INSTYTUT ŁĄCZNOŚCI
WARSZAWA-MIEDZESZYN

BIULETYN INFORMACYJNY

7 (173)
1978

MINISTERSTWO ŁĄCZNOŚCI

BIULETYN INFORMACYJNY

ROK 18 WARSZAWA 1978 NR 7/173/

INSTYTUT ŁĄCZNOŚCI
Branżowy Ośrodek
Informacji Naukowo-Technicznej i Ekonomicznej

Redakcja Biuletynu Informacyjnego

Redaktor Naczelny - prof. mgr inż. Lesław Kędzierski
Z-ca Redaktora Naczelnego - doc. dr inż. Krystyn Plewko

Redaktorzy działów:
doc. mgr inż. Władysław Cetner, doc. mgr inż. Adam Moniuszko

Adres Redakcji:
Instytut Łączności
Branżowy Ośrodek
Informacji Naukowo-Technicznej i Ekonomicznej
Warszawa-Miedzeszyn, ul. Szachowa 1

NA PRAWACH RĘKOPISU - DO UŻYTKU SŁUŻBOWEGO

Redaktor: J. Borkowska

Montaż tekstu: B. Drabik

Dział Wydawniczy Instytutu Łączności
Format B5. Nakład 620. Wpłynęło do
Działu Wydawniczego 24.05.1978 r.
Druk ukończono w lipcu 1978 r.

Maria Tyrowicz

ARCHITEKTURA SYSTEMÓW MIKROPROCESOROWYCH

SPIS TREŚCI

	Str.
1. Wstęp	1
2. Architektura systemów mikroprocesorowych	2
2.1. Wprowadzenie	2
2.2. Bloki systemu mikroprocesorowego	3
2.2.1. Pamięci	3
2.2.2. Współpraca elementów pamięciowych z mikroprocesorem	5
2.2.3. Mikroprocesor	6
2.2.4. Bloki wejścia/wyjścia	10
2.2.5. Inne bloki systemu mikroprocesorowego	12
3. Podsumowanie	12
Wykaz literatury	12

Maria Tyrowicz

PROGRAMOWANIE SYSTEMÓW MIKROPROCESOROWYCH

1. Wstęp	19
2. Etapy programowania	20
3. Języki programowania i translatory	21
3.1. Języki wewnętrzne mikroprocesorów	21
3.2. Języki symboliczne	24
4. Metody adresowania pamięci	26
5. Programowanie operacji wejścia/wyjścia	28
6. Podsumowanie	30
Wykaz literatury	30

Maria Tyrowicz

ARCHITEKTURA SYSTEMÓW MIKROPROCESOROWYCH

1. WSTĘP

Elektroniczne systemy cyfrowe w ciągu ostatnich kilku lat przechodziły kolejne ewolucje. Było to wynikiem rozwoju technologii wytwarzania elementów półprzewodnikowych - opracowania układów scalonych początkowo o małej /SSI/, a następnie średniej /MSI/ oraz wielkiej /LSI/ skali integracji.

Obecnie najciekawszymi funkcjonalnie elementami o wielkiej skali integracji są mikroprocesory. Umożliwiają one w połączeniu z elementami pamięciowymi LSI oraz elementami pomocniczymi realizację systemu cyfrowego o możliwościach funkcjonalnych "klasycznego" komputera , złożonego z kilku do kilkunastu układów scalonych.

Pierwsze mikroprocesory /1971 r./ zapoczątkowały w elektronice przewrót porównywalny z przewrotem, który w swoim czasie wywołały tranzystory. Mikroprocesory dzięki swoim możliwościom obliczeniowym i sterującym, małym gabarytom, niskiemu poborowi mocy oraz dużej niezawodności zastąpiły układy złożone z wielu elementów małej i średniej skali integracji /układy bramek i przerzutników/oraz umożliwiły zautomatyzowanie wielu urządzeń technicznych.

Jednym z głównych kierunków zastosowań mikroprocesorów jest sprzęt telekomunikacyjny. Mikroprocesory znalazły zastosowanie między innymi przy transmisji danych w sieciach komputerowych /głównie jako bloki procesorów komunikacyjnych/, w telefonicznych centralach elektronicznych /między innymi jako bloki procesora głównego/, przy cyfrowym kodowaniu mowy, przy sterowaniu potokami pojazdów oraz w telekomunikacyjnej aparaturze pomiarowej.

Poza sprzętem telekomunikacyjnym, mikroprocesory są obecnie szeroko stosowane w układach sterowania liniami produkcyjnymi, w zautomatyzowanych stanowiskach pomiarowych, w sprzęcie komputerowym, w elektronice samochodowej oraz w sprzęcie powszechnego użytku /programowane kuchenki i pralki, gry telewizyjne/.

Jakkolwiek mikroprocesor jest pod względem elektronicznym układem bardzo rozbudowanym, to dla pełnienia swoich funkcji wymaga dodatkowo bloków pomocniczych /pamięci i bloków wejścia/wyjścia/ oraz tak zwanych urządzeń zewnętrznych - wejściowych i wyjściowych. Wynika to z zasady jego działania. Mikroprocesor pobiera z pamięci instrukcje, a następnie dekoduje je i wykonuje. Instrukcje zawarte w pamięci tworzą tak zwany program. Bloki wejścia/wyjścia służą do komunikacji mikroprocesora ze światem zewnętrznym /wprowadzanie danych, wyprowadzanie wyników operacji obliczeniowych/.

Projektowanie systemu sterowanego mikroprocesorem - tak zwanego systemu mikroprocesorowego polega na zaprojektowaniu układu połączeń pomiędzy elementami wchodzącymi w skład systemu oraz na zaprojektowaniu odpowiedniego programu. Tak więc na koszt opracowania nowego systemu mikroprocesorowego składa się koszt samego układu oraz koszt opracowania programu. Przy ciągle malejących cenach układów scalonych LSI koszt opracowania programu odgrywa rolę coraz bardziej dominującą /rys. 1/[x] [7]

Celem serii opracowań, rozpoczętych niniejszym artykułem jest omówienie podstawowych zagadnień związanych z zasadami pracy i z projektowaniem systemów mikroprocesorowych. Artykuł niniejszy pt. "Architektura systemów mikroprocesorowych" omawia zasady budowy mikroprocesora, pamięci i bloków wejścia/wyjścia, ich rolę w systemie mikroprocesorowym oraz sposoby wzajemnego łączenia. W drugim artykule pt. "Programowanie systemów mikroprocesorowych" przedstawiono kolejne etapy budowy programu, ze szczególnym uwzględnieniem zagadnień dotyczących języka wewnętrznego mikroprocesora. Trzeci artykuł pt. "Przegląd mikroprocesorów oraz przegląd systemów wspomagających projektowanie" zawiera omówienie podstawowych "rodzin" mikroprocesorów wraz z tablicowym zestawieniem parametrów poszczególnych typów mikroprocesorów oraz przedstawia programy i urządzenia pomocne przy projektowaniu i testowaniu systemów mikroprocesorowych. Wreszcie w ostatnim artykule pt. "Przegląd wybranych zastosowań systemów mikroprocesorowych" omówiono telekomunikacyjny obszar zastosowań systemów mikroprocesorowych. Artykuły zawierają zestawienia bibliograficzne na poszczególne tematy.

2. ARCHITEKTURA SYSTEMÓW MIKROPROCESOROWYCH

2.1. Wprowadzenie

Pojęcie "architektura systemu" stosowane jest do określania struktury funkcjonalnej bloków systemu cyfrowego. W odniesieniu do systemów mikroprocesorowych pojęcie "architektura" odnosi się do funkcjonalnej struktury wewnętrznej mikroprocesora oraz do sposobu łączenia mikroprocesora z pozostałymi blokami systemu mikroprocesorowego.

Prosty system mikroprocesorowy składa się z pamięci, mikroprocesora, bloku wejścia/wyjścia oraz urządzeń zewnętrznych. Urządzeniem zewnętrznym jest na przykład czytnik taśmy papierowej, perforator, drukarka, monitor ekranowy itd.

Pamięć systemu mikroprocesorowego składa się zwykle z kilku do kilkunastu pamięciowych układów scalonych LSI. W pamięci przechowuje się w postaci binarnej informacje o kolejnych operacjach, które ma wykonywać mikroprocesor oraz magazynuje się dane oraz wyniki operacji. Informacja określająca pojedynczą operację

---

[x] Rysunki są zamieszczone na końcu artykułu.

nosi nazwę instrukcji. Programem nazywa się uporządkowany zespół instrukcji, który opisuje konkretne zadanie systemu mikroprocesorowego.

Mikroprocesor, który składa się z jednego lub kilku układów scalonych LSI, ma zdolność przetwarzania danych pobranych z pamięci lub z urządzeń zewnętrznych oraz nadzorowania procesu przesyłania danych oraz wyników operacji w obrębie systemu mikroprocesorowego. Mikroprocesor wykonuje kolejno wszystkie operacje zgodnie z instrukcjami programu, a więc pobiera instrukcję z pamięci, dekoduje ją, wykonuje operację określoną pobraną instrukcją, pobiera kolejną instrukcję z pamięci, itd. Jednocześnie mikroprocesor generuje zespół sygnałów sterujących pracą całego systemu.

Blok wejścia/wyjścia pośredniczy w wymianie informacji pomiędzy urządzeniami zewnętrznymi i resztą systemu mikroprocesorowego [1, 2, 11],

W systemie mikroprocesorowym informacje są przesyłane i przetwarzane w sposób równoległy. Liczba bitów jednocześnie przesyłanych lub przetwarzanych nosi nazwę długości słowa mikroprocesora. W celu realizacji jednoczesnego przesyłania zespołu bitów pomiędzy blokami systemu mikroprocesorowego bloki te połączone są ze sobą wiązkami przewodów. Wiązki przewodów, które łączą ze sobą poszczególne bloki noszą nazwę szyn. Ze względu na rodzaje informacji przesyłane poszczególnymi szynami wyróżnia się szynę danych, szynę adresową i szynę sterującą. Szyną danych przesyła się dane z lub do mikroprocesora, do lub z pamięci czy bloku wejścia/wyjścia. Szyną adresową przesyła się z mikroprocesora do pamięci informacje o fragmencie pamięci, z którego należy pobrać instrukcję, lub w którym należy umieścić daną czy wynik operacji. Szyną sterującą przesyła się sygnały,które nadzorują pracę poszczególnych bloków systemu. Poszczególne szyny obsługują zwykle kilka bloków systemu. Aby uniknąć wzajemnego obciążania się bloków dołączonych do tych samych szyn,bloki posiadają tak zwane trójstanowe wyjścia na szyny. W czasie pracy bloku jego wyjścia na szyny mogą przybierać stan logicznego zera lub logicznej jedynki /zwykle poziomy napięciowe TTL/. Jeśli blok nie pracuje - jego wyjścia na szyny ustawiają się w stan wysokiej impedancji wyjściowej.

Systemy mikroprocesorowe pracują synchronicznie. Do synchronizacji pracy służy sygnał zegarowy. Generator sygnału zegarowego może być oddzielnym elementem systemu lub może znajdować się wewnątrz mikroprocesorowego układu scalonego.

Na rysunku 2 przedstawiono blokowo sposób wzajemnego połączenia mikroprocesora, pamięci, bloku wejścia/wyjścia oraz urządzeń zewnętrznych przy wykorzystaniu szyny danych, szyny sterującej i szyny adresowej.

2.2. Bloki systemu mikroprocesorowego

2.2.1. Pamięci

Pamięć jest zespołem logicznych elementów dwustanowych. Podstawową jednostką

pamięci jest komórka. W jednej komórce pamięci można w danym momencie przechowywać jeden bit informacji. Liczba bitów jednocześnie wpisywanych do czy odczytywanych z pamięci nosi nazwę długości słowa pamięci. Obszar pamięci konieczny do zapisania jednego słowa tworzy tak zwaną lokację pamięci. Każdej lokacji przypisany jest jej numer kolejny, zwany adresem. Adresy przyporządkowane są poszczególnym lokacjom pamięci w sposób jednoznaczny. Liczba lokacji /słów/ zawartych w pamięci określa pojemność pamięci.

