

POLSKIE TOWARZYSTWO INFORMATYCZNE
CENTRUM SZKOLENIA INFORMATYCZNEGO
ZETO-ŁÓDŹ

MIKROKOMPUTERY SZESNASTOBITOWE

Architektura systemów i układy rodziny
mikroprocesorowej Intel 8086

Materiały VI Szkoły Mikroprocesorowej
pod redakcją Wojciecha Cellarego

3—5 grudnia 1984 r.

POLSKIE TOWARZYSTWO INFORMATYCZNE
CENTRUM SZKOLENIA INFORMATYCZNEGO
ZETO-ŁÓDŹ

MIKROKOMPUTERY SZESNASTOBITOWE

Architektura systemów i układy rodziny
mikroprocesorowej Intel 8086

Materiały VI Szkoły Mikroprocesorowej
pod redakcją Wojciecha Cellarego

3—5 grudnia 1984 r.

POLSKIE TOWARZYSTWO INFORMATYCZNE
CENTRUM SZKOLENIA INFORMATYCZNEGO
ZRYTO-LODY

MIKROKOMPUTERY SZESNASTOBITOWE

Architektura systemów i układy robocze
mikroprocesorowej Intel 8086

Materiały VI Szkoły Mikroprocesorowej
pod redakcją Wojciecha Gellera

3-5 grudnia 1984 r.

PROGRAM SZKOŁY

1. Wojciech Cellary	
Architektura systemów wielomikroprocesorowych	17 5
2. Piotr Misiurewicz	
Mikroprocesor centralny Intel 8086/8088	10 22
3. Zbigniew Barański	
Układy rodziny mikroprocesorowej Intel 8086	32
4. Marek Tadeusz Jankowski	
Standard magistrali Multibus i standardy pochodne	58
5. Jerzy Kręglewski	
Hierarchiczny system przerwań w systemach wielomikroprocesorowych	73
6. Mieczysław Drabowski, Leszek Szklarczyk	
Koprocesor arytmetyczny Intel 8087	90
7. Piotr Krzyżagórski	
Mikroprocesor wejścia/wyjścia Intel 8089	103
8. Roman Trehciński	
Standard magistrali Multibus II	121
9. Grzegorz Dzięglewski, Krzysztof Rzymkowski, Janusz Zalewski	
Organizacja logiczna systemu Multibus II	130
10. Krzysztof Buchała	
Wprowadzenie do emulacji mikroprocesorów szesnastobitowych na przykładzie procesora 8086	140
11. Jerzy Brzeziński, Wojciech Cellary, Jerzy Kręglewski	
Wielomikroprocesorowy system 8/16 bitowy	158
12. Andrzej Syrczyński	
Pakiet jednostki centralnej MM86 systemu MIR-PROWAY	166
13. Krzysztof M. Święcicki	
Układ sterowania numerycznego obrabiarek "NUXON 500".	177

SŁOWO WSTĘPNE

W środowisku informatyków nie trzeba nikogo przekonywać, jak wielkie znaczenie dla rozwoju nauki i techniki ma tak zwana rewolucja mikroprocesorowa i jak istotne pociąga ona za sobą zmiany cywilizacyjne. Technika mikroprocesorowa liczy obecnie około piętnastu lat i te piętnaście lat jest okresem niespotykanego w innych dziedzinach nauki i techniki rozwoju zarówno w sferze technologii jak i zastosowań mikroprocesorów.

W rozwoju techniki mikroprocesorowej można wyróżnić pewne momenty, w których ulega ona skokowemu przyspieszeniu. Momenty te związane są ze zmianą generacji mikroprocesorów, umownie obejmujących mikroprocesory o danej długości słowa maszynowego. Mniej więcej pięć lat temu dokonana się taka zmiana generacji mikroprocesorów na skutek pojawienia się mikroprocesorów szesnastobitowych. W chwili obecnej, a więc po pięciu latach od chwili pojawienia się mikroprocesorów szesnastobitowych zaczynają dominować w zastosowaniach w krajach wysoko rozwiniętych.

W Polsce obserwuje się naturalne zainteresowanie mikroprocesorami szesnastobitowymi, trudno jednak mówić o ich zastosowaniach. W chwili obecnej zaledwie kilka ośrodków prowadzi prace konstrukcyjne nad mikrokomputerami szesnastobitowymi. Taki stan rzeczy jest wynikiem przede wszystkim braku odpowiednich układów scalonych. Należy jednak przewidywać, iż w bliższej lub dalszej przyszłości układy te staną się w Polsce dostępne i możliwe będzie wówczas ich powszechne stosowanie. O szybkości wdrażania techniki mikroprocesorów szesnastobitowych decydować będzie jednak wtedy wykształcenie specjalistów. Świadomość tego faktu stała u podstaw zorganizowania przez Sekcję Systemów Mikrokomputerowych Polskiego Towarzystwa Informatycznego oraz Centrum Szkolenia Informatycznego ZETO Łódź niniejszej Szkoły.

Szkoła ta adresowana jest do osób zajmujących się profesjonalnie mikrokomputerami i znających problematykę mikrokomputerów osmiobitowych. Dlatego w programie Szkoły eksponowane są przede wszystkim różnice, rozszerzenia i nowości mikroprocesorów szesnastobitowych w stosunku do mikroprocesorów osmiobitowych, w szczególności aspekt wielomikroprocesorowości.

Szkoła ma charakter monotematyczny i poświęcona jest architekturze systemów wielomikroprocesorowych zbudowanych z układów rodziny mikroprocesorowej Intel 8086. Wybór tej właśnie rodziny mikroprocesorowej wynika z przewidywanego dominowania jej na polskim rynku mikroprocesorów szesnastobitowych. Natomiast wybór takiej tematyki Szkoły wynika z podstawowego charakteru problematyki architektury systemów.

Na program Szkoły składa się trzynaście wykładów opracowanych specjalnie na zaproszenie Organizatorów. W programie można wyróżnić dwie części. W pierwszej, obejmującej dziewięć pierwszych wykładów omawiana jest architektura systemów wielomikroprocesorowych, mikroprocesory i układy towarzyszące rodziny mikroprocesorowej Intel 8086 oraz standardy magistral. W drugiej części przedstawione są prowadzone obecnie na terenie kraju oryginalne prace konstrukcyjne z zakresu mikrokomputerów szesnastobitowych.

Organizatorzy Szkoły mają nadzieję, że przyczyni się ona do podniesienia poziomu wiedzy w Polsce o mikroprocesorach szesnastobitowych i tym samym przyspieszenia w przyszłości wdrażania techniki wielomikroprocesorowej w naszym kraju. Jednocześnie Organizatorzy mają nadzieję, że Szkoła ta będzie wstępem do szeroko zakrojonego szkolenia z tej problematyki.

Poznań, październik 1984

Wojciech Cellary

VI Szkoła Mikroprocesorowa CSI/PTI
Łódź, grudzień 1984.

ARCHITEKTURA SYSTEMÓW
WIELOMIKROPROCESOROWYCH

doc.dr hab. inż. Wojciech Cellary
Instytut Automatyki
Politechnika Poznańska
ul. Piotrowo 3A
60-965 Poznań, tel. 782-370

1. WSTĘP

Jednym z najważniejszych sposobów podwyższania efektywności działania systemów komputerowych jest zwiększanie stopnia równoległości wykonywanych przez nie operacji. Można to osiągnąć między innymi konstruując systemy komputerowe, w skład których wchodzi wiele procesorów centralnych, działających równoległe i komunikujących się między sobą. Procesory te nie muszą być przy tym identyczne.

Architektura wieloprocessorowa systemów komputerowych ma, oprócz wspomnianej możliwości uzyskania wzrostu efektywności systemu, jeszcze dwie dalsze potencjalne zalety. Pierwszą z nich jest możliwość podwyższenia niezawodności systemu. Uzyskuje się ją na przykład na drodze równoległej realizacji tych samych funkcji systemu przez kilka procesorów z okresowym porównywaniem wyników, na drodze testowania i diagnostyki jednych procesorów /ściślej biorąc całych modułów/ przez drugie, na drodze przejmowania funkcji uszkodzonych procesorów przez inne sprawne, itp.

Drugą potencjalną zaletą jest możliwość modularyzacji systemu zgodnie z wymaganiem danego zastosowania i uzyskanie dzięki temu dużej elastyczności systemu, co ma szczególne znaczenie w przypadku jego modyfikacji i rozwoju.

Zalety wieloprocesorowej architektury systemów komputerowych dostrzeżone zostały już na dość wczesnym etapie rozwoju tych systemów. Począwszy od początku lat siedemdziesiątych rozpoczęto budowę eksperymentalnych systemów o takiej architekturze. Pierwszymi były systemy C.mmp /computer. multi-mini-processor/, którego budowę rozpoczęto w 1971 roku oraz pluribus, którego budowę rozpoczęto w 1972 roku. W systemach tych wykorzystywano technologię, a nawet gotowe moduły minikomputerowe.

Jednakże dopiero rozwój mikroprocesorów, a ściślej rzecz biorąc, rozwój mikroprocesorów trzeciej generacji - szesnastobitowych stworzył techniczne i ekonomiczne warunki do szerokiego stosowania systemów o architekturze wieloprocesorowej. Mówimy w tym przypadku o systemach wielomikroprocesorowych.

Dla uściślenia pojęć dodajmy, że mówiąc w tym referacie o systemach wielomikroprocesorowych mamy na myśli systemy wielomikroprocesorowe silnie powiązane /ang. tightly /closely/ coupled microcomputer systems/, to jest takie, w których komunikacja między mikroprocesorami odbywa się przez wspólną pamięć operacyjną. W przeciwieństwie do systemów silnie powiązanych, istnieją jeszcze systemy luźno powiązane /ang. loosely coupled microprocessor systems/, w których komunikacja między mikroprocesorami odbywa się poprzez porty wejścia/wyjścia.

Celem tego referatu jest przedstawienie architektur systemów wielomikroprocesorowych w podziale na systemy z dzieloną magistralą lokalną i systemy z dzieloną magistralą systemową oraz wyjaśnienie zasad działania takich systemów.

2. SYSTEMY WIELOMIKROPROCESOROWE Z DZIELONĄ MAGISTRALĄ LOKALNĄ

W tym rozdziale przedstawimy systemy wielomikroprocesorowe z dzieloną magistralą lokalną. Dokładną klasyfikację magistral w systemie wielomikroprocesorowym przeprowadzimy w rozdziale 3. Tutaj wyjaśnimy jedynie, że magistrala lokalna utworzona jest przez linie bezpośrednio dochodzące do nóżek mikroprocesora.

Systemy wielomikroprocesorowe z dzieloną magistralą lokalną są architektonicznie bliskie systemom jednomikroprocesorowym z kanałem bezpośredniego dostępu DMA /ang. Direct Memory Access/. Przypomnijmy zasadę współpracy mikroprocesora centralnego z kanałem DMA

w celu ukazania następnie różnic w stosunku do systemów wielomikroprocesorowych z dzieloną magistralą lokalną.

Z teoretycznego punktu widzenia system jednoprocessorowy wyposażony w kanał DMA może być uznany za ograniczony system wielomikroprocesorowy. Kanał DMA można bowiem uznać za specjalizowany mikroprocesor, z tego względu, że może on samodzielnie ubiegać się o dostęp do magistrali. Rozważmy jednak problem równoległości działania mikroprocesora i kanału DMA w takim systemie. Jak wiadomo, uaktywnienie kanału DMA powoduje zawieszenie działania mikroprocesora; praca jednego układu wyklucza więc pracę drugiego - równoległość ich działania na poziomie cykli maszynowych jest zerowa. Taka cecha współpracy jest sprzeczna z podstawowym celem budowy systemów wielomikroprocesorowych jakim jest zwiększenie efektywności systemu przez równoległą realizację jego funkcji.

Dla ścisłości należy stwierdzić, że jeśli transmisja przez kanał DMA nie jest na tyle szybka aby wykorzystywać wszystkie kolejne cykle maszynowe magistrali, to kanał DMA zwalnia magistralę, umożliwiając wznowienie pracy mikroprocesorowi na przykład na jeden cykl. W horyzoncie całej transmisji możliwa jest zatem pewna równoległość pracy mikroprocesora i kanału DMA. Powyższe wyjaśnienia nie zmieniają jednak faktu, że gdyby na podobnej zasadzie połączyć w systemie dwa mikroprocesory drugiej generacji, to stopień równoległości ich pracy byłby minimalny. Z tego względu mikroprocesory trzeciej generacji, których przedstawicielem jest Intel 8086, wyposażone są w mechanizmy umożliwiające znaczne podniesienie tego stopnia.

Istotną cechą mikroprocesora Intel 8086 jest to, że wyposażony jest on w wewnętrzną kolejkę, do której pobiera rozkazy z pamięci operacyjnej wyprzedzając w stosunku do potrzeb. Dzięki temu, w ogólności, w sytuacji, w której mikroprocesor nie ma dostępu do magistrali ze względu na jej przydział innemu mikroprocesorowi w systemie wielomikroprocesorowym, praca jego nie jest zatrzymywana, gdyż rozkazy pobierane są z wewnętrznej kolejki. Oczywiście nie są wykluczone sytuacje, w których mikroprocesor, żądając pobrania rozkazu, natrafia na pustą kolejkę wewnętrzną oraz niedostępną magistralę i musi czekać. Sytuacja taka może pojawić się na przykład po wykonaniu rozkazu skoku, gdy kolejka wewnętrzna musi być wyzerowana.

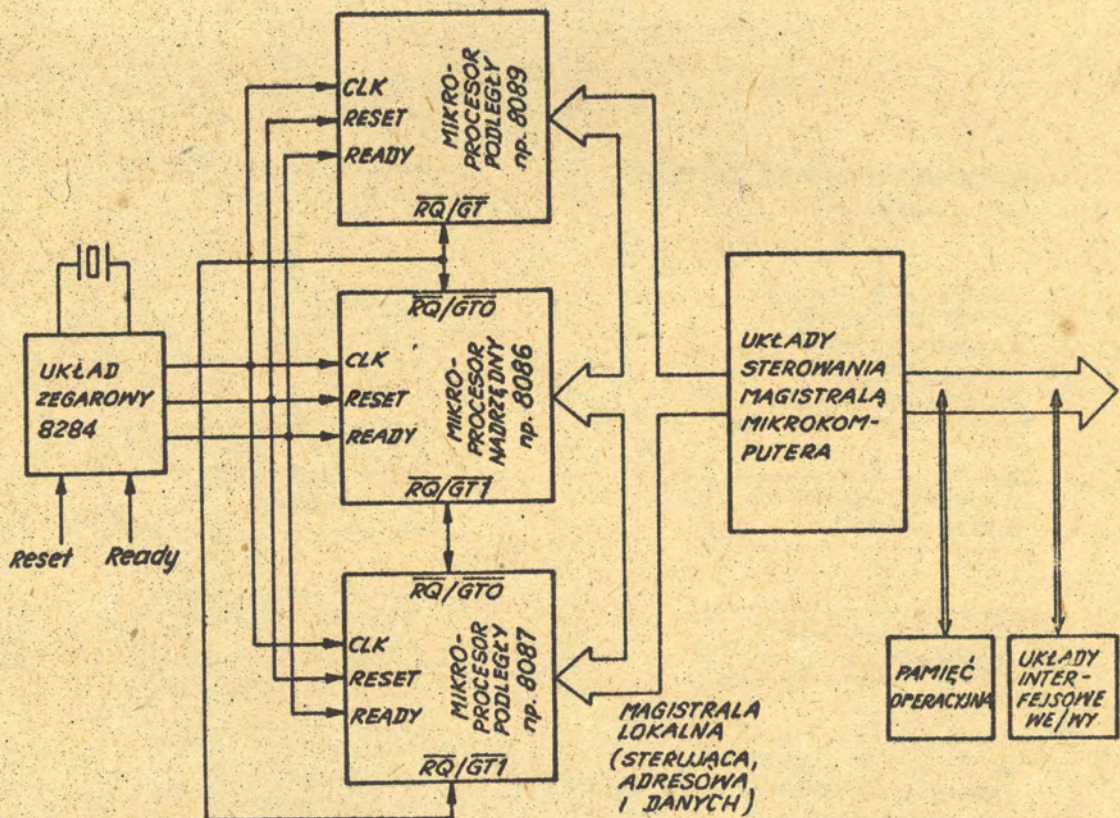
Cykle dostępu do magistrali są na ogół krótsze niż niezależne

od nich cykle wykonywania rozkazów. Z tego względu stosunkowo często zdarza się, że wewnętrzna kolejka mikroprocesorowa jest pełna. Mikroprocesor odłącza się wówczas od magistrali zwalniając ją dla innych mikroprocesorów. Ponowne dołączenie się mikroprocesora do magistrali może nastąpić w dowolnym cyklu zegarowym, bez potrzeby dodatkowej synchronizacji.

Reasumując, należy stwierdzić, że możliwości buforowania pobieranych wyprzedzająco rozkazów bardzo istotnie redukuje wpływ oczekiwania na dostęp do magistrali na efektywność pracy mikroprocesora, umożliwiając tym samym budowę efektywnych systemów wielomikroprocesorowych.

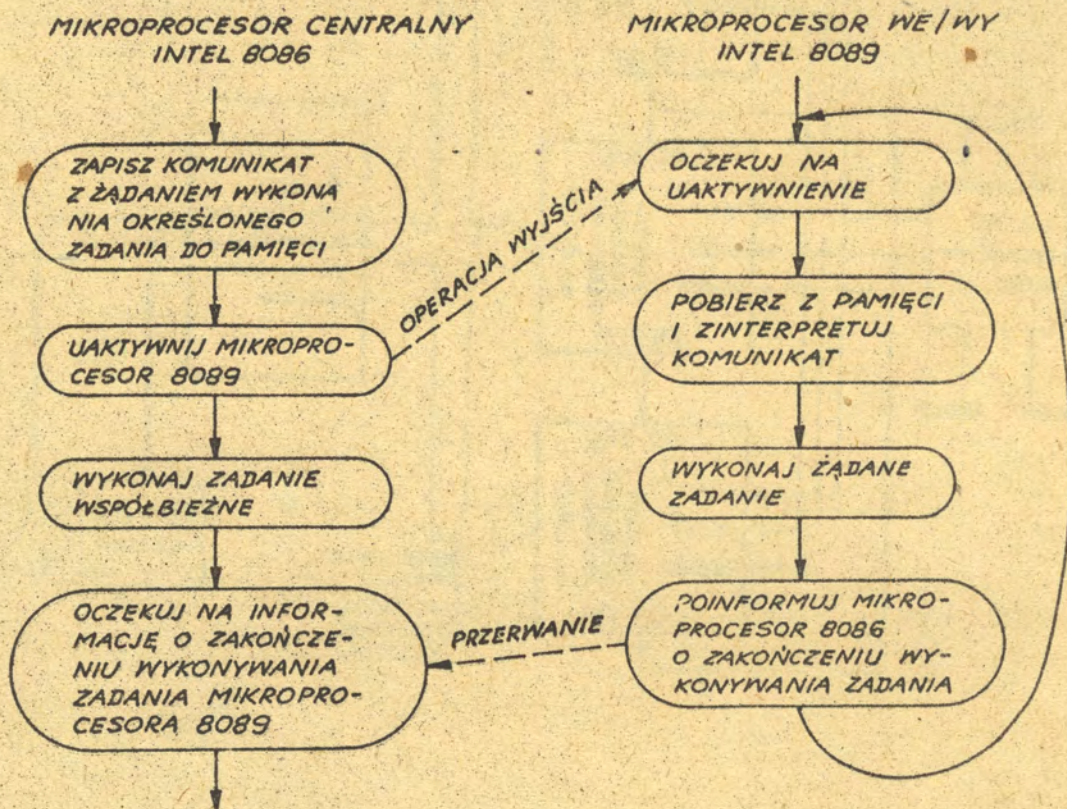
Przedstawimy obecnie zasadę działania systemów wielomikroprocesorowych z dzieloną magistralą lokalną. Możliwość budowy takich systemów wynika z wbudowania do mikroprocesora Intel 8086 i innych mikroprocesorów jego rodziny pewnego systemu arbitrażu tej magistrali. Umożliwia on współpracę trzech mikroprocesorów. Jeden z nich pełni rolę nadrzędną i nadzoruje dostęp do magistrali lokalnej dwóch pozostałych, z których jeden ma wyższy priorytet niż drugi. Wymiana sygnałów żądania przydziału magistrali lokalnej, potwierdzenia przyjęcia żądania i zwolnienia magistrali pomiędzy mikroprocesorem nadrzędnym a jednym z mikroprocesorów podległych odbywa się na jednej linii sygnałowej oznaczonej RQ/GT 0 lub 1 /request/grant/. Żądanie przydziału magistrali lokalnej ze strony mikroprocesora podległego zgłaszane jest do mikroprocesora nadrzędnego przez podanie impulsu na linii RQ/GT. Po otrzymaniu tego żądania mikroprocesor nadrzędny, po zakończeniu wykonywania bieżącego cyklu maszynowego, wysyła tą samą linią impuls potwierdzenia przyjęcia żądania i odłącza się od magistrali. Oczywiście może on kontynuować swoje działanie pobierając rozkazy ze swojej wewnętrznej kolejki rozkazów. Mikroprocesor podległy po zrealizowaniu dostępu do magistrali odłącza się od niej i informuje o tym fakcie mikroprocesor nadrzędny przesyłając impuls linią RQ/GT.

Architektura systemu wielomikroprocesorowego z dzieloną magistralą lokalną przedstawiona jest na rysunku 1. Warto zauważyć, że ten typ organizacji systemu wielomikroprocesorowego wymaga synchronizacji wszystkich mikroprocesorów. Stąd w systemie znajduje się jeden wspólny układ zegarowy 8284.



Rys. 1. Architektura systemu wielomikroprocesorowego z dzieloną magistralą lokalną.

Istotną zaletą systemów wielomikroprocesorowych z dzieloną magistralą lokalną jest to, że nie wymagają one stosowania specjalnych dodatkowych układów towarzyszących. Odpowiednie mechanizmy wbudowane są w samych mikroprocesorach, choć należy zwrócić uwagę, że mechanizmy te nie są wystarczające do połączenia w systemie kilku mikroprocesorów centralnych Intel 8086. Głównym przeznaczeniem omawianej architektury jest łączenie mikroprocesorów specjalizowanych. Przykładową konfiguracją tego typu jest połączenie mikroprocesora centralnego Intel 8086 /lub Intel 8088/ ze specjalizowanym mikroprocesorem we/wy Intel 8089. Ten ostatni łączy w sobie cechy mikroprocesorów centralnych, a więc zdolność do wykonywania programów, z cechami kanałów DMA, a więc ze zdolnością do szybkiej, ciągłej transmisji danych. Synchronizacja na poziomie zadań wykonywanych przez dwa mikroprocesory przedstawiona jest na rysunku 2.



Rys. 2. Synchronizacja mikroprocesorów: centralnego i we/wy na poziomie zadań.

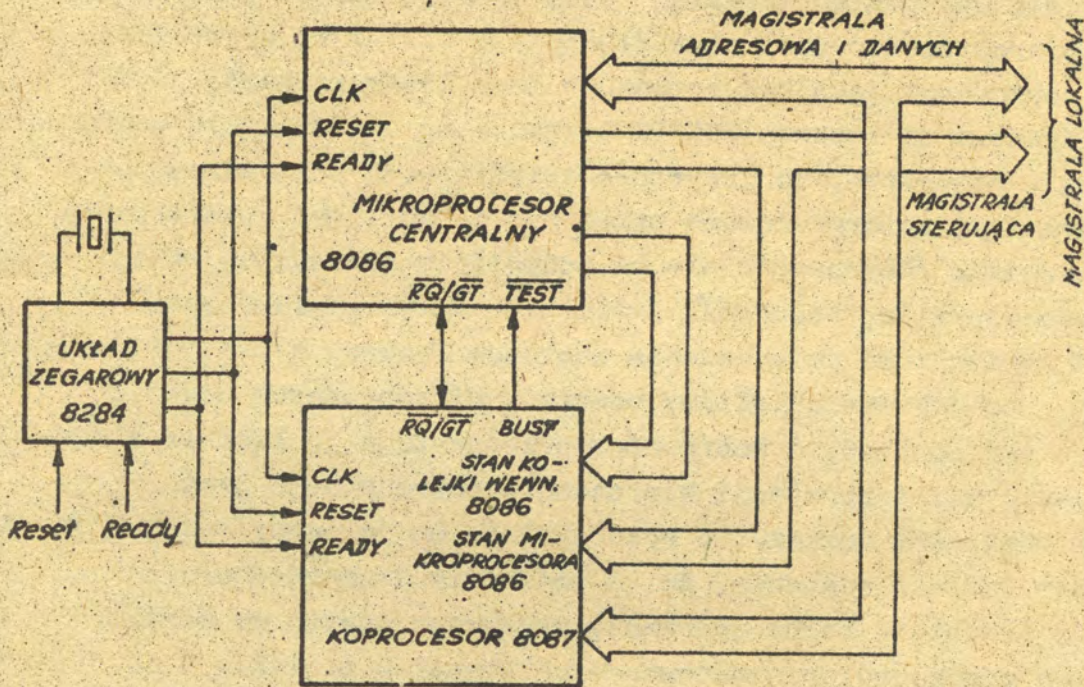
Jak widać, synchronizacja ta odbywa się z wykorzystaniem operacji we/wy, wspólnej pamięci operacyjnej i przerwań.

Inną typową konfiguracją systemu wielomikroprocesorowego z dzieloną magistralą lokalną jest połączenie mikroprocesora centralnego Intel 8086 /lub Intel 8088/ z koprocesorem, np. koprocesorem arytmetycznym Intel 8087.

Koprocesor jest nowym typem mikroprocesora, który nie jest mikroprocesorem centralnym. Nie wykonuje on własnego programu i z tego względu zawsze towarzyszy pewnemu mikroprocesorowi centralnemu. Jego zadaniem jest rozszerzenie listy rozkazów mikroprocesora centralnego w danym kierunku i wykonywanie tych uzupełniających rozkazów równoległe z programem mikroprocesora centralnego. Przykładowo, koprocesor Intel 8087 rozszerza listę

rozkazów mikroprocesora Intel 8086 o złożone operacje arytmetyczne, w szczególności zmiennoprzecinkowe.

Architekturę systemu wielomikroprocesorowego z dzieloną magistralą lokalną w konfiguracji mikroprocesor centralny - koprocesor przedstawiono na rysunku 3.



Rys. 3. System wielomikroprocesorowy z mikroprocesorem centralnym i koprocesorem.

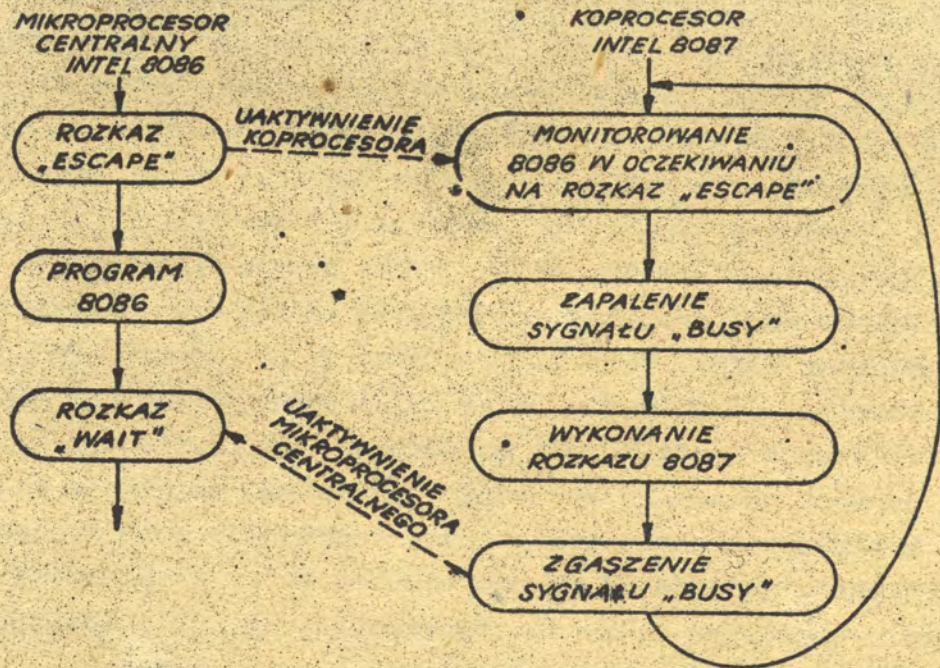
Współpraca koprocesora z mikroprocesorem centralnym polega na monitorowaniu rozkazów pobieranych przez mikroprocesor centralny Intel 8086 w oczekiwaniu na rozkazy przeznaczone dla niego. Rozkazy te to 64 rozkazy ESCAPE. Koprocesor rozpoznaje rozkazy ESCAPE /w szczególności odróżnia je od danych/ na podstawie badania stanu mikroprocesora centralnego. Ponadto koprocesor bada stan kolejki wewnętrznej mikroprocesora centralnego. Zauważmy bowiem, że z faktu pobrania rozkazu przez mikroprocesor centralny nie wynika jeszcze fakt jego wykonania.

Mikroprocesor centralny ignoruje rozkazy ESCAPE. Wykonanie takiego rozkazu odpowiada wykonaniu klasycznego rozkazu "nie nie

rób", z tą różnicą, że jeśli tryb adresowania rozkazu ESCAPE wskazuje na istnienie operandu w pamięci operacyjnej, to operand ten jest pobierany.

Koprocesor posiada analogiczną kolejkę wewnętrzną jak mikroprocesor centralny i równoległe do mikroprocesora centralnego pobiera do niej te same rozkazy. W przypadku stwierdzenia na podstawie kodu rozkazu oraz stanu kolejki wewnętrznej mikroprocesora centralnego wykonanie przez niego rozkazu ESCAPE, rozpoczyna jego równoległe wykonanie. Określone pola rozkazu ESCAPE ignorowane przez mikroprocesor centralny /łącznie 5 bitów/ są traktowane przez koprocesor jako kod rozkazu z jego listy rozkazów. Jeśli tryb adresowania rozkazu ESCAPE wskazuje na umieszczenie operandu rozkazu w rejestrze, to zawartość trzybitowego pola adresowania rejestru jest interpretowana przez koprocesor jako rozszerzenie kodu rozkazu. /Koprocesor nie ma oczywiście dostępu do rejestrów mikroprocesora centralnego/. Jeśli tryb adresowania rozkazu ESCAPE wskazuje na umieszczenie operandu rozkazu w pamięci operacyjnej, to, jak wspomnieliśmy powyżej, mikroprocesor centralny pobiera ten operand, a koprocesor przechwytuje go łącznie z adresem. Jeśli w interpretacji koprocesora pobrana dana jest rzeczywiście operandem rozkazu to wykorzystuje on ją bezpośrednio. Jeśli jest częścią operandu, to na podstawie zapamiętanego adresu pobiera on dalsze bajty operandu, uzyskując dostęp do magistrali lokalnej przez wysłanie odpowiednich sygnałów na linię "Request/Grant". Wreszcie, jeśli pobrana przez mikroprocesor centralny dana ma być w interpretacji koprocesora wynikiem, to koprocesor ignoruje ją, zapamiętując jedynie jej adres, pod który pošle wynik wykonanego rozkazu, uzyskując dostęp do magistrali lokalnej jak poprzednio.

Przebieg synchronizacji pomiędzy mikroprocesorem centralnym a koprocesorem na poziomie rozkazów, który odpowiada poziomowi zadań w przypadku mikroprocesora centralnego i mikroprocesora we/wy, przedstawiony jest na rysunku 4.



Rys. 4. Synchronizacja mikroprocesora centralnego i koprocatora na poziomie rozkazów.

Synchronizacja w kierunku od mikroprocesora centralnego do koprocatