroprogramowania wielopoziomowego. Pamięć obrazowa 30 składa się z rejestrów adresowalnych, które informują procesor o zawartości wszystkich rejestrów pozostałych, przy czym mikroprogram przechowywany w pamięci mikroprogramowej 11 ma dostęp jedynie do niektórych rejestrów pamięci obrazowej.

Sterujący system peryferyjny 31 obsługuje od strony procesora końcówki szyny 32 urządzeń peryferyjnych i jest wyposażony w sieć priorytetów, określającą które spośród urządzeń peryferyjnych i w jakim czasie może uzyskać dostęp do procesora.

Każde urządzenie peryferyjne jest podłączone do systemu interfejsowego szyny peryferyjnej 32 za pośrednictwem złącza, które przepuszcza sygnały sterujące urządzeń peryferyjnych generowane przez mikroprogram, a również wprowadza lub wyprowadza informację. Przewidziano specjalne złącza, w które są wyposażone: pamięć dyskowa 33, czytnik kart 34, drukarka 35, monitor ekranowy 36, interfejs standardowy 37 i układ bezpośredniego wprowadzania danych 38. W przypadku wyposażenia systemu w układ bezpośredniego wprowadzania danych należy rozszerzyć pamięć mikroprogramową, co zaznaczono na fig. 1 linią przerywaną tworzącą dodatkowy blok 39.

Linią przerywaną 40 zaznaczono na fig. 2 interfejs dołączony do złącza urządzeń peryferyjnych 33—37 z fig. 1, a linia ta odpowiada w ogólności połączeniu 32 z fig. 1. Pierwszy tor wieloliniowy 41 wyprowadzony z interfejsu 40 zostaje pobudzony, gdy jakieś urządzenie peryferyjne zażąda dostępu do procesora. Dekoder priorytetów na podstawie pobudzenia toru 41 decyduje, które z urządzeń peryferyjnych ma pierwszeństwo w stosunku do innych urządzeń, które w tym samym czasie również żądają dostępu.

W ogólności, najwygodniej jest w taki sposób zorganizować tor wieloliniowy 41, aby obejmował on po jednej linii wyprowadzonej z łącza każdego urządzenia peryferyjnego.

W przykładzie zastosowano jednak bardziej zwarte kodowanie, aby uniknąć ograniczenia dotyczącego liczby obsługiwanych łączy urządzeń peryferyjnych.

Sygnał na wyjściu 43 sieci kodowania priorytetu 42, który identyfikuje określone łącze peryferyjne, reprezentuje w każdej chwili żądanie o największym priorytecie. Sygnał pojawiający się na wyjściu 43 jest przesyłany linią 46 do układu 47, który wysyła sygnał obecności linią 48 do urządzenia peryferyjnego o danym priorytecie. Pod wpływem sygnału obecności na linii 48 wybrane łącze pobudza wieloliniowy tor 49 adresem pamięci mikroprogramowej, który odpowiada początkowi określonego programu wymaganego do obsługi danego żądania. Może to dotyczyć transmisji informacji i/lub sterowania określonej fazy roboczej urządzenia peryferyjnego.

Tor wieloliniowy 49 jest wspólny dla wszystkich łączy peryferyjnych systemu. Drugi wieloliniowy tor 50, wspólny dla wszystkich łączy, jest również ładowany w tym samym czasie informacją przeznaczoną dla rejestru modyfikującego procesora.

Sygnał ustawiony na wyjściu 43 sieci kodowania priorytetu 42 jest również podawany do drugiej sieci kodowania priorytetu 51, która na wyjściu 52 dostarcza sygnał reprezentujący poziom mikroprogramowy odpowiadający wybranemu urządzeniu peryferyjnemu. Przedstawiono dwie sieci kodujące 42 i 51 ze względu na możliwość wykorzystywania tego samego poziomu mikroprogramowego przez dwa lub więcej urządzenia peryferyjne i związana z tym konieczność identyfikacji wybranego łącza (na linii 48) i odpowiadającego mu poziomu mikroprogramowego (na linii 52). Fizycznie, sieci kodujące 42 i 51 mogą być wykonane w postaci pojedyn-

czej jednostki wyposażonej w wyjścia wyprowadzone z jej różnych stopni.

Wynik dotyczący decyzji o wyborze priorytetu i określenia poziomu mikroprogramowego żądań urządzeń peryferyjnych jest przesyłany z wyjścia 52 drugiej sieci kodowania priorytetu 51 do wyjścia multiplexera 53. Multiplexer 53 wybiera jedno spośród swych dwóch wejść, na które jest podawany sygnał z wyjścia drugiej sieci kodowania 51 i sygnał z wyjścia licznika rewersyjnego 55 wykorzystywanego do określania poziomów mikroprogramowych operacji procesora nie związanych z żądaniami urządzeń peryferyjnych. Multiplexer 53 dostarcza na wyjściu 56 sygnału przesyłanego do rejestru identyfikacji poziomu 57, który to rejestr ma wyjście wieloliniowe 58.

Poziomy mikroprogramowe związane z operacjami urządzeń peryferyjnych mają priorytet taki, że multiplexer będzie faworyzował swoje wejście 52 za wyjątkiem przypadku, gdy nie występuje żądanie związane z operacjami urządzeń peryferyjnych.

Również żądanie wysyłane przez urządzenia peryferyjne mają określone priorytety i poziom mikroprogramowy może przerwać poziom o priorytecie niższym. Operacjom peryferyjnym przyporządkowano siedem poziomów, są one reprezentowane liczbami binarnymi od 1 do 7, przy czym wartość „1” reprezentuje priorytet najwyższy.

Przerwania poziomów wywoływane żądaniami wysyłanymi przez urządzenia peryferyjne są obsługiwane za pomocą komparatora 59, który porównuje sygnały ustawiony na wyjściu sieci kodującej 51, przesyłany odgałęzieniem 60, z sygnałem na wyjściu wieloliniowym 58 rejestru identyfikacji poziomu 57. Ten ostatni sygnał jest przesyłany odgałęzieniami 61 i 62.

Gdy sygnał na wyjściu drugiej sieci kodowania priorytetu 51 ma mniejszą wartość binarną (wyższy priorytet) od zawartości rejestru identyfikacji poziomu, komparator 59 wytwarza ustalony uprzednio sygnał wyjściowy na linii 63. Sygnał ten jest przesyłany linią 63 do układu przerwań 64, który sprawdza, czy przerwanie nie zakłóci określonej sekwencji sterowania, np. cyklu pamięciowego. Układ przerwań 64 ma również za zadanie ograniczanie częstotliwości występowania przerwań. Układ przerwań 64 dopuszcza przerwanie peryferyjne tylko wtedy, gdy od ostatniego przerwania minęły co najmniej trzy cykle procesora. Układ ten blokuje również przerwanie na okres pięciu cykli od chwili rozpoczęcia operacji dostępu do pamięci. Wszystkie te funkcje spełnia układ przerwań 64 za pomocą małego licznika lub rejestru przesuwanego.

W tym cyklu procesora, w którym rejestr identyfikacji poziomu mikroprogramowego 57 zmienia stan, następuje przełączenie od rejestru adresu bieżącego mikroprogramu do odpowiedniego rejestru nowego. Tego rodzaju rejestry adresowe mikroprogramu zajmują różne komórki pamięci zapisywalnej, którą stanowi mikroprogramowy rejestr adresowy 65 o pojemności 16 słów, adresowany odgałęzieniem 61 wyjścia rejestru identyfikacji poziomu 57. Zmiana poziomu przerwania powoduje również, że drugi multiplexer 66 wybiera tor wie-

lolinowy 49, którym jest podawany jeden z sygnałów wejściowych tego multiplexera i na podstawie tego sygnałuysterowuje swe wyjście 67. Sygnał z wyjścia 67 drugiego multiplexera 66 dochodzi do układu adresowania pamięci 68, który adresuje 5 układ pamięci 10 za pośrednictwem toru 69, który odpowiada strzałce 14 na fig. 1.

Inne wejście 70 drugiego multiplexera 66 zawiera wyjście licznika 71 i jest wybierane podczas 10 zwykłego adresowania sekwencyjnego pamięci operacyjnej lub pamięci danych 12 wchodzącej w skład układu pamięci 10. Układ adresowania pamięci 68 jest dołączony do wyjścia 67 drugiego multiplexera 66, ponieważ układ ten jest na ogół przeznaczony do powiększania rozmiarów pamięci, przy czym 15 nie odbywa się to kosztem szybkości, ponieważ przy adresowaniu zastosowano technikę zakładkowania adresów parzystych i nieparzystych układu pamięci 10.

Zmiana zawartości rejestru identyfikacji poziomu mikroprogramowego 57 powoduje wybranie 20 innej komórki w mikroprogramowym rejestrze adresowym 65. Ma to na celu zabezpieczenie powrotu adresu pamięci mikroprogramowej do uprzednio wybranej komórki w rejestrze adresowym 65. Nowy adres mikroprogramowy jest otrzymywany z toru wieloliniowego 49 i pojawia się w torze 69 25 tak, że operacja pobierania rozpoczyna się w pamięci mikroprogramowej 11 od pierwszego mikro-rozkazu sekwencji żądanej przez urządzenie peryferyjne o wyższym priorytecie, które wysłało żądanie. Tor 69 ma odgałęzienie poprowadzone do rejestru adresowego 65 i nowy adres pamięci mikroprogramowej jest więc zapisywany do rejestru adresu mikroprogramu wybranego przez odgałęzienie 30 61 wyjścia rejestru poziomu.

Ze względu na technikę zakładkowania stosowaną przy pobieraniu mikroprogramu, dekodowaniu i wykonaniu, dwa poprzednie mikro-rozkazy nadal są obecne w procesorze, przy czym jeden z nich jest w stopniu dekodującym układu obróbki danych 21. Jeżeli którykolwiek z tych rozkazów wymaga wykonania skoku lub dostępu do pamięci operacyjnej, wtedy konieczny będzie powrót do poziomu przerwanej w celu osiągnięcia tego adresu 45 pamięci mikroprogramowej, który wymagał skoku, to jest w celu wyzerowania rozkazu odpowiadającego mu i rozkazu następnego. Zerowanie to wykonuje układ logiczny 72, który wytwarza na wyjściu 73 sygnał o wartości „2” w każdym cyklu procesora, poza przypadkiem wystąpienia przerwania, 50 kiedy to wartość tego sygnału jest zmniejszana do „1”, a następnie do „0” dla następnych cykli procesora, chyba że wystąpiły operacje skoku lub dostępu do pamięci, które zablokują sygnałem na linii 75 zmiany sygnału na wyjściu 73. Sygnał z wyjścia układu logicznego 72 jest przesyłany na wejście następnej zapisywalnej pamięci zerowej 74 o dostępie swobodnym i pojemności 16 słów, w 55 której adres zapisywanej komórki jest również określane sygnałem w odgałęzieniu 61 wyjścia układu poziomu, który to sygnał ustawia odpowiednią wartość w rewersyjnym liczniku adresowym 77.

Odpowiednia komórka pamięci zerowej 74 będzie 65

więc przechowywała liczbę reprezentującą ilość 8 mikro-rozkazów, które wymagają wyzerowania po powrocie z przerwania. Sygnał ustawiony na wyjściu 76 pamięci zerowej 74 jest podawany na wejściu 78, która stanowi 8 część układu obróbki danych 21 i odejmuje stan zerowy (null count) ustawiony w torze 76 od wartości ustawionej na wyjściu 79 rejestru adresowego 65. Odpowiedni zmodyfikowany adres pamięciowy jest dostępny na linii 80, która stanowi 10 jednocześnie wejście drugiego multiplexera 66. Wejście to jest wybierane przez multiplexer 66 podczas powrotu do przerwanej poziomu mikroprogramowego i ma ono tę samą wartość co zawartość 15 odpowiedniej komórki rejestru adresowego 65, poza przypadkiem, gdy wystąpi konieczność wykonania operacji zerowania.

Wskazanie poziomu mikroprogramowego o wyższym priorytecie wymaga zmiany stanu rejestru 20 identyfikacji poziomu 57, zmiany odpowiednich komórek pamięci rejestrowych 65 i 74 i zmiany adresu podawanego torem 69 do pamięci. Spowoduje to pobranie mikro-rozkazu określonego na torze 49. W następnym cyklu procesora jest wykonywane 25 dekodowanie w dekoderze mikro-rozkazów 19, i wykonanie ostatniego mikro-rozkazu poziomu przerwanej w układzie obróbki danych 21.

W ogólności należy spodziewać się, że technika 30 taśmowego wykonywania rozkazów pociąga za sobą dalsze wykonanie funkcji zakładkowania, a dotyczy to zwłaszcza tworzenia adresu pamięci mikroprogramowej.

W drugim cyklu procesora następującym po 35 zmianie zawartości rejestru identyfikacji poziomu mikroprogramowego 57 wymagane jest wykonanie pierwszego mikro-rozkazu z mikro-rozkazów przerywających. Pociąga to za sobą przełączenie pomiędzy rejestrami roboczymi.

Sygnał ustawiony na wyjściu wieloliniowym 58 40 rejestru identyfikacji poziomu 57 jest wprowadzany w odpowiednim czasie do rejestru 81. Sygnał z wyjścia 82 jest pobierany z rejestru 81 w drugim cyklu po zmianie poziomu mikroprogramowego. Jest on wykorzystywany do zmiany adresów przesyłanych do zapisywanych pamięci o dostępie 45 bezpośrednim 83, 84, 85 i 86, które obejmują odpowiednie rejestry robocze A, B, K i C. Wielorejestrowe pamięci K i C mają pojemność szesnastu słów, z których jedno jest wybierane w czasie odpowiadającym sygnalizacji poziomu mikroprogramowego na linii 82. 50

Jeden rejestr wybrany z rejestrów K i C przyjmuje bieżącą zawartość odpowiedniego licznika 55 rewersyjnego 87 lub 88 przesyłaną odpowiednio torem 89 lub 90. Rejestry K85 są wykorzystywane do sterowania pętlami mikroprogramowymi na odpowiednich poziomach mikroprogramowych, rejestry te są zerowane, gdy określony poziom jest wykorzystywany po raz pierwszy. Rejestry C86 są 60 ładowane podczas wprowadzania odpowiednich poziomów mikroprogramowych, zawartość tych rejestrów stanowią modyfikatory adresów pamięci roboczej 28, przy czym każdy poziom mikroprogramowy, a zwłaszcza każde łącze peryferyjne ma 65 zapewniony dostęp do określonego obszaru pamięci

robotycznej, które to obszary mogą być i zwykle są różne dla różnych łączy. Rejestry **K** są zerowane przy każdym ładowaniu rejestrów **C**, to jest przy pierwszym wybieraniu łącza. Podczas przerywania wysyłanego przez łącze peryferyjne odpowiedni rejestr **C** jest wybierany po upływie dwu cykli procesora od chwili zainicjowania przerywania, rejestr ten jest ładowany wartością ustawioną na linii 50 wyprowadzonej z interfejsu podsystemu łącza.

Jest to wykonywane za pośrednictwem toru 91 na zasadzie wybrania odpowiedniego wejścia multiplexera 92, przy jednoczesnym wybraniu zerowego wejścia 93 multiplexera 94 tak, abyysterować licznik 87 za pośrednictwem wyjścia 95 multiplexera.

Zastosowanie rejestrów **C** jako modyfikatorów adresowych pamięci robotycznej umożliwia wykonanie tego samego fragmentu mikro kodu na różnych poziomach mikroprogramowych, jak również na tym samym poziomie, gdy fragment ten zostanie po prostu wprowadzony ponownie. Operację tę można rozpocząć na jednym poziomie i przed jej zakończeniem dla tego poziomu można zakończyć przetwarzanie jednego lub kilku przerwań poziomów o wyższych priorytetach przed powrotem do zakończenia wykonywania poziomu pierwotnego. Możliwość ta jest użyteczna przy sterowaniu standardowych łączy interfejsowych.

Podczas ładowania wybranego rejestru **C** na wejściowym poziomie mikroprogramowym łącza peryferyjnego ładowany modyfikator jest podawany odgałęzieniem 96 wyjścia 90, multiplexera do układu dekodera 97 w celu przesłania sygnału sterowania wybieraniem—linią 98 i interfejsem łącza 32 — na pierwotne łącze peryferyjne, które pod wpływem tego sygnału wprowadzi swe dane na wejście wspólnej szyny danych prowadzącej do procesora i zarejestruje sygnały sterujące procesora.

Wyjścia wybranych rejestrów **C** i **K** są dostępne na liniach 99 i 100 odpowiednio dla jednostki arytmetycznej procesora w celu zaadresowania odpowiedniej pamięci robotycznej i w celu sterowania wykonywaniem pętli mikroprogramowych. Również wejścia 101 i 102 multiplexerów 92 i 94 stanowią wyjścia jednostki arytmetycznej, dzięki czemu można dokonywać zmian zawartości rejestrów podczas przetwarzania. Przedstawiono również tory sprzężenia zwrotnego 103 i 104 wyprowadzone z wyjść 99 i 100 rejestrów na wejścia multiplexerów 92 i 94. Wymienione tory sprzężenia zwrotnego są wykorzystywane podczas operacji wybierania wymaganej podczas ładowania liczników 88 i 87.

Rejestry **A** i **B** 83 i 84 są wykorzystywane jako źródła argumentów dla jednostki arytmetycznej i dla innych zastosowań jak: ładowanie pamięci robotycznej i dostarczanie adresów pamięci operacyjnej, dostarczanie danych, które mają być zapisane do pamięci i ładowanie rejestrów **C** i **K**. Część każdego rejestru **A** i **B** może być wykorzystywana jako modyfikator adresu pamięci robotycznej.

Argumenty są dostarczane wyjściami 105 i 106 wybranych rejestrów **A** i **B** podłączonych jako wybieralne wejścia do obu multiplexerów 107 i 108. Wyjścia 109 i 110 multiplexerów 107 i 108

wchodzą do rejestrów argumentów 111 i 112 przeznaczonych dla jednostki arytmetycznej. W praktyce wymienione multiplexery mają kilka innych wejść alternatywnych, a ich wyjścia 109 i 110 są rozgałęzione, zwielokrotnione i wykorzystywane do innych celów. W skład wymienionych wejść innych wchodzi wejście binarnego zera, wejście binarnej jedynki, wejście danych pochodzących z wybranego łącza peryferyjnego i wejście danych pochodzących z pamięci operacyjnej, wejść tych nie pokazano w celu zachowania przejrzystości schematu.

Każdy z rejestrów **A** i **B** jest wykonany jedynie w dziewięciu egzemplarzach i tak: po jednym rejestrze przeznaczono na każdy poziom mikroprogramowy (1—7) związany z operacjami wykonywanymi przez urządzenia peryferyjne, jeden rejestr jest zarezerwowany na uszkodzenia (poziom „0”) i jeden jest wykorzystywany przez wszystkie poziomy mikroprogramowe (8 do 15) związane z operacjami procesora. Jest tak dlatego, ponieważ poziomy mikroprogramowe związane jedynie z operacjami procesora są wykorzystywane do budowy stosów tak, że w przypadku, gdy wymagają one oddzielnych informacji sterujących dla każdego poziomu, to wystarczy jedynie dostarczyć jednego zbioru źródeł danych, to jest argumentów lub słów dla i z pamięci.

Tor wybierania poziomu mikroprogramowego 82 prowadzi również do następnej zapisywalnej pamięci 113 o dostępie swobodnym, która ma wejścia i wyjścia połączone z jednostką arytmetyczną i przeznaczone do przesyłania informacji związanych ze stanami znaczników porównawczych i wskaźników występujących w jednostce arytmetycznej, które są wymagane do sterowania i przetwarzania zgodnie z podstawową listą rozkazów procesora. Pamięć 113 ma szesnaście rejestrów, po jednym dla każdego poziomu mikroprogramowego.

Opisano powyżej w jaki sposób łącze peryferyjne odpowiadające poziomowi mikroprogramowemu o wyższym priorytecie może przerwać dowolny poziom mikroprogramowy o niższym priorytecie, ponadto przedstawiono również w jaki sposób, podczas powrotu do poziomu mikroprogramowego, operacja modyfikacji zerującej powoduje, że skoki lub dostępy do pamięci nie powodują żadnych strat informacji, nawet wtedy gdy występują one podczas wykonywania tego poziomu, który jest przerywany. Może jednak wystąpić przypadek, że podczas takiego przerywania pojawi się żądanie obsługi wygenerowane przez inne łącze peryferyjne, które ma poziom mikroprogramowy o niższym priorytecie niż łącze przerywające, ale o wyższym priorytecie od poziomu przerywanego.

W takim przypadku łącze żądające obsługi powinno zostać obsłużone przed wykonaniem powrotu do poziomu przerywanego.

Jednakże o ile zwykły powrót do poziomu przerywanego spowoduje po prostu ponowne wybranie odpowiedniego mikroprogramu i rejestrów robotycznych, to obsłużenie nowego żądania o priorytecie pośrednim będzie wymagało zebrania informacji dotyczących adresu początkowego pamięci mikroprogramowej i modyfikatora pamięci robotycznej,

ustawionych na liniach 49 i 50. Wejście do określonego poziomu mikroprogramowego wiąże się z ustawieniem odpowiednich bitów znaczkowych w dodatkowym rejestrze o długości szesnastu bitów, po jednym dla każdego poziomu mikroprogramowego. Osiągnięto to wykorzystując dodatkowe pozycje bitowe w każdej komórce pamięci zerowej 74, która jest ładowana za pośrednictwem linii ML przy każdej zmianie poziomu.

Badanie zawartości tej pamięci odbywa się na linii M podczas powrotu z przerwania. Po zakończeniu przerwania żądanie o najwyższym priorytecie poziomu mikroprogramowego, jak określono to w peryferyjnym systemie sterowania 31, wytwarza wskaźnik definiujący odpowiedni poziom mikroprogramowy. Badany jest bit znacznika odpowiadający temu poziomowi. Jeżeli bit ten jest zapalony, wykorzystywane są odpowiednie wartości adresu mikroprogramowego, rejestrów C i K itd. Jeżeli natomiast wartością tego bitu jest zero, co oznacza, że wejście do tego poziomu jest wykonywane po raz pierwszy, wtedy sygnały są zbierane z linii 49 i 50 w taki sposób, jak dla przerwania.

Następną cechą układu przerwania jest to, że żądanie dostępu do procesora wysyłane przez łącza peryferyjne są zliczane w tych fazach przez impulsy zegarowe procesora, wymienione impulsy zegarowe mogą być blokowane przez procesor w chwilach, gdy przerwania nie zostanie przyjęte przez procesor.

Dzięki takiemu rozwiązaniu zadania peryferyjne są zsynchronizowane z taktowaniem procesora. Żądania pochodzące z łącza pozostają zapalone do chwili wysłania przez procesor impulsu EXIT łącznie z sygnałem sterowania wybieraniem. Dzięki takiemu rozwiązaniu zapewnione jest, że poziom przerwania nie zostanie skasowany przed powrotem z przerwania lub przerwania.

Procesor według wynalazku obsługuje żądanie wysyłane przez łącza peryferyjne na jednym poziomie mikroprogramowym wybranym spośród poziomów od 1 do 7. Procesor ten może oczywiście rozpocząć wykonywanie swoich operacji wykorzystując odpowiedni poziom mikroprogramowy. Operacje wykonywane na poziomach 1—7 nie mogą być przerywane przez poziom stojące wyżej w hierarchii poziomów. W przypadku pojawienia się takiego przerwania operacja będzie wykonywana na następnym wyższym poziomie wymagającym obsługi, co może oznaczać dalsze opóźnienie przed powrotem do pierwszego poziomu przerwane. Operacje procesora na poziomie mikroprogramowym „0” — jest to poziom o najwyższym priorytecie — są zarezerwowane dla przerwania krytycznych, takich jak błędy parzystości lub uszkodzenia układów. W przypadku gdy nie występuje żądanie wykonania operacji peryferyjnych, zwykle przetwarzanie danych odbywa się na poziomach mikroprogramowych 8—15.

Operacje wykonywane na poziomach mikroprogramowych 8—15 różnią się od operacji na poziomach 0—7 tym, że przejścia z poziomu na poziom odbywają się sekwencyjnie zgodnie ze sterowaniem stosowanym do programów zagłębionych. W przypadku wystąpienia skoku do programu następnego,

z którego jest oczekiwany powrót, mikrorozkaz LINK, po zdekodowaniu, spowoduje przesłanie impulsu zliczającego do licznika rewersyjnego 55 w celu dostarczenia następnego poziomu o wyższym priorytecie, który wybierze odpowiedni adres mikroprogramowy i rejestry zliczające w celu załadowania adresu początkowego określonego rozkazem LINK.

Po powrocie ze skoku mikrorozkaz EXIT, po zdekodowaniu, spowoduje przesłanie rewersyjnego impulsu zliczającego do licznika rewersyjnego 55, który wybiera rejestr adresowy mikroprogramowego poziomu o niższym priorytecie w celu kontynuowania przetwarzania na tym poziomie. Przy każdym przejściu jest wykorzystywana technika taśmowa, dzięki czemu odpowiednie zmiany rejestrów roboczych C i K i porównania znaczników i rejestrów wskaźników 113 będą występować po upływie dwu cykli procesora od rozpoczęcia wymienionego przejścia.

System ten umożliwia obsłużenie maksymalnej liczby ośmiu zagłębionych skoków, przy czym nie ma tu żadnych wymagań dotyczących czasochłonnego składowania i rozładowywania. Rozkazy skokowe mogą być symulowane na zasadzie przepisywania do rejestru adresowego mikroprogramu poziomu bieżącego adresu obliczonego. Pociąga to za sobą deklarowanie rejestru adresowego mikroprogramu jako argumentu przeznaczenia, a sam rozkaz będzie traktowany jako skok bezwarunkowy bez zerowania sekwencji wykonywanej, która powoduje straty cykli procesora.

Przyczyną, dla której operacje dostępu do pamięci operacyjnej wymagają zerowania jeżeli wystąpią one podczas przetwarzania w technice taśmowej na poziomach od 0 do 7 jest to, że rejestry wejścia, wyjścia pamięci operacyjnej nie są powielone dla różnych poziomów mikroprogramowych, a zaleta techniki taśmowej polega na organizacji sterowania pamięci.

Na figurze 4 przedstawiono podstawowy układ pamięci operacyjnej 12. Pokazano tu rejestr 114 przeznaczony dla słów zapisywanych do pamięci, rejestr 115 — dla słów odczytywanych z pamięci, tor adresowy 17 wyprowadzony z wyjścia drugiego multiplexera 66 w przypadku wybrania jego wejścia 70 przychodzącego z licznika adresowego 71 pamięci operacyjnej. Aby zabezpieczyć zawartość liczników 114, 115 i licznika adresowego 71 do chwili ich wykorzystania przez bieżący poziom mikroprogramowy zapala się wskaźnik zajętości pamięci, który blokuje zmiany poziomów mikroprogramowych do chwili zgaszenia go. Zaleca się, aby gaszenie tego wskaźnika nie następowało później niż po wykonaniu określonej liczby mikrorozkazów następujących po załadowaniu licznika adresowego 71 pamięci tak, aby nie dopuścić do nieuzasadnionego blokowania zmiany poziomów, co mogłoby doprowadzić do kryzysu czasowego w urządzeniu peryferyjnym, przy czym chodzi tu o synchroniczne, niebuforowe urządzenia takie jak dyski i czytniki kart.

Tę uprzednio określoną liczbę można ustalić automatycznie za pomocą sprzętu lub za pomocą stosowania odpowiedniej konwencji pisania mikropro-

gramów, w przykładzie wykonania urządzenia według wynalazku wynosi ona pięć.

Podczas operacji odczytu pamięci operacyjnej licznik adresowy 71 jest ładowany jednym mikro-rozkazem, który powoduje zapalenie warunku zajętości pamięci, np. jest wyzwalany przerzutnik 116 przedstawiony w pobliżu licznika 71.

Dane odczytane spod tego adresu zostaną określone jako znaczące w pamięciowym rejestrze wyjściowym 115 w trzecim mikro-rozkazie po załadowaniu licznika adresowego 71, wskaźnik zajętości pamięci jest gaszony po wykorzystaniu pamięciowego rejestru wyjściowego 115 jako źródła argumentu. Dwie operacje odczytu mogą zachodzić na siebie w sensie czasowym zgodnie z tzw. techniką zakładkowania, jeżeli zapoczątkowano drugą operację odczytu na zasadzie załadowania licznika 71 po rozkazie, który bezpośrednio poprzedza rozkaz wykorzystujący rejestr 115 jako źródło argumentu. Następnie, drugie słowo wczytane do rejestru 115 musi również zostać wykorzystane jako argument nie później niż w pięć rozkazów po pierwszym załadowaniu licznika 71.

Podczas operacji zapisu do pamięci jest ładowany pamięciowy licznik adresowy 71 i wyzwalany przerzutnik 116, po czym następuje załadowanie pamięciowego rejestru wejściowego 114. Przerzutnik 116 przestanie być wyzwalany po pobraniu zawartości pamięciowego rejestru wejściowego 114. Operacje zapisu nie mogą być zakładkowane, ale operacja zapisu może zachodzić w czasie na operację odczytu poprzez załadowanie licznika 71 rozkazem, który poprzedza rozkaz wykorzystujący rejestr wyjściowy 115 jako argument. Następnie jest ładowany rejestr wejściowy 114, przy czym następuje to nie później jak w pięć rozkazów po załadowaniu licznika 71 dla operacji odczytu.

Stosowana jest również operacja pamięciowa odczytaj — zmodyfikuj — zapisz, która jest wykonywana w czasie pięciu mikro-rozkazów odpowiadających zajętości pamięci.

Łącznie np. 33—38 są fizycznymi układami, które dzięki odpowiedniemu zaprojektowaniu uwzględniającemu wszelkie potrzebne specjalizowane sekwencje mikro-rozkazowe, maskują wszelkie różnice, które w przeciwnym przypadku byłyby widoczne dla kodu operacyjnego, z którego pamiętany mikroplan tłumaczy programy składające się z rozkazów wchodzących w skład listy rozkazów podstawowych, na sygnały sterujące wewnętrznymi operacjami procesora. Dzięki takiemu rozwiązaniu osiągnięto dużą elastyczność w stosunku do nowych urządzeń peryferyjnych lub mechanizmów, ponieważ po wyposażeniu ich w odpowiednie łącze nie zachodzi konieczność zmiany układów procesora.

Podstawowa lista mikro-rozkazów obejmuje wiele rozkazów konwencjonalnych. Na przykład dwuadresowe rozkazy 32-bitowe mogą dotyczyć operacji arytmetycznych dodawania, odejmowania i wprowadzania oraz operacji logicznych jak przesuwanie, mnożenie logiczne, sumowanie logiczne, równoważność, natomiast rozkazy trójadresowe mogą dotyczyć podobnych operacji logicznych, a ponadto operacji i-nie, równoważności, nie-i łącznie z operacją porównaj i skocz i operacjami wyzwalania

znaczników i rejestrów. Za pomocą rozkazów szesnastobitowych można wykonywać operacje ładowania rejestrów, operacje arytmetyczne i logiczne łącznie z operacjami przesuwania i wyzwalania znaczników, które to operacje mogą być traktowane jako rozkazy główne. Pomocnicze rozkazy 16-bitowe są wykorzystywane przy ładowaniu i operacjach arytmetycznych i wspomagają one główne rozkazy 16-bitowe, ponadto mogą być one wykorzystywane do wykonywania różnych operacji skokowych wykonywanych w zależności od stanów znaczników, wskaźników lub wyników operacji porównywania.

Obecnie zostaną omówione pewne szczególnie użyteczne mikro-rozkazy dodatkowe. Jeden z nich jest związany z zalecaną organizacją bajtową procesora, który pracuje na słowach 32-bitowych i 4-bajtowych, 8 bitów na bajt. Ten rodzaj rozkazu pobiera dowolny bajt argumentu i zastępuje nim dowolny bajt argumentu innego. Rozkaz ten może być wykorzystywany do szeregowego przesyłania określonych bajtów, a również do budowania słów, w których jeden bajt pochodzi z jednego źródła, a pozostałe trzy bajty pochodzą ze źródła innego. W ogólności oba źródła i miejsca przeznaczenia mogą być zamienione w zależności od danego rozkazu. Jeżeli np. argumenty są umieszczone w rejestrach A i B, 83, 84, poszczególne bajty mogą być pobierane selektywnie z dowolnego argumentu na zasadzie odpowiedniego wybrania wejść multiplexera 107 lub 108. Operacje sterowane wykorzystujące odpowiednie modyfikatory umożliwiają użyteczną i elastyczną obróbkę bajtów.

Jeżeli znaki są traktowane jako bajty, to omówiony wyżej rozkaz może być uważany za efektywne narzędzie obróbki znaków.

Fakt, że bajtowa organizacja procesora może być uważana za narzędzie obróbki znaków oznacza, że wynikowe kody rozkazowe mające organizację słowową, a nie bajtową mogą być przystosowane do organizacji bajtowej ze wszystkim wypływającymi z tego zaletami.

Dane dotyczące wynikowych kodów rozkazowych zorientowanych słowowo mogą być przyjmowane przez procesor zorientowany bajtowo jeżeli długość słowa jest taka sama w obu przypadkach. Co więcej, kod wynikowy dotyczący słowa o długości mniejszej od długości słowa procesora mogą również być przystosowywane za pomocą dwóch innych rodzajów mikro-rozkazów, z których jeden rozdziela format słowowy na części mniejsze od długości bajtu procesora i wypełnia brakujące obszary bajtów zerami. Rozkaz ten nazywa się EXPAND i działa na rejestrze przesuwalnym w celu przeprowadzenia odpowiednich manipulacji na bajtach i dodawanych zerach. Rozkaz drugi jest wykorzystywany po przetworzeniu w celu usunięcia zer i połączenia znaczących bitów słowa, rozkaz ten ma nazwę COMPRESS i również działa na rejestrze przesuwalnym.

W przypadku 4-bajtowych, 32-bitowych słów procesora i 24-bitowych słów kodu wynikowego odpowiedni rozkaz EXPAND wprowadza słowo 24-bitowe na pozycje 0—23 rejestru przesuwalnego i wykonuje następujące przesunięcia w prawo:

z pozycji dla bitów 0—5, 4 pozycje dla bitów 6—11, 6 pozycje dla bitów 12—17 i 8 pozycje dla bitów 18—23. Na pozycje bitowe 0 i 1, 8 i 9, 16 i 17, 24 i 25 są wprowadzane zera. W taki sposób zostaje utworzone słowo czterobajtowe, przy czym na dwóch bardziej znaczących pozycjach bitowych w każdym bajcie są ustawione zera.

Innym rodzajem rozkazu jest rozkaz, który wykonuje dowolne określone operacje logiczne pomiędzy argumentem istniejącym, określonym na podstawie adresu występującego w rozkazie pamięci obrazowej w zmodyfikowanym lub nie rozkazie dla pamięci roboczej, a argumentem zbudowanym specjalnie, który ma albo literał „1” albo stan określonego znacznika wprowadzonego na wyznaczoną pozycję bitową zerowego słowa 32-bitowego. Wynik może być wykorzystany do przepisania argumentu istniejącego, lub też znacznik wyznaczony może być zastąpiony wynikiem dla wyznaczonej pozycji bitowej. Różne pola mikrorozkazu określają argument istniejący, rodzaj operacji logicznej, wyznaczoną pozycję bitową, określony znacznik (gdy taki występuje) i zastosowanie wyniku.

Ten rodzaj rozkazu nadaje się zwłaszcza do badania stanu operacyjnego urządzenia peryferyjnego. Rozkaz ten powoduje przeładowanie kilku pozycji słowa statusowego (słowo stanu), co jest wykonywane na zasadzie kolejnego wybierania różnych wyznaczonych pozycji bitowych w celu wybrania odpowiednich operacji logicznych, przy czym wyniki pośrednie są przechowywane na pozycji znacznikowej w akumulacyjnej funkcji boolowskiej.

Lista rozkazów, obok rozkazów dotyczących skoków i porównań pomiędzy bajtami i słowami, obejmuje również mikrorozkaz przeznaczony do sterowania skokiem w zależności od stanu wyznaczonego bitu określonego argumentu. Rozkaz ten nadaje się zwłaszcza do testowania wyniku badania statusu urządzenia peryferyjnego przeprowadzonego przez poprzedni mikrorozkaz. Tłumaczenie określonego kodu wynikowego na podstawowy zbiór funkcji mikroprogramowych procesora wykonuje inny rodzaj mikrorozkazu, który jest wykorzystywany do czasowego usuwania dowolnego pola z określonego słowa.

Operacja ta jest wykonywana na zasadzie przesuwania słowa o określoną liczbę pozycji bitowych, a następnie wybrania określonej liczby bitów tego słowa, na ogół od strony mniej znaczącej słowa. Wybrana część lub pole słowa może być następnie przetworzona w celu dostosowania jej do adresu wymaganej funkcji mikroprogramowej. Pociąga to za sobą operację indeksowania od adresu bazowego tablicy funkcji w pamięci operacyjnej. Tablica ta, obok adresu początkowego mikroprogramu może zawierać inne informacje dotyczące adresowanych programów.

Jako przykład przedstawia się operacje wykonywane w maszynach cyfrowych ICL 1900. Wybrana operacja dotyczy ładowania rejestru (LDX) wymaganego przy dowolnej realizacji kodu wynikowego i składa się z kilku etapów podstawowych: przy czym występujące tu liczby mają postać szesnastkową.

W pierwszym etapie adres rozkazu ICL 1900 jest

ładowany do rejestru adresowego pamięci i pobierany jest odpowiadający mu rozkaz (kod wynikowy). Ma to postać rozszerzoną obejmującą zera uzupełniające cztery bajty 8-bitowe.

W drugim etapie rozkaz przechowywany w pamięciowym rejestrze wyjściowym jest przetwarzany na format 1900 o długości 24 bitów na zasadzie wykorzystania rejestru przesuwanego i operacji usuwania zer. Wynik jest wprowadzany do rejestru **B**. Następnie jest wykonywana operacja czasowego usuwania dotycząca zawartości rejestru **B**, co ma na celu wyizolowanie części operacyjnej rozkazu, która zostaje przesunięta na pozycje mniej znaczące słowa.

Tak więc odbywa się przesuwanie 14 pozycji bitowych i usuwanie 9 mniej znaczących pozycji bitowych. Wynik jest wykorzystywany do indeksowania adresu bazowego tablicy funkcji w celu wykonania operacji dostępu do pamięci roboczej. Następnie jest wykonywana następną operacją czasowego usuwania dotycząca zawartości rejestru **B** (przesunięcie o jeden bit i usunięcie 5 bitów mniej znaczących). Operacja ta dostarcza adresu określonego rejestru i adres ten jest wprowadzany do znanego adresu w pamięci roboczej. W następnym etapie odbywa się trzecie czasowe usunięcie pola rejestru **B** (przesunięcie w prawo o 8 pozycji i pobranie **C** bitów), co zapewnia informacje o tym, co ma być zapisane do pożądanego rejestru i wprowadza ją do innego znanego adresu w pamięci roboczej. W kolejnym etapie następuje załadowanie określonego rejestru wymaganą zawartością, co może być wykonane po przeprowadzeniu różnych testów, np. po zbadaniu czy informacja, która ma być załadowana jest literałem, czy adresem rejestru, a jeżeli jest adresem rejestru, to czy określa ona rejestr, do którego wprowadzono adres rejestru usuniętego, lub czy oba te adresy określają ten sam rejestr. Inne czynniki, które mogą być wykorzystane przed wykonaniem ostatniego etapu dotyczą określenia modyfikatora, np. z rejestru **C** lub **B**.

Przy początkowej operacji dostępu do pamięci występuje opóźnienie wynoszące dwa cykle procesorowe pomiędzy pierwszym a drugim etapem, opóźnienie to jest zwykle wykorzystywane do zbadania ważności określonego adresu. Stanowi to po prostu rozszerzenie testu rozpoczętego przed wykonaniem pierwszego dostępu do pamięci.

Na figurze 2 i 3 przedstawiono wyjścia różnych rejestrów procesora, wyjścia te są zaopatrzone w odgałężenia oznaczone IS. Oznacza to, że zawartości tych rejestrów są zapamiętywane w pamięci obrazowej przedstawionej na fig. 1. W ten sposób dzięki zastosowaniu odpowiedniego bramkowania tych odgałżeń otrzymuje się ciągły zapis stanu aktywności procesora, co jest szczególnie użyteczne do celów diagnostyki.

Zastrzeżenia patentowe

1. System przetwarzania danych wyposażony w mikroprogramowo sterowany procesor mający mikroprogramową pamięć, pamięć operacyjną, układ sterowania pamięci operacyjnej i wiele mikropro-