Pamięć dołączona jest do szyny danych, szyny adresowej i szyny sterującej systemu mikroprocesorowego /rys. 2/. Z szyny danych poprzez swoje wejścia pamięć pobiera informację, która ma być do niej wpisana, a na szynę danych poprzez swoje wyjścia pamięć wysyła informację, która ma być przesłana do innego bloku systemu. Szyna adresowa, dołączona do wejść adresowych pamięci, zawiera adres lokacji pamięci, do której należy wpisać informację z szyny danych,lub z której należy wyprowadzić informację na szynę danych.Z szyny sterującej doprowadza się do wejść sterujących pamięci sygnał zapisu lub sygnał odczytu oraz sygnał zegarowy. Sygnał zapisu/odczytu określa czy w danej chwili pamięć ma pracować jako źródło czy jako miejsce docelowe informacji. Sygnał zegarowy synchronizuje moment pobierania /wysyłania/ informacji z /na/ szyny danych z wysyłaniem /pobieraniem/ informacji przez inny blok systemu mikroprocesorowego.

Informacje zawarte w pamięci można podzielić na informacje stałe i informacje chwilowe. Informacje stałe stanowią instrukcje programu, które nie ulegają zmianom w czasie pracy systemu. Informacje chwilowe - dane oraz wyniki operacji ulegają zmianom w czasie pracy systemu /wprowadza się sukcesywnie kolejne dane, otrzymuje się kolejne, pośrednie i końcowe wyniki operacji/. Do przechowywania informacji stałych stosuje się półprzewodnikowe pamięci stałe LSI - współczesne odpowiedniki dawniej stosowanych pamięci diodowych /matryc diodowych/. Podstawową zaletą pamięci stałych jest przechowywanie informacji po wyłączeniu napięć zasilających. Do przechowywania informacji chwilowych służą półprzewodnikowe lub magnetyczne pamięci typu zapis/odczyt - współczesne odpowiedniki pamięci zbudowanej z pojedynczych przerzutników [4,6].

Wprowadzanie informacji do pamięci stałych nosi nazwę programowania pamięci. Ze względu na sposoby programowania pamięci stałych można wyróżnić kilka rodzajów pamięci tego typu - to znaczy pamięci typu ROM, pamięci typu PROM oraz pamięci typu REPROM. Pamięci ROM /Read Only Memory/ programowane są przez producenta w oparciu o zamówienie późniejszego użytkownika. Do użytkownika trafia już zaprogramowana pamięć. Jej zawartości nie może on zmienić. Pamięci PROM /Programmable Read Only Memory/ są programowane przez użytkownika. Ponieważ w tym przypadku proces programowania polega na przepalaniu złączy półprzewodnikowych w wybranych komórkach pamięci, pamięć typu PROM nadaje się do jednokrotnego programowania. Pamięci REPROM /Reprogrammable Programmable Read Only Memory/

są również programowane przez użytkownika, jednakże proces programowania pamięci typu REPROM nie narusza struktury półprzewodnikowej pamięci. Polega on na wprowadzaniu ładunków elektrycznych do wybranych komórek pamięci. Dzięki temu poprzez odprowadzenie ładunków zawartych w tych komórkach istnieje możliwość wymazania zawartości pamięci. Ze względu na metody stosowane przy wymazywaniu zawartości pamięci typu REPROM rozróżnia się pamięci typu EPROM /Erasable Programmable Read Only Memory/ i pamięci typu EAROM /Electrically Alterable Read Only Memory/. Wymazywanie zawartości pamięci typu EPROM odbywa się poprzez naświetlanie struktury półprzewodnikowej światłem nadfioletowym, a wymazywanie zawartości pamięci typu EAROM poprzez doprowadzanie do poszczególnych komórek odpowiednich sygnałów elektrycznych. Na rysunku 3 przedstawiono niektóre obecnie produkowane pamięci REPROM.

Własności elementów pamięciowych określa się na ogół przez podanie następujących parametrów:

1/ długości słowa pamięci /długości słowa obecnie produkowanych pamięci wahają się od 1 do 8 bitów/,

2/ pojemności pamięci /pojemności obecnie produkowanych elementów pamięciowych wahają się od 256 do 10^6 słów/,

3/ czasu dostępu - czasu potrzebnego na pobranie lub umieszczenie informacji w wybranej lokacji pamięci /czasy dostępu obecnie produkowanych pamięci wahają się od 10 ns do 10 ms/,

4/ materiału zastosowanego do wykonania struktury pamięciowej /do wytwarzania struktur pamięciowych stosuje się obecnie materiały półprzewodnikowe i materiały magnetyczne/,

5/ technologii wytwarzania struktury pamięciowej /pamięci półprzewodnikowe wykonuje się w technologiach stosowanych do wytwarzania elementów scalonych LSI, na przykład T^2L, ECL, MOS, I^2L, natomiast pamięci magnetyczne wytwarzane są głównie w technologii pęcherzykowej [1,3,4,6].

2.2.2. Współpraca elementów pamięciowych z mikroprocesorem

Pamięć dołącza się do szyny danych, szyny adresowej i szyny sterującej systemu mikroprocesorowego. Liczba przewodów szyny adresowej określa tak zwaną przestrzeń adresową pamięci. Przestrzeń adresowa odpowiada liczbie kombinacji liczb dwójkowych, którą można przesłać szyną adresową. Ponieważ każdy przewód szyny zdolny jest przenosić jeden bit informacji, przy N przewodach szyny adresowej przestrzeń adresowa wynosi 2^N. Przestrzeń adresowa określa, ile lokacji pamięci można zaadresować przy wykorzystaniu danej szyny adresowej.

Na ogół przestrzeń adresowa jest większa niż łączna liczba słów programu i da-

nych. Jednakże często pojemność określonego, pojedynczego pamięciowego układu scalonego nie wystarcza, aby umieścić w nim cały program lub wszystkie dane. W związku z tym występuje potrzeba zestawiania pamięciowych układów scalonych ROM /pamięć programu/ i układów scalonych RAM /pamięć danych/ w zespół elementów pamięciowych. Aby zaadresować lokację pamięci w tak utworzonym bloku układów scalonych, należy wybrać układ, w obrębie którego znajduje się żądana lokacja oraz określić jej położenie w obrębie tego układu. Przewody szyny adresowej dzieli się więc na dwie grupy: jedną grupą przewodów przesyła się kod dwójkowy odpowiadający "numerowi" żądanego układu scalonego w bloku pamięciowym, drugą grupą przewodów przesyła się kod dwójkowy odpowiadający "numerowi" żądanej lokacji w obrębie układu scalonego.

Na rysunku 4 przedstawiono sposób przyłączenia bloku układów pamięciowych do szyny danych i szyny adresowej. Symbolami od A_0 do A_9 oznaczono kolejne przewody szyny adresowej, które służą do adresowania lokacji pamięci w obrębie poszczególnych pamięciowych układów scalonych. Symbolami od A_{10} do A_{15} oznaczono kolejne przewody szyny adresowej, które służą do adresowania układów scalonych w obrębie bloku pamięciowego. Symbole od D_0 do D_7 odpowiadają kolejnym przewodom szyny danych. Przewody, którymi przesyłane są sygnały zapisu i odczytu oznaczono symbolami W i R, zaś przewód, którym przesyłany jest sygnał zegarowy oznaczono symbolem Ø.

2.2.3. Mikroprocesor

Obecnie produkuje się wiele typów mikroprocesorów, które różnią się między sobą szeregiem parametrów. Jednym z podstawowych parametrów jest długość słowa mikroprocesora. Spotyka się mikroprocesory o stałej długości słowa równej 4, 8, 12 lub 16 bitom, a także mikroprocesory, które można łączyć ze sobą równolegle, uzyskując żądaną długość słowa, tak zwane mikroprocesory modułowe. Mikroprocesory o stałej długości słowa charakteryzują się stałą, określoną przez producenta listą instrukcji. Mikroprocesory modułowe są zwykle mikroprogramowalne.

Spośród mikroprocesorów dostępnych na rynku większość stanowią układy o stałej liczbie instrukcji i stałej długości słowa. Im poświęcone jest poniższe omówienie. Informacje na temat mikroprocesorów modułowych oraz mikroprogramowania są dostępne w literaturze [1, 3, 8, 12, 14, 15].

Mikroprocesory, podobnie jak inne elementy półprzewodnikowe, przechodziły kolejne etapy ewolucji technologicznej. Pierwsze mikroprocesory budowane były z kilku układów scalonych. Wynikało to z początkowych trudności w uzyskaniu skomplikowanej struktury mikroprocesora na jednej płytce półprzewodnika. Poszczególne funkcje mikroprocesora realizowały oddzielne układy scalone. Postęp w technologii LSI umożliwił jednak wykonanie pełnego mikroprocesora w pojedynczym ukła-

dzie scalonym, a w ubiegłym roku pojawiły się układy scalone, które zawierają wewnątrz nie tylko pełne mikroprocesory, lecz także pamięci ROM, PROM lub REPROM,. pamięci RAM, generatory zegarowe oraz bloki wejścia/wyjścia. Układy te są samodzielnymi systemami mikroprocesorowymi. Ich pamięci wewnętrzne są wystarczająco pojemne, aby pomieścić przeciętny program pracy systemu. W razie potrzeby istnieje jednak możliwość dołączania do nich dodatkowych pamięci zewnętrznych. Należy zauważyć, że kolejne modyfikacje mikroprocesorów nie wpływają na zmianę zasad ich działania, a jedynie na powiększenie ich możliwości funkcjonalnych.

Na rysunku 5 przedstawiono schemat blokowy typowego współczesnego mikroprocesora o stałej długości słowa i stałej liczbie instrukcji. Poniżej omówiono funkcje jego poszczególnych wyodrębnionych bloków.

1. L i c z n i k r o z k a z ó w - licznik programowany o pojemności równej objętości przestrzeni adresowej

Licznik rozkazów zawiera adres lokacji pamięci, z której należy pobrać do wykonania następną, po aktualnie wykonywanej, instrukcję. Po pobraniu instrukcji z pamięci zawartość licznika rozkazów automatycznie zwiększa się o liczbę równą liczbie słów zajmowanych w pamięci przez pobraną instrukcję. Wskutek powyższej modyfikacji licznik rozkazów znów zawiera adres lokacji następnej instrukcji. Istnieje również możliwość programowego załadowania licznika rozkazów zadaną wartością. Dzięki takiej operacji można zmienić, w stosunku do ułożenia w pamięci, kolejność wykonywania instrukcji.

2. A k u m u l a t o r - rejestr o długości równej długości słowa mikroprocesora.

Akumulator zawiera jeden z argumentów przed wykonaniem operacji arytmetycznej lub logicznej. Drugim argumentem, przy operacjach dwuargumentowych, jest dana pobrana z pamięci, z bloku wejścia/wyjścia lub dana natychmiastowa. Do akumulatora odsyłany jest wynik operacji. Mikroprocesory mogą zawierać jeden lub więcej akumulatorów. Jeśli mikroprocesor zawiera kilka akumulatorów, wybór jednego z nich jako źródła argumentu operacji oraz miejsca docelowego wyniku operacji odbywa się poprzez wykonanie odpowiedniej instrukcji programowej. Argumenty operacji noszą nazwę operandów.

3. R e j e s t r w s k a ź n i k ó w - rejestr sprzężony z akumulatorem.

Rejestr wskaźników przechowuje, w postaci stanów logicznych grupy przerzutników, pewne informacje o wyniku ostatnio wykonanej operacji arytmetycznej lub logicznej. Poszczególne przerzutniki rejestru wskaźników noszą nazwę wskaźników mikroprocesora. Stan binarny poszczególnych wskaźników może określać na przykład zerową wartość liczby znajdującej się w akumulatorze czy znak tej liczby, prze-

pełnienie akumulatora itp. Niektóre wskaźniki można programowo zerować, ustawiać w stan logicznej jedynki lub negować, a wszystkie testować. Jeśli mikroprocesor ma kilka akumulatorów, to każdy z nich ma swój własny rejestr wskaźników.

4. J e d n o s t k a a r y t m e t y c z n o - l o g i c z n a /w skrócie ALU/.

W jednostce arytmetyczno-logicznej wykonują się operacje arytmetyczne i logiczne na jednym lub dwóch operandach. Jednym z operandów jest zawartość akumulatora. Do akumulatora odsyłany jest z ALU wynik operacji. Dokładny wykaz operacji, które mogą być realizowane w ALU, zawiera lista instrukcji arytmetyczno-logicznych mikroprocesora.

5. R e j e s t r I n s t r u k c j i - rejestr o długości równej długości słowa mikroprocesora.

Rejestr instrukcji przechowuje kod operacyjny instrukcji pobranej z pamięci do wykonania. Kod operacyjny jest to część instrukcji, która określa rozkaz zawarty w instrukcji. Zawartość rejestru instrukcji przekazywana jest do dekodera instrukcji.

6. D e k o d e r i n s t r u k c j i

Dekoder instrukcji rozszyfrowuje rozkaz zawarty w instrukcji i przekazuje informacje do zespołu sterowania.

7. Z e s p ó ł s t e r o w a n i a - blok odpowiedzialny za nadzór nad całością pracy systemu.

Zespół sterowania na podstawie informacji z dekodera instrukcji generuje zespół sygnałów sterujących pracą poszczególnych bloków systemu oraz rejestrów i liczników mikroprocesora. Za przykład można podać sygnał zapisu/odczytu pamięci, czy też sterowanie licznikiem rozkazów czy jednostką arytmetyczno-logiczną mikroprocesora.

8. R e j e s t r y s t o s u - grupa rejestrów równoległych, z których każdy ma długość równą długości słowa mikroprocesora.

Rejestry stosu - zwane także pamięcią stosową lub po prostu stosem - połączone są ze sobą tak, że informacje mogą być przekazywane równolegle z rejestru do rejestru. W czasie pracy z rejestrów stosu wyprowadza się jako pierwsze to słowo, które było do nich wprowadzone jako ostatnie. Drugim w kolejności wyprowadzanym słowem jest to, które było wprowadzone jako przedostatnie itd. Rejestry stosu służą do przechowywania zawartości licznika rozkazów i zawartości rejestrów mikroprocesora przy zmianie aktualnie wykonywanego programu na inny program /przesunięcia sterowania do innego miejsca pamięci/ bez straty informacji, w jakim momencie wykonywania programu zaszła ta zmiana. Jeśli w nowo wykonywanym programie wystąpi instrukcja powrotu do uprzednio wykonywanego programu, ze stosu

pobiera się informacje dotyczące uprzedniej zawartości licznika programu oraz zawartości rejestrów mikroprocesora.

Pamięć stosowa może znajdować się w obrębie mikroprocesora - nosi wtedy nazwę stosu hardware'owego lub może być utworzona w pewnym obszarze pamięci RAM - nosi wtedy nazwę stosu software'owego. Aby można było wykorzystać fragment pamięci RAM jako stos, mikroprocesor musi zawierać tak zwany wskaźnik stosu. Wskaźnik stosu jest programowanym licznikiem rewersyjnym o pojemności równej objętości przestrzeni adresowej. Jego zawartość określa adres lokacji pamięci, która stanowi aktualnie dostępną pozycję stosu. Zamiast przesuwania słów w obrębie rejestrów stosu /stos hardware'owy/ zmienia się zawartość wskaźnika stosu. Zmiana zawartości wskaźnika stosu odbywa się automatycznie przy wykonywaniu instrukcji typu "pobierz ze stosu" czy "odeślij na stos".

9. R e j e s t r y i n d e k s o w e - rejestry pomocnicze

Rejestry indeksowe umożliwiają stosowanie indeksowej metody adresowania pamięci. Indeksowa metoda adresowania pamięci jest szczególnie użyteczna przy przetwarzaniu danych umieszczonych w kolejnych lokacjach pamięci.

10. U k ł a d s t e r o w a n i a s y s t e m e m p r z e r w a ń

Układ sterowania systemem przerwań przyjmuje od urządzeń zewnętrznych sygnały żądania obsługi przerwania. Po przyjęciu żądania obsługi przerwania mikroprocesor przerywa wykonywanie programu głównego i przechodzi do wykonywania programu obsługi przerwania /np. program przyjęcia danej z urządzenia zewnętrznego i umieszczenia jej w pamięci/. Po wykonaniu programu obsługi przerwania mikroprocesor powraca do wykonywania programu głównego. System przerwań może być programowo uaktywniany /mikroprocesor akceptuje sygnały żądania przerwania/ lub dezaktywowany /mikroprocesor ignoruje sygnały żądania obsługi przerwania i nie przerywa programu głównego/ [1, 2, 3, 11].

11. R e j e s t r y o g ó l n e g o p r z e z n a c z e n i a - grupa rejestrów pomocniczych

Rejestry ogólnego przeznaczenia umożliwiają chwilowe przechowywanie wyników operacji bez konieczności odsyłania ich do pamięci.

Przedstawione wyżej bloki funkcjonalne występują na ogół we wszystkich mikroprocesorach ogólnego przeznaczenia. Struktury poszczególnych bloków rzutują na własności użytkowe systemu mikroprocesorowego. Na przykład pojemność licznika rozkazów określa przestrzeń adresową systemu mikroprocesorowego. Od budowy dekodera instrukcji i bloku sterowania zależy różnorodność instrukcji realizowanych przez system. Pojemność pamięci stosowej określa liczbę możliwych poziomów podprogramów, a rejestry indeksowe i rejestry ogólnego przeznaczenia w wielu wypadkach zmniejszają objętość programu.

2.2.4. Bloki wejścia/wyjścia

Bloki wejścia/wyjścia pośredniczą w wymianie informacji pomiędzy mikroprocesorem i pamięcią oraz urządzeniami zewnętrznymi. Zwykle system mikroprocesorowy pracuje z kilkoma blokami wejścia/wyjścia, a każdy z bloków obsługuje więcej niż jedno urządzenie peryferyjne. Każdemu urządzeniu zewnętrznemu przyporządkowuje się jego numer, zwany adresem.

Bloki wejścia/wyjścia pracują jako bufory danych przesyłanych lub odbieranych przez mikroprocesor do lub z urządzeń zewnętrznych. Odbierają z mikroprocesora informacje dotyczące kierunku transmisji. Dekodują adresy urządzeń zewnętrznych. Zapewniają synchronizację pomiędzy momentem wysłania /pobrania/ pojedynczej danej przez mikroprocesor i momentem pobrania /wysłania/ danej przez urządzenie zewnętrzne.

Stosuje się dwie techniki wymiany informacji pomiędzy urządzeniem zewnętrznym i systemem mikroprocesorowym. Pierwszą z nich jest programowane przesyłanie danych /tak zwane programowane we/wy/ - wykorzystywane do przesyłania pojedynczych danych. Przy programowanym we/wy blok wejścia/wyjścia pośredniczy w wymianie informacji pomiędzy mikroprocesorem i urządzeniem zewnętrznym. Drugą techniką jest przesyłanie danych z bezpośrednim dostępem do pamięci /tak zwane DMA/ - wykorzystywane do przesyłania bloków danych. Przy DMA blok wejścia/wyjścia pośredniczy, bez udziału mikroprocesora, w wymianie informacji pomiędzy pamięcią RAM i urządzeniem zewnętrznym. Poszczególne dane bloku danych przesyłane do urządzenia zewnętrznego /do pamięci RAM/ pobierane są /wpisywane są/ z /do/ kolejnych lokacji pamięci.

Programowane wejście/wyjście oraz DMA wymagają stosowania różnych bloków wejścia/wyjścia. Schemat blokowy bloku wejścia/wyjścia stosowanego przy programowanym przesyłaniu danych przedstawiony jest na rys. 6. Blok wejścia/wyjścia przyłączony jest do szyny danych i szyny sterującej mikroprocesora. Źródłem lub miejscem docelowym danej przy przesyłaniu jej między mikroprocesorem i urządzeniem zewnętrznym /w obu kierunkach/ jest akumulator. Do przesłania danej do lub z akumulatora wykorzystuje się specjalne instrukcje programowe, które służą do komunikacji z blokami wejścia/wyjścia. Instrukcje te zawierają informacje o kierunku przesyłania danych oraz o numerze urządzenia zewnętrznego.

Po pobraniu z pamięci instrukcji określającej przesłanie danej mikroprocesor wysyła rozkaz szyną sterującą do dekodera znajdującego się w bloku wejścia/wyjścia. Dekoder bloku wejścia/wyjścia na podstawie odebranego rozkazu uaktywnia jeden z kanałów multipleksera wejściowego lub demultipleksera wyjściowego. Każdy z kanałów ma możliwość przesłania wszystkich bitów danej. W następnych krokach programowych mikroprocesor sprawdza stan gotowości urządzenia zewnętrznego do wysłania lub pobrania danej. Jeśli urządzenie zewnętrzne jest już gotowe do prze-

stania lub przyjęcia danej, to odpowiednio dana z urządzenia zewnętrznego zostaje wczytana do akumulatora poprzez uprzednio uaktywniony kanał multipleksera wejściowego i szynę danych lub dana z szyny danych zostaje wczytana do bufora wyjściowego i odczytana przez urządzenie zewnętrzne poprzez uprzednio uaktywniony kanał demultipleksera wyjściowego.

Istnieje również możliwość przyłączenia bloku wejścia/wyjścia do szyny adresowej i szyny danych systemu mikroprocesorowego. Do przesyłania danych wykorzystuje się wtedy te instrukcje programowe, które służą do komunikacji z pamięcią mikroprocesora. W takim przypadku źródłem lub miejscem docelowym przy przesyłaniu danych z lub do mikroprocesora może być akumulator oraz inne rejestry mikroprocesora, które zdolne są pobierać lub przesyłać dane z lub do pamięci. Urządzenia zewnętrzne dekoduje się na podstawie ich numeru, który pojawia się na szynie adresowej. Adresy przyporządkowane urządzeniom zewnętrznym nie mogą być wykorzystywane do adresowania lokacji pamięci.

Na rysunku 7 przedstawiono schemat blokowy bloku wejścia/wyjścia stosowanego przy przesyłaniu danych bezpośrednio do lub z pamięci RAM z lub do urządzenia zewnętrznego. Blok wejścia/wyjścia tego typu nosi nazwę sterownika DMA. Poniżej naszkicowano ogólną zasadę jego działania.

Stan przerzutnika żądania DMA wskazuje, czy urządzenie zewnętrzne wysłało do systemu mikroprocesorowego sygnał żądania DMA, to znaczy sygnał żądania uzyskania bezpośredniego dostępu do pamięci. Sygnał żądania DMA ustawia przerzutnik żądania DMA w stan jedynki logicznej. Zerowanie przerzutnika żądania DMA odbywa się po zakończeniu przesyłania danych. Rejestr adresowy zawiera adres lokacji pamięci RAM z lub do której należy pobrać lub przesłać pierwsze słowo bloku danych. Po przesłaniu pojedynczej danej zawartość rejestru adresowego zwiększa się o jeden. Rejestr danych pracuje jako bufor pojedynczych danych. Sterowanie DMA pełni kilka funkcji. Sprawdza stan przerzutnika żądania DMA i jeśli stwierdzi, że przerzutnik żądania DMA znajduje się w stanie jedynki logicznej - sterowanie DMA wystawia do mikroprocesora sygnał żądania DMA, to znaczy sygnał żądania przejścia mikroprocesora w stan jałowy[1/]. Poza tym sterowanie DMA przyjmuje od mikroprocesora sygnał potwierdzenia DMA - to znaczy przejścia mikroprocesora w stan jałowy. Sterowanie DMA przyjmuje również od urządzenia zewnętrznego sygnał określający kierunek przesyłania danych oraz wysyła informację dotyczącą kierunku transmisji do pamięci RAM. Po zakończeniu przesyłania pojedynczej danej sterowanie DMA zwiększa zawartość rejestru adresowego o jeden. Zlicza liczbę przesłanych słów oraz zapewnia synchronizację przy przesyłaniu pojedynczych

---

[1/] Na czas przesyłania danych z bezpośrednim dostępem do pamięci mikroprocesor ustawia się w stan jałowy, to znaczy przerywa pracę, zachowuje jednak zawartości swoich rejestrów wewnętrznych oraz ustawia swoje wyjścia na szynę danych i szynę adresową w stan wysokiej impedancji.

danych. Po przesłaniu całego bloku danych sterowanie DMA zeruje przerzutnik żądania DMA oraz przerywa wysyłanie sygnału żądania DMA do mikroprocesora.

Sterownik DMA przyłącza się do szyny danych, szyny adresowej i szyny sterującej mikroprocesora. Sposób przyłączenia sterownika DMA do systemu mikroprocesorowego przedstawiono na rysunku 7.

Przed przesłaniem bloku danych, odpowiednim programem, wprowadza się do sterownika DMA informacje dotyczące objętości bloku danych oraz adresu lokacji pamięci RAM w lub z której należy umieścić lub pobrać pierwszą daną bloku danych.

2.2.5. Inne bloki systemu mikroprocesorowego

W celu rozszerzenia możliwości funkcjonalnych systemów mikroprocesorowych do minimalnego zestawu elementów można dołączać dodatkowe bloki. Należą do nich przetworniki A/C i przetworniki C/A, bloki do synchronicznej i asynchronicznej szeregowej transmisji danych, bloki sterowania systemem przerwań, programowane liczniki i czasomierze, multipleksery, dekodery, bufory, wzmacniacze szyn z wyjściami trójstanowymi itp. Opisy zasad pracy powyższych bloków oraz sposobów włączania ich do systemu mikroprocesorowego są dostępne w literaturze [1,2,9].

3. PODSUMOWANIE

Powyższe opracowanie zawiera szkicowe omówienie zagadnień związanych z architekturą systemów mikroprocesorowych. Bardziej szczegółowe informacje na temat technologii układów scalonych LSI, budowy mikroprocesorów oraz systemów mikroprocesorowych są dostępne w podanych poniżej pozycjach bibliograficznych.

Należy jednak zauważyć, że budowa systemów mikroprocesorowych stanowi jedynie część wiedzy koniecznej do zaprojektowania, a także użytkowania systemu mikroprocesorowego. Z uwagi na fakt, że system mikroprocesorowy wykonuje jedynie to, co w formie programu zawarte jest w jego pamięci, projektant oraz użytkownik systemu powinien zapoznać się z zagadnieniami dotyczącymi zasad budowy programu. Zagadnieniom tym poświęcony jest kolejny artykuł pt. "Programowanie systemów mikroprocesorowych".

WYKAZ LITERATURY

1. Osborne A.: An introduction to microcomputers. Berkeley Calif. 1976 Vol. 1.
2. Hilburn J.L., Julich P.N.: Microcomputers /Microprocessors: Hardware, software and applications. Englewood Cliffs: Prentice-Hall Inc. 1976.
3. Glynn D.R.Mc: Microprocessors technology, architecture and applications. A Wiley-Interscience Publications 1976.

4. Hackmeister D.: Focus on semiconductor RAMs. Electronic Design 1977 nr 17.

5. Winfield J.: Dynamic memories offer advantages over static RAMs. Electronic Design 1977 nr 14.

6. Altman L.: Memories. Electronics 1977 nr 20.

7. Muething G.: Designing the maximum performance into bit slice microcomputers. Electronics 1976 nr 30.

8. Amendt A.J.: The obsolence of the microprocessor. Microelectronics and Reliability Vol. 16 s. 323-332.

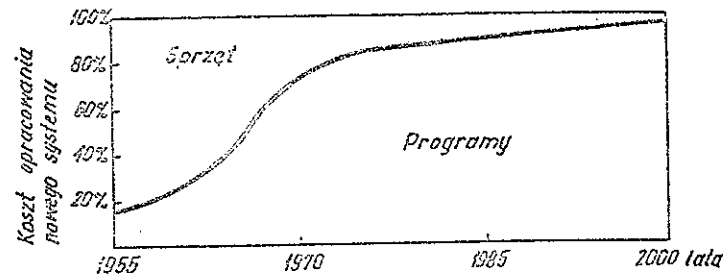
9. Guzman D.: Marry your μP to monolithic a/ds. Electronic Design 1977 nr 2.

10. Balph T.: Get the best processor performance by building it from ECL bit slices. Electronic Design 1977 nr 12.

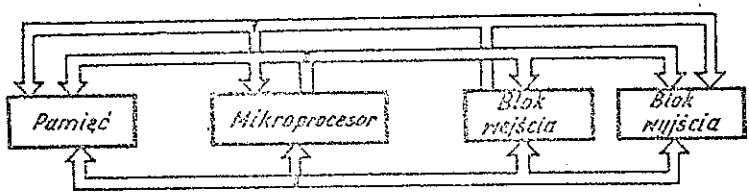
11. Gellender E.: Learn microprocessor fundamentals. Electronic Design 1977 nr 21.

12. Organizacja maszyn cyfrowych i mikroprogramowanie. Warszawa. WNT 1977.

13. Bursky D.: Microprocessor selection guide. Electronic Design 1977 nr 21.



Rys. 1. W całkowitym koszcie opracowania nowego systemu mikroprocesorowego coraz większą część stanowi koszt programów

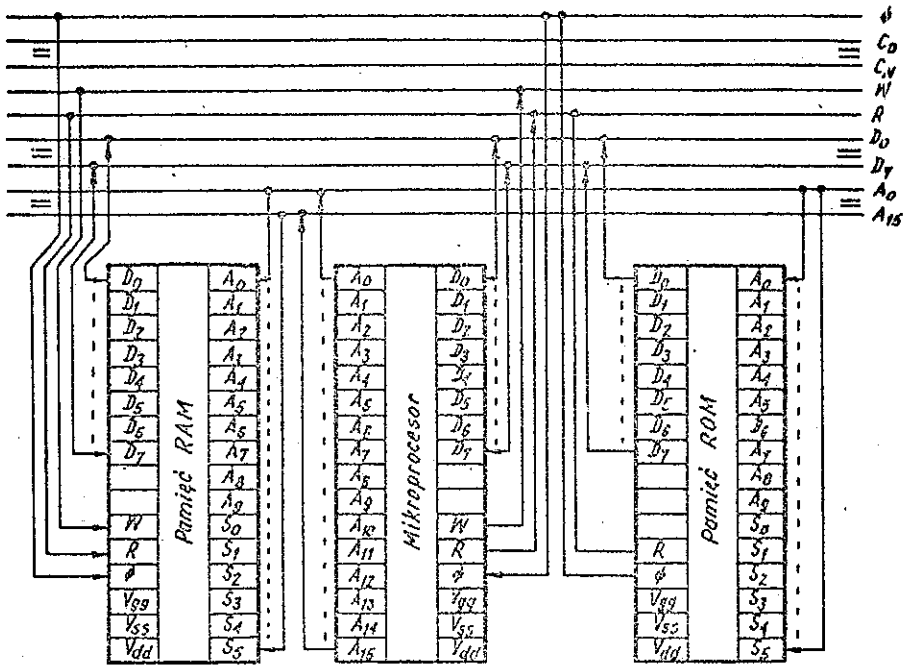


Rys. 2. Schemat blokowy systemu mikroprocesorowego o najmniejszej konfiguracji

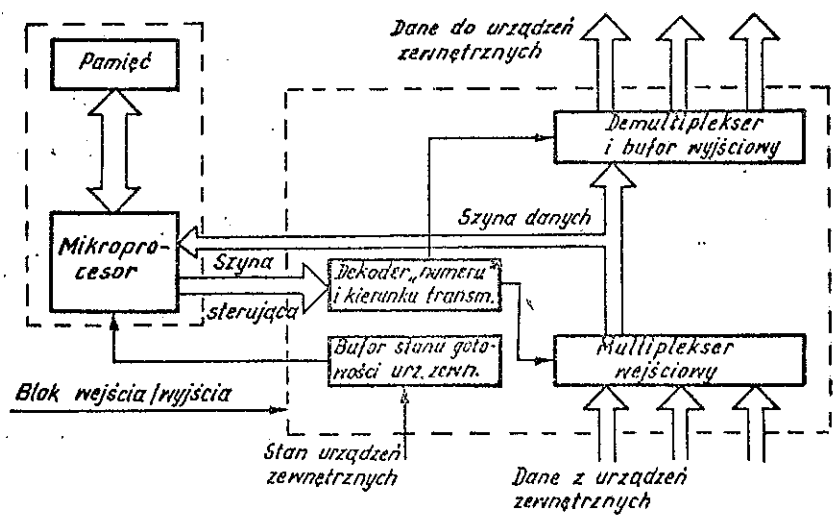
1 - szyna danych, 2 - szyna adresowa, 3 - szyna sterująca

Producent	Typ	Organizacja [bity]	Czas dostępu [μs]	Czas programowania [s]	Metoda wymazywania
Intel Co.	1702	256 x 8	0,5 ÷ 1,0	30 ÷ 100	światło nadfioletowe
	2704	1024 x 4	0,4 ÷ 1,0	30 ÷ 100	
	2708	1024 x 8	0,4 ÷ 1,0	30 ÷ 100	
Nitron	NC 7051	1024 x 1	2 ÷ 5	0,1 ÷ 0,5	elektrycznie
General Instr. Co.	ER 2400	1024 x 4	2	0,1 ÷ 0,2	
	ER 3400	1024 x 4	2	0,01	

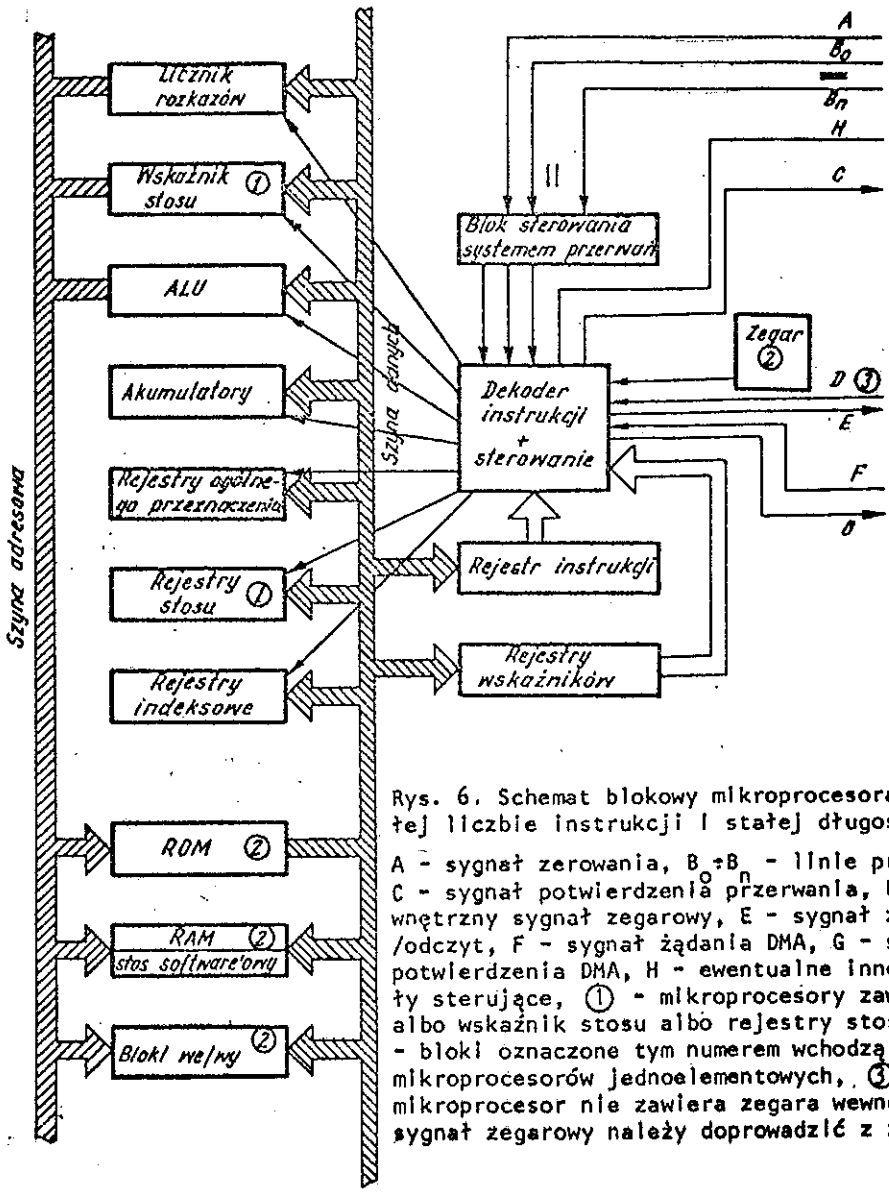
Rys. 3. Niektóre typy pamięci REPROM



Rys. 4. Schemat typowego połączenia mikroprocesora i pamięci RAM oraz ROM

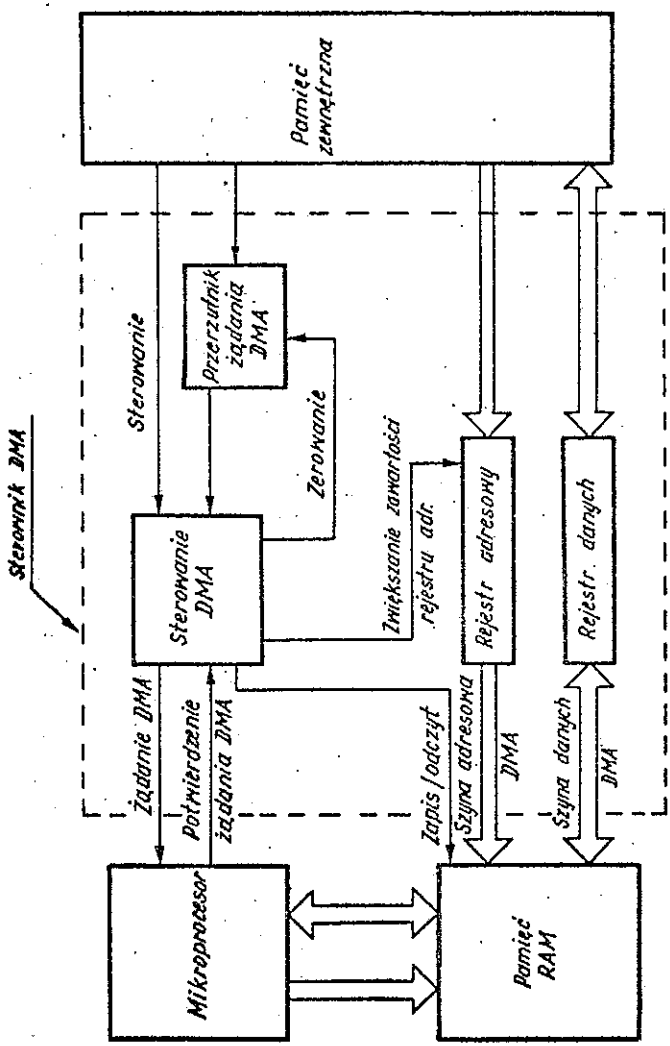


Rys. 5. Schemat blokowy bloku wejścia/wyjścia do programowanego przesyłania danych



Rys. 6. Schemat blokowy mikroprocesora o stałej liczbie instrukcji i stałej długości słowa

A - sygnał zerowania, $B_0 \div B_n$ - linie przerwań, C - sygnał potwierdzenia przerwania, D - zewnętrzny sygnał zegarowy, E - sygnał zapis/odczyt, F - sygnał żądania DMA, G - sygnał potwierdzenia DMA, H - ewentualne inne sygnały sterujące, ① - mikroprocesory zawierają albo wskaźnik stosu albo rejestry stosu, ② - bloki oznaczone tym numerem wchodzą w skład mikroprocesorów jednoelementowych, ③ - jeśli mikroprocesor nie zawiera zegara wewnętrznego, sygnał zegarowy należy doprowadzić z zewnątrz



Rys. 7. Schemat blokowy sterownika DMA

Maria Tyrowicz

PROGRAMOWANIE SYSTEMÓW MIKROPROCESOROWYCH

1. WSTĘP

System mikroprocesorowy podczas swojej pracy wykonuje kolejno operacje, które określone za pomocą instrukcji zostały uprzednio wprowadzone do jego pamięci. Uporządkowany zbiór instrukcji, opisujący zadanie, które ma wykonać system mikroprocesorowy nosi nazwę programu. Programowanie można określić jako zespół działań prowadzących do otrzymania poprawnego programu i umieszczenia go w pamięci stałej systemu mikroprocesorowego.

Poniższe opracowanie przedstawia niektóre wybrane zagadnienia związane z programowaniem systemów mikroprocesorowych[1/]. Omówiono w nim kolejne etapy tworzenia programu i podstawowe grupy języków programowania. Przedstawiono także różne metody adresowania pamięci oraz sposoby programowania operacji przesyłania danych z lub do mikroprocesora do lub z urządzeń zewnętrznych.

Tworzeniem programów na różnym poziomie budowy systemów mikroprocesorowych zajmują się producenci mikroprocesorowych układów LSI, producenci systemów mikroprocesorowych, producenci aparatury sterowanej systemem mikroprocesorowym, a także użytkownicy systemów mikroprocesorowych oraz użytkownicy aparatury sterowanej systemem mikroprocesorowym. Ilustruje to rysunek 1.[2/].

Producenci mikroprocesorowych układów scalonych LSI coraz częściej oferują już zaprogramowane pamięci stałe. W pamięciach tego typu umieszczone są z reguły pewne typowe programy, które mogą być przydatne w wielu zastosowaniach systemów mikroprocesorowych. Przykładem może być układ scalony pamięci stałej 8298 f-my Intel, który zawiera translator języka BASIC dla mikroprocesora Intel 8080.

Producenci systemów mikroprocesorowych oferują zwykle zestawy programów standardowych ułatwiających opracowywanie nowych programów dla poszczególnych, oferowanych przez nich systemów mikroprocesorowych. W skład tych programów wchodzą translatory języków symbolicznych, programy ułatwiające wyszukiwanie i usuwanie pomyłek w nowo powstających programach itp.

Producenci aparatury sterowanej systemami mikroprocesorowymi opracowują dla niej programy użytkowe zgodnie z zadanymi funkcjami tej aparatury.

1/ Wcześniejsze opracowanie poświęcone było tematowi: "Architektura systemów mikroprocesorowych". Następne dwa przedstawiają kolejne tematy "Przegląd mikroprocesorów i systemy wspomagające projektowanie" oraz "Wybrane zastosowania systemów mikroprocesorowych".

2/ Rysunki są zamieszczone na końcu artykułu.

Użytkownik mikroprocesorowych elementów LSI po zmontowaniu systemu lub użytkownik gotowego systemu mikroprocesorowego opracowuje zwykle własne programy użytkowe dostosowane do planowanych zastosowań systemu. Często użytkownicy tworzą także nowe programy dla zakupionej aparatury sterowanej systemem mikroprocesorowym w celu rozszerzenia lub zmiany możliwości funkcjonalnych tej aparatury.

2. ETAPY PROGRAMOWANIA

Przy opracowywaniu programu postępuje się według takich samych zasad, jakie są na ogół stosowane przy opracowywaniu nowego urządzenia elektronicznego. Określenia "programowanie" i "projektowanie programu" praktycznie są równoznaczne. Można przytoczyć szereg analogii pomiędzy kolejnymi fazami programowania i kolejnymi etapami projektowania i budowy aparatury elektronicznej.

Opracowanie nowego urządzenia wiąże się z wykonaniem prototypu, ze sporządzeniem dokumentacji technicznej, a także z opracowaniem instrukcji dla użytkownika, która powinna obejmować zasady obsługi i konserwacji tego urządzenia. Poszczególne etapy tworzenia prototypu, jak wiadomo, obejmują kolejno: opracowanie opisu funkcjonalnego urządzenia, wykonanie jego ogólnego, funkcjonalnego schematu blokowego, wybór metody rozwiązania poszczególnych bloków, wykonanie schematów blokowych, a następnie schematów ideowych każdego z bloków, montaż poszczególnych bloków z reguły w formie oddzielnych paneli, uruchamianie paneli, łączenie ze sobą poszczególnych paneli oraz uruchamianie całego prototypu.

Analogiczne etapy pracy można wyodrębnić w procesie tworzenia nowego programu dla systemu mikroprocesorowego. Początkowo wykonuje się szczegółowy z reguły słowny opis zadania, które ma być realizowane przez system mikroprocesorowy. W zadaniu tym wyodrębnia się następnie funkcjonalnie rozdzielne fragmenty w taki sposób, aby mogły być one dalej opracowywane równolegle, niezależnie od siebie. Wydzielanie rozdzielnych funkcjonalnie fragmentów programu ściśle odpowiada wykonywaniu ogólnego schematu blokowego urządzenia. W następnym etapie projektowania programu wybiera się metodę rozwiązania każdego z fragmentów oraz dla każdego z fragmentów opracowuje się algorytmy programowe.

Algorytmy programowe stanowią odpowiednik schematów blokowych poszczególnych funkcjonalnych części urządzenia. Należy zauważyć przy tym, że przy opracowywaniu algorytmów programowych coraz częściej zaleca się stosowanie tak zwanych struktur programowych [8, 9]. Po otrzymaniu algorytmicznego rozwiązania zagadnienia następuje etap kodowania algorytmów w wybranym języku programowania. Odpowiada to rozwiązywaniu schematów blokowych urządzenia elektronicznego w formie schematów ideowych. Po zakodowaniu algorytmów otrzymuje się bloki programów, które stanowią rozwiązania poszczególnych fragmentów zadania.

Kolejną fazę pracy nad programem stanowi wprowadzanie otrzymanych bloków pro-

gramów do pamięci RAM systemu mikroprocesorowego oraz sprawdzenie poprawności działania tych bloków w rzeczywistych warunkach pracy. Odpowiada to montażowi poszczególnych paneli urządzenia. Jeśli poszczególne bloki programów działają już poprawnie, łączy się je w jedną całość i sprawdza się poprawność pracy całego programu. Gdy cały program pracuje poprawnie, wprowadza się go do pamięci stałej. Odpowiada to łączeniu paneli, sprawdzaniu poprawności pracy całego urządzenia, wykonaniu paneli na płytkach drukowanych i końcowemu montażowi prototypu.

Widać więc, że w projektowaniu prototypu urządzenia oraz w projektowaniu programu można wyodrębnić cztery podstawowe stopnie rozwiązywania zagadnienia. Pierwszy stopień odpowiada analizie zagadnienia, drugi - budowie rozwiązania o charakterze ogólnym, trzeci - tworzeniu konkretnego rozwiązania szczegółowego, ostatni - uruchamianiu i sprawdzaniu zrealizowanego projektu.

W odniesieniu do projektowania programu pierwszy stopień odpowiada tworzeniu opisu zadania i podziału tego zadania na rozdzielne fragmenty. Drugi stopień odpowiada rozwiązywaniu fragmentów zadania w formie algorytmów. Trzeci - kodowaniu algorytmów w wybranym języku programowania, ostatni - wprowadzeniu programów do pamięci RAM systemu mikroprocesorowego, sprawdzeniu programów i wprowadzaniu ich do pamięci stałej systemu mikroprocesorowego.

Przy opracowywaniu programu, podobnie jak przy opracowywaniu prototypu, należy sprawdzać i weryfikować projekt na każdym etapie /a szczególnie w początkowych etapach/ jego powstawania.

Odpowiednikiem dokumentacji technicznej prototypu jest zbiór algorytmów pracy oraz wykaz instrukcji programowych i komentarzy dla poszczególnych fragmentów programu. Ponadto do programu powinna być dołączona instrukcja dla użytkownika, w której znajdowałyby się informacje na temat zasad pracy programu, sposobów wprowadzania i wyprowadzania danych itp. [1,2,8,9].

Na rysunku 2 przedstawiono, w formie algorytmu, kolejne etapy budowy programu.

3. JĘZYKI PROGRAMOWANIA I TRANSLATORY

3.1. Języki wewnętrzne mikroprocesorów

Przez język wewnętrzny mikroprocesora rozumie się zwykle zbiór instrukcji, który dany mikroprocesor może realizować. Każda z instrukcji tego zbioru określa operację, którą wykonuje mikroprocesor przy realizowaniu tej instrukcji. Wykaz wszystkich instrukcji języka wewnętrznego danego mikroprocesora nosi nazwę jego listy instrukcji.

Poszczególne instrukcje języka wewnętrznego odpowiadają określonym liczbom dwójkowym i są ściśle związane z budową dekodera instrukcji i bloku sterowania zawartych w mikroprocesorze. W związku z tym, w zasadzie każdy mikroprocesor dy-

sponuje swoją własną listą instrukcji. Należy podkreślić, że lista instrukcji stanowi bardzo istotny parametr mikroprocesora.

Program napisany w języku wewnętrznym mikroprocesora ma formę ciągu, a właściwie kolumny liczb dwójkowych. Poszczególne wiersze tej kolumny odpowiadają instrukcjom oraz danym. Długość każdego wiersza równa jest długości słowa mikroprocesora. W podobnej formie można wyobrazić sobie program wprowadzony do pamięci systemu mikroprocesorowego. Poszczególne wiersze kolumny zajmują oddzielne lokacje pamięci. Każdy następny wiersz zajmuje kolejną lokację pamięci.

Poszczególne instrukcje języka wewnętrznego składają się zwykle z dwóch części. Pierwszą stanowi tak zwany kod operacyjny instrukcji, a drugą tak zwany kod adresowy. Kod operacyjny instrukcji, który zawarty jest w jednym lub dwu kolejnych słowach mikroprocesora określa operację, która ma być wykonana przez mikroprocesor w czasie realizacji tej właśnie instrukcji oraz określa, w jaki sposób należy identyfikować kod adresowy instrukcji. Kod adresowy może bowiem określać adres danej, która ma wziąć udział w operacji opisanej kodem operacyjnym instrukcji, może być tą daną lub określać adres, którym należy załadować licznik rozkazów w czasie wykonywania tej instrukcji.

Instrukcje języka wewnętrznego ze względu na wykonywane operacje można podzielić na instrukcje przesunięć danych, instrukcje arytmetyczno-logiczne, instrukcje skoku i instrukcje sterowania procesorem. Obecnie krótko omówione będą powyższe typy instrukcji.

Instrukcje przesunięć danych

Instrukcje tej grupy służą do przesuwania danych z jednej wybranej lokacji, tak zwanej lokacji źródłowej, do innej wybranej lokacji, tak zwanej lokacji docelowej. Lokacją źródłową lub docelową może być akumulator, dowolny rejestr lub para rejestrów w obrębie mikroprocesora, dowolna lokacja pamięci lub para kolejnych lokacji pamięci, a także urządzenie zewnętrzne dołączone do bloku wejścia/wyjścia. Ponadto źródłem danej może być tak zwana dana natychmiastowa, to znaczy dana, która występuje w instrukcji bezpośrednio za kodem operacyjnym. Należy zauważyć, że przy wykonywaniu instrukcji przesunięć danych nie zmienia się zawartość rejestru wskaźników mikroprocesora.

Instrukcje arytmetyczno - logiczne

Instrukcje tej grupy służą do modyfikacji zawartości wewnętrznych rejestrów i wskaźników mikroprocesora za pomocą dokonywania na nich operacji arytmetycznych lub logicznych.

Instrukcje arytmetyczno-logiczne działają na zawartości jednej lub dwóch wybranych lokacji źródłowych. Można je w związku z tym podzielić na instrukcje,

które wymagają jednego lub dwu tzw. operandów. W związku z tym można wyróżnić instrukcje jednooperandowe i dwuoperandowe. Wynik operacji arytmetycznej lub logicznej umieszczany jest w określonej lokacji docelowej. Jedną z lokacji źródłowych, jak również lokacją docelową jest zwykle akumulator mikroprocesora. Drugą lokacją źródłową może być wybrana lokacja pamięci, urządzenie zewnętrzne dołączone do bloku wejścia wyjścia lub dana natychmiastowa. O wyborze drugiej lokacji źródłowej decyduje kod operacyjny instrukcji.

W wyniku operacji arytmetycznych lub logicznych odpowiednio są zerowane lub ustawiane poszczególne przerzutniki rejestru wskaźników. W związku z tym, aby przetestować daną wejściową lub zawartość pamięci należy wykonać na niej operację arytmetyczną lub logiczną, a następnie programowo zbadać stan wskaźników.

I n s t r u k c j e s k o k u

Instrukcje skoku służą do zmiany kolejności wykonywania instrukcji w stosunku do ułożenia ich w pamięci. Nosi to niekiedy również nazwę przesunięcia sterowania do innego miejsca pamięci.

Przy wykonywaniu instrukcji mikroprocesor wprowadza do swojego licznika rozkazów adres zawarty w instrukcji skoku. Następnie jako kolejną instrukcję do wykonania pobiera z pamięci tę instrukcję, która znajduje się w lokacji pamięci określonej adresem skoku.

Przesunięcie sterowania może mieć charakter powrotny - nosi ono wtedy nazwę skoku do podprogramu lub bezpowrotny - nosi ono wtedy nazwę skoku programowego lub krócej - skoku.

Podprogram jest fragmentem programu, który realizuje część całego zadania. Stanowi on pewną sekwencję rozkazów, która może być - dzięki instrukcjom skoku powrotnego - wykorzystywana wielokrotnie w czasie wykonywania programu pomimo to, że występuje tylko w jednym obszarze pamięci. Przy wykonywaniu instrukcji skoku do podprogramu zostaje bowiem zapamiętany adres lokacji pamięci, w której znajduje się ta właśnie instrukcja skoku. Adres ten - tak zwany adres powrotny - służy do powrotu do miejsca pamięci, z którego odbył się skok po wykonaniu podprogramu - tak zwanego powrotu z podprogramu. Adres powrotny pamiętany jest w rejestrach stosu w obrębie mikroprocesora lub w lokacji pamięci RAM adresowanej wskaźnikiem stosu. Aby umożliwić powrót z podprogramu po jego wykonaniu, podprogramy kończy się specjalną instrukcją powrotu. Przy wykonywaniu tej instrukcji mikroprocesor pobiera ze stosu adres powrotny i wprowadza go do swojego licznika rozkazów.

W czasie wykonywania instrukcji skoku programowego mikroprocesor wprowadza do swojego licznika rozkazów adres zawarty w części adresowej instrukcji. Nie jest wtedy pamiętany adres powrotny.

Instrukcje skoków programowych oraz skoków do programów są zwykle dostępne w formie warunkowej i bezwarunkowej. Skoki bezwarunkowe są wykonywane zawsze. Wykonanie skoków warunkowych jest uzależnione od aktualnego stanu rejestru wskaźników. Skok się odbywa przy spełnieniu odpowiedniego warunku zawartego w kodzie operacyjnym instrukcji. Warunkiem wykonania skoku jest żądany stan wybranego przerzutnika rejestru wskaźników mikroprocesora. Jeśli wymagany w instrukcji warunek nie jest spełniony, mikroprocesor ignoruje instrukcję skoku i przechodzi do wykonania kolejnej instrukcji.

I n s t r u k c j e s t e r o w a n i a p r o c e s o r e m

Instrukcje tej grupy umożliwiają bezpośrednie oddziaływanie na pracę mikroprocesora. Obejmują zwykle instrukcję uaktywniania systemu przerwań, instrukcję dezaktywacji systemu przerwań, instrukcję całkowitego zatrzymania pracy mikroprocesora, instrukcję wyzerowania mikroprocesora oraz instrukcję wstrzymania pracy procesora na czas realizacji jednej instrukcji.

Jeśli system przerwań mikroprocesora jest uaktywniony, to mikroprocesor - po przyjęciu od urządzenia zewnętrznego sygnału żądania przerwania - przerywa wykonywany program, przechodzi do wykonywania podprogramu obsługi przerwania. Po ukończeniu podprogramu obsługi przerwania mikroprocesor powraca do uprzednio wykonywanego programu. Na czas wykonywania podprogramu obsługi przerwania mikroprocesor odsyła na stos zawartości swoich wewnętrznych liczników i rejestrów. Jeśli system przerwań jest zdezaktywowany, mikroprocesor ignoruje sygnały żądania przerwania i nie przerywa wykonywania programu.

Instrukcja zatrzymania pracy mikroprocesora całkowicie wstrzymuje jego pracę. Zwykle wyjście z takiego stanu jest możliwe wyłącznie przez doprowadzenie do mikroprocesora odpowiedniego sygnału /sygnału żądania obsługi przerwania/.

Wyzerowanie mikroprocesora - to znaczy wyzerowanie jego rejestrów i liczników wewnętrznych prowadzi do tzw. restartu systemu mikroprocesorowego. Po ustawieniu zerowej zawartości licznika programu system mikroprocesorowy rozpoczyna wykonywanie programu od początku.

Instrukcja wstrzymania pracy mikroprocesora na czas wykonania jednej - tej właśnie - instrukcji nie powoduje żadnych zmian zawartości rejestrów czy liczników mikroprocesora. Stosuje się ją do wprowadzania znanych opóźnień czasowych w pętlach programowych lub w celu pozostawienia miejsca w programie na późniejsze uzupełnienia.

3.2. Języki symboliczne

Programowanie z wykorzystaniem dwójkowej formy języka wewnętrznego jest ogromnie pracochłonne i uciążliwe. Stwarza możliwość popełniania pomyłek oraz utrudnia

wprowadzanie zmian w czasie pisania programu. Postać dwójkowa poszczególnych instrukcji jest szczególnie trudna do zapamiętania i niewygodna przy sprawdzaniu programu. W związku z tym często do opisu poszczególnych instrukcji stosuje się notację ósemkową lub szesnastkową zapisu liczb dwójkowych /rys. 3/. W notacji ósemkowej poszczególnym kolejnym trzyznakowym grupom bitów liczby dwójkowej przyporządkowuje się ich odpowiedniki dziesiętne. Na przykład liczbie dwójkowej równej 10,11.1,001 odpowiada liczba ósemkowa równa 271. Przecinki wskazują, w jaki sposób dzieli się liczbę dwójkową na grupy trzyznakowe. W notacji szesnastkowej poszczególnym kolejnym czteroznakowym grupom bitów liczby dwójkowej przyporządkowuje się ich odpowiedniki cyfrowe 0,...,9 oraz literowe A,...,F. Przedstawionej wyżej liczbie dwójkowej odpowiada liczba szesnastkowa równa B9. Kropką rozdzielono całą liczbę dwójkową na grupy czteroznakowe.

Ósemkowy lub szesnastkowy zapis dwójkowych instrukcji i danych wymaga mniejszej liczby znaków. Dzięki temu zmniejsza się liczba pomyłek, pisanie programu trwa krócej i jest mniej uciążliwe. Program jest jednak w dalszym ciągu słabo czytelny. Dodatkową wadę języków maszynowych stanowi przyporządkowywanie poszczególnym instrukcjom i danym - w czasie pisania programu - adresów rzeczywistych, to znaczy dokładnie takich, jakie będą im przyporządkowane w pamięci systemu mikroprocesorowego. Wskutek tego każda zmiana w programie, która wywołuje zmiany w tych adresach /na przykład usunięcie z wnętrza programu grupy instrukcji lub wprowadzenie do wnętrza programu na przykład nowych instrukcji/, wymaga dokonania licznych poprawek w obrębie całego programu. Należy bowiem dokonać zmian w tych wszystkich instrukcjach, które odwołują się do określonych lokacji pamięci /instrukcje skoku, instrukcje zawierające adres danej/.

Aby ułatwić pamiętanie kodów instrukcji, a w związku z tym skrócić czas pisania i sprawdzenia programu, zwiększyć czytelność programu oraz uniknąć kłopotów wynikających z korzystania z adresów rzeczywistych, opracowano tak zwane języki symboliczne.

W językach symbolicznych poszczególne instrukcje lub grupy instrukcji języka wewnętrznego, adresy oraz dane wyraża się za pomocą kodów literowych lub kodów literowo-cyfrowych. Wśród języków symbolicznych można wyróżnić trzy podstawowe rodziny języków. Są to języki typu assembler, typu macroassembler oraz języki wyższego rzędu.

Języki typu assembler należą do grupy języków symbolicznych, ściśle odpowiadających językom wewnętrznym mikroprocesorów, a więc poszczególne instrukcje języków typu assembler odpowiadają poszczególnym instrukcjom języka wewnętrznego. Na rysunku 4 przedstawiono kilka instrukcji języka wewnętrznego mikroprocesora 8080. Podano kody dwójkowe tych instrukcji oraz ich odpowiedniki w notacji ósemkowej, szesnastkowej oraz w języku typu assembler mikroprocesora 8080.

Języki typu macroassembler tworzy się dla określonych zastosowań. Pojedyncze

Instrukcje języka macroassembler - macroinstrukcje odpowiadają grupom instrukcji języka assembler. Jeśli na przykład przy programowym rozwiązywaniu zagadnienia stwierdzi się, że często występują takie same sekwencje programowe, można je zastąpić macroinstrukcjami. W czasie pisania programu operuje się utworzonymi macroinstrukcjami.

Języki wyższego rzędu stanowią grupę języków, tak zwanych zorientowanych problemowo. Tworzy się je pod kątem programowego rozwiązywania określonych typów zagadnień. Istnieją języki wyższego rzędu, szczególnie wygodne przy programowaniu zagadnień technicznych, ekonomicznych czy symulacyjnych. Są to na przykład języki Fortran, Algol, Cobol, Simula itp. Poszczególne języki mają swoje poszczególne składnie dostosowane do charakteru języka.

Aby można było efektywnie korzystać z języków symbolicznych przy kodowaniu algorytmów programowych, musi istnieć możliwość szybkiej zamiany programu zrealizowanego w języku symbolicznym na program zapisany w dwójkowym języku wewnętrznym mikroprocesora. Służą do tego specjalne programy tłumaczące, zrealizowane w językach wewnętrznych mikroprocesorów - tak zwane translatory. Proces przekładu programu napisanego w języku symbolicznym na program w języku wewnętrznym mikroprocesora nosi nazwę translacji. Program poddawany procesowi translacji nosi nazwę programu źródłowego, a program uzyskiwany w wyniku translacji nosi nazwę programu wynikowego.

Translatory traktują program źródłowy jako pewnego rodzaju zbiór danych. Na podstawie programu źródłowego oraz dodatkowych informacji o rzeczywistych wartościach danych symbolicznych i informacji określającej adres rzeczywisty pierwszej instrukcji programu wynikowego translatory generują program wynikowy w formie zbioru dwójkowych instrukcji języka wewnętrznego. Oczywiście każdy z języków symbolicznych wymaga dla danego mikroprocesora oddzielnego translatora [1,2,3,4,5].

4. METODY ADRESOWANIA PAMIĘCI

Dla wykonania na danej żądanej operacji należy uprzednio pobrać tę daną z pamięci i przekazać ją do akumulatora mikroprocesora. Aby pobrać żądaną daną z pamięci należy zastosować odpowiednią instrukcję przesunięcia danej z podaniem adresu lokacji pamięci, w której znajduje się ta dana. Do określania adresu danej wykorzystuje się informację występującą w części adresowej instrukcji przesunięcia danej. Kod operacyjny tej instrukcji, poza symbolem operacji przesunięcia danej, określa również, w jaki sposób należy spożytkować informację z części adresowej instrukcji do identyfikacji adresu lokacji, w której znajduje się dana - to znaczy do utworzenia tak zwanego adresu efektywnego. Sposoby tworzenia adresu efektywnego w oparciu o adres zawarty w instrukcji noszą nazwę metod adresowania pamięci. Istnieje szereg metod adresowania pamięci: metoda bezpośrednia,

pośrednia pamięciowa,pośrednia rejestrowa,natychmiastowa,względna,indeksowa,stronicowa strony zerowej, stronicowa strony bieżącej,stronicowa strony ustawionej.

Przy adresowaniu bezpośrednim, informacja umieszczona w części adresowej bieżącej instrukcji jest adresem efektywnym.

Przy adresowaniu pośrednim pamięciowym adres zawarty w instrukcji wskazuje lokację pamięci, w której znajduje się część bitów adresu efektywnego. Druga część adresu efektywnego znajduje się w następnej lokacji pamięci.

Przy adresowaniu pośrednim rejestrowym adres efektywny pobiera się z jednej z par rejestrów ogólnego przeznaczenia w obrębie mikroprocesora. Adres zawarty w instrukcji służy do identyfikacji jednej z par rejestrów ogólnego przeznaczenia, jako źródła adresu efektywnego. Oczywiście tę parę rejestrów należy uprzednio programowo załadować żądanym adresem.

Przy adresowaniu natychmiastowym daną stanowi informacja występująca w części adresowej instrukcji - jest to tak zwana dana natychmiastowa.

Przy adresowaniu względnym adres efektywny powstaje w wyniku zsumowania danej natychmiastowej i aktualnej zawartości licznika programu. Dana natychmiastowa określa położenie żądanej danej w pamięci względem bieżącej instrukcji.

Przy adresowaniu indeksowym adres efektywny powstaje w wyniku zsumowania danej natychmiastowej i zawartości rejestru indeksowego znajdującego się w obrębie mikroprocesora. Rejestr indeksowy należy uprzednio programowo załadować żądaną wartością.

Przy adresowaniu stronicowym zakłada się podział pamięci na "strony", tzn. obszary o ustalonej i jednakowej długości. Objętość pojedynczej strony równa się 2^x, gdzie x jest długością słowa mikroprocesorowego. Do tworzenia adresu efektywnego wykorzystuje się daną natychmiastową występującą w instrukcji. Dana natychmiastowa określa adres lokacji danej:

a/ na zerowej stronie pamięci - przy adresowaniu stroną zerową,

b/ na stronie pamięci, na której znajduje się bieżąca instrukcja - przy adresowaniu stroną bieżącą,

c/ na stronie pamięci, której numer znajduje się w rejestrze strony w obrębie mikroprocesora - przy adresowaniu stroną ustawioną.

Optymalne wykorzystanie metod adresowania pamięci sprzyja maksymalnemu wykorzystaniu systemu. Zaleca się stosowanie adresowania natychmiastowego dla stałych danych, indeksowego i pośredniego rejestrowanego do przetwarzania tablic danych, względnego, gdy zachodzi potrzeba stałego wzajemnego usytuowania danej i instrukcji w pamięci, stronicowego, gdy zależy na szybkim tworzeniu adresów efektywnych [6].

5. PROGRAMOWANIE OPERACJI WEJŚCIA/WYJŚCIA

Przy programowaniu operacji wejścia/wyjścia ma się do wyboru kilka metod przesyłania danych - synchroniczne przesyłanie danych, asynchroniczne przesyłanie danych oraz przesyłanie danych przy wykorzystaniu techniki przerwań.

Synchroniczne przesyłanie danych stosuje się, jeśli wiadomo, że urządzenie zewnętrzne jest gotowe do transmisji danej. Program transmisji danej sprowadza się do pojedynczych instrukcji: czytaj we/wy, pisz we/wy - dla wejścia/wyjścia programowanego akumulatorowego lub czytaj pamięć, pisz pamięć - dla wejścia/wyjścia programowanego pamięciowo. Na przykład, w języku assembler mikroprocesora 8080, program transmisji danej przedstawia się następująco:

```
      .
      .
      .
IN2   /czytaj daną z urządzenia we/wy nr 2/
      .
      .
      .
OUT5  /pisz daną do urządzenia we/wy nr 5/
      .
      .
      .
```

Przy asynchronicznym przesyłaniu danych mikroprocesor, bezpośrednio przed każdą transmisją, sprawdza stan gotowości urządzenia zewnętrznego do transmisji danej. Jeśli urządzenie zewnętrzne jest w stanie gotowości, następuje przesłanie tej danej.

P r z y k ł a d 1

Załóżmy, że przetwornik A/C jest dołączony do bloku we/wy jako urządzenie zewnętrzne o numerze 2. Do innego wejścia bloku we/wy należy doprowadzić przewód, na którym występuje sygnał gotowości przetwornika. Załóżmy, że przewód dołączono do wejścia czwartego bitu urządzenia zewnętrznego o numerze 3, a sygnał jedynki logicznej oznacza stan gotowości. Program czytania danej z przetwornika, zgodnie z wyżej wymienionymi założeniami, wygląda następująco /język assembler mikroprocesora 8080/ [12].

TEST:IN:3: - wczytaj daną z urządzenia zewnętrznego o numerze 3;

ANI:10H: - mnóż logicznie wartość wczytaną przez 00010000, /jedynka występuje na bicie sygnału gotowości, 10H oznacza 10 w notacji szesnastkowej/;

JZ:TEST: - jeśli w wyniku mnożenia otrzymano zerową wartość akumulatora /brak sygnału gotowości/ skocz do rozkazu o etykiecie TEST;

IN2: - jeśli w wyniku mnożenia nie otrzymano zerowej wartości akumulatora /wczytano 1 na czwartym bicie/, czytaj daną z urządzenia zewnętrz-

Technikę przerwań jako metodę współpracy z urządzeniami zewnętrznymi stosuje się, jeśli nie wiadomo, w jakim momencie urządzenie zewnętrzne będzie gotowe do transmisji danej. Jeżeli urządzenie zewnętrzne jest gotowe do transmisji danej - - wysyła do mikroprocesora sygnał żądania przerwania. Mikroprocesor , jeśli jego system przerwań jest uaktywniony, sprawdza stan wejścia sygnału żądania przerwania na początku każdej instrukcji. Algorytm /rys. 5/ przedstawia kolejne kroki realizacji transmisji danej przy wykorzystaniu techniki przerwań.

P r z y k ł a d 2

Załóżmy, że urządzenie zewnętrzne o numerze 2 przyłączone jest do systemu przerwań mikroprocesorem 8080. Mikroprocesor 8080 wykonuje m.in. jednobajtową instrukcję RST "nr" /RESTART z lokacji "nr" pamięci. "Nr" jest liczbą całkowitą z przedziału $\langle 0,7 \rangle$. Po wysyłaniu sygnału żądania przerwania i po otrzymaniu sygnału potwierdzenia, urządzenie zewnętrzne nr 2 wysyła na szynę danych kod instrukcji RST2. Mikroprocesor przechodzi do wykonania tej instrukcji. Ponieważ instrukcja RST2 jest typu przywołania podprogramu, zawartość licznika rozkazów jest automatycznie odsyłana na stos, a sterowanie przechodzi do lokacji pamięci, której adres znajduje się w lokacji 2 pamięci /w lokacji 2 i następnej umieszczony jest rozkaz skoku i adres skoku/. Poniżej przedstawiono fragment programu obsługi przerwania w języku assembler mikroprocesora 8080 [2]

```
"nr" : JMP   INTR2;   skok do podprogramu obsługi przerwania od urządzenia nr 2

                      podprogram obsługi przerwania

INTR2 :  PUSH B;
         PUSH C;      odesłanie zawartości rejestrów wewnętrznych mikroproce-
         PUSH H;      sora na stos
         PUSH PSW;
           .  }
           .  }       podprogram obsługi transmisji danej
           .  }
         POP PSW;
         POP H;       pobranie zawartości rejestrów wewnętrznych mikroprocesora
         POP C;       ze stosu
         POP B;
         EI;          uaktywnienie systemu przerwań
         RET;         powrót do uprzednio wykonywanego programu głównego.
```

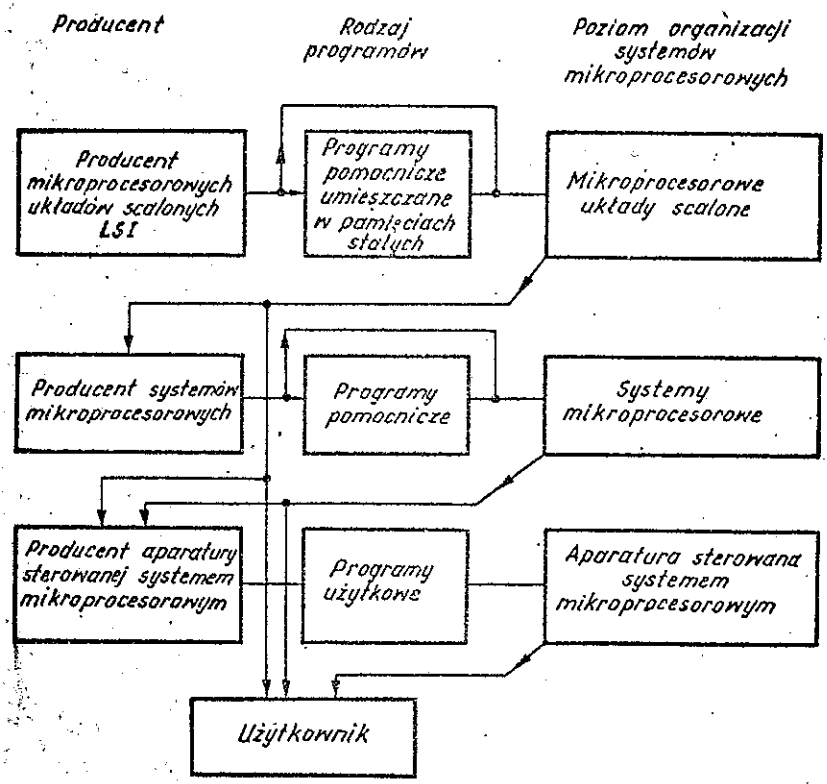
6. PODSUMOWANIE

Programowanie odgrywa dominującą rolę w procesie tworzenia nowego systemu mikroprocesorowego. Program określa możliwości użytkowe systemu. Objętość programu wpływa, poprzez liczbę koniecznych elementów pamięciowych, na całkowity koszt systemu. Odpowiednia budowa programu, to znaczy zastosowanie tak zwanych struktur programowych, ułatwia późniejsze modyfikacje programu.

Powyższy materiał zawiera szkicowe omówienie podstawowych zagadnień związanych z programowaniem systemów mikroprocesorowych. Bogate informacje na ten temat dostępne są w podanych poniżej pozycjach bibliograficznych.

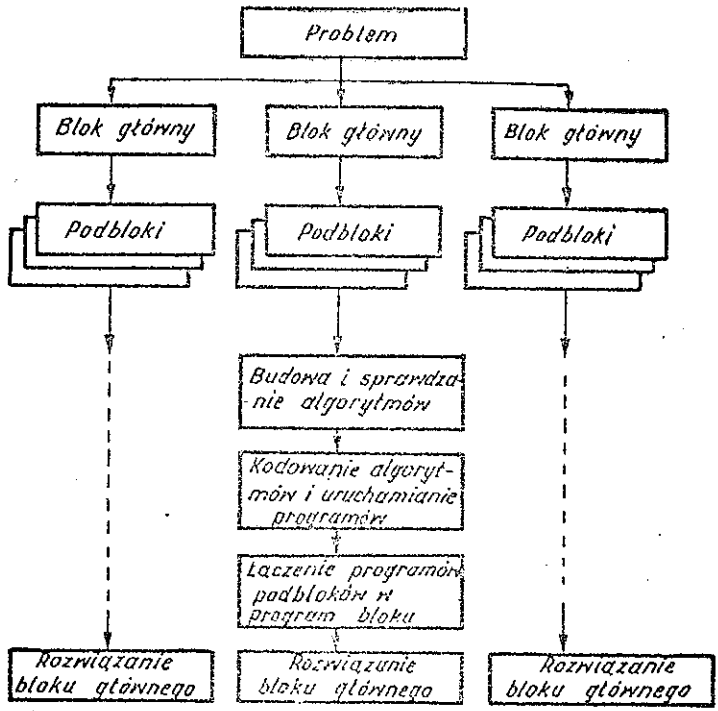
WYKAZ LITERATURY

1. Osborne A.: An Introduction to microcomputers. Basic Concepts, Berkeley California 1976 Vol. 1.
2. Hilbura J.L., Julich P.N.: Microcomputers/Microprocessors: Hardware, software and applications. Englewood Cliffs. Prentice-Hall Inc. 1976.
3. Glynn D.R.Mc: Microprocessors technology, architecture and applications. A Wiley - Interscience Publications, John Wiley Sons Inc. 1976.
4. Ulrickson R.W.: Software is a vital part of any computer. Electronic Design 1977 nr 1.
5. Perrin R.A.: High-level languages and the microprocessor. Electronic Eng. May 1977.
6. Leventhal L.: Put microprocessor software to work. Electronic Design 1977 nr 16.
7. Leventhal L.: Cut your processors computation time. Electronic Design 1977 nr 17.
8. Ulrickson R.: Solve software problems step by step. Electronic Design 1977 nr 2.
9. Ulrickson R.: Software modules are the building blocks. Electronic Design 1977 nr 3.
10. Faure J.C.: Zarys oprogramowania maszyn cyfrowych. Warszawa: WNT 1977.

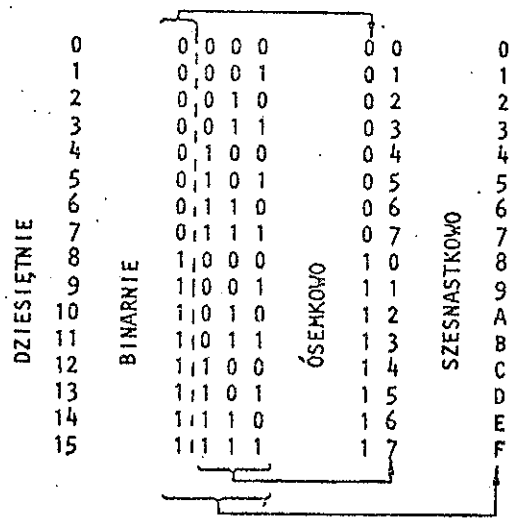


Rys. 1. Schemat poglądowy ilustrujący, kto i na jakim poziomie organizacji systemów mikroprocesorowych zajmuje się ich programowaniem

Użytkownik może mieć do czynienia z oprogramowanymi lub nie oprogramowanymi mikroprocesorowymi układami scalonymi, systemami mikroprocesorowymi lub aparaturą sterowaną systemem mikroprocesorowym wyposażoną w program użytkowy



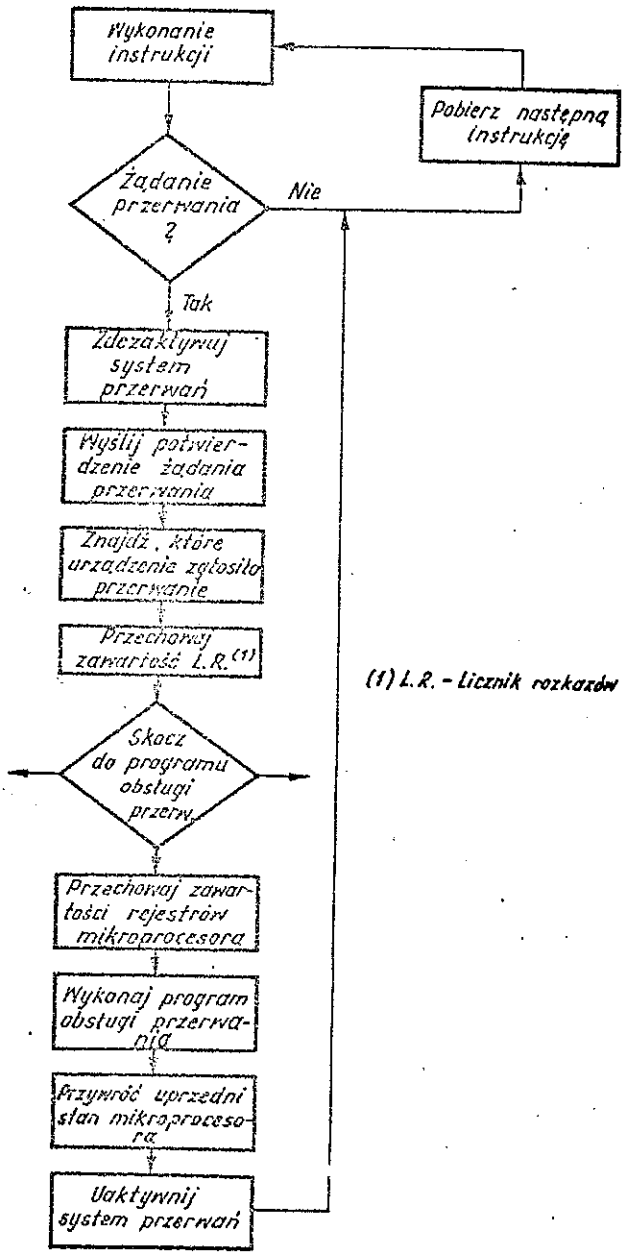
Rys. 2. Algorytm procesu tworzenia programu



Rys. 3. Ósemkowy i szesnastkowy zapis liczb dwójkowych

Nazwa instrukcji	Postacie kodu operacyjnego			
	Dwójkowa	Osemkowa	Szestnastkowa	Język assembler
Skok bezwarunkowy	1 1 Ø Ø Ø Ø 1 1	303	C3	JMP
Skok, jeśli bit przeniesienia równy jest 1	1 1 Ø 1 1 Ø 1 Ø	332	DA	JC
Skok, jeśli bit przeniesienia równy jest Ø	1 1 Ø 1 Ø Ø 1 Ø	322	D2	JNC
Skok, jeśli zawartość akumulatora równa jest Ø	1 1 Ø Ø 1 Ø 1 Ø	312	CA	JZ
Skok, jeśli zawartość akumulatora nie równa jest Ø	1 1 Ø Ø Ø Ø 1 Ø	302	C2	JNZ
Skok, jeśli liczba w akumulatorze jest dodatnia	1 1 1 1 Ø Ø 1 Ø	362	F2	JP
Skok, jeśli liczba w akumulatorze jesz ujemna	1 1 1 1 1 Ø 1 Ø	372	FA	JM
Skok, jeśli akumulator zawiera parzystą liczbę jedynek	1 1 1 Ø 1 Ø 1 Ø	352	EA	JPE
Skok, jeśli akumulator zawiera nieparzystą liczbę jedynek	1 1 1 Ø Ø Ø 1 Ø	342	E2	JPO

Rys. 4. Kody operacyjne instrukcji skoku mikroprocesora Intel 8080



Rys. 5. Ogólny algorytm procesu obsługi żądania przerwania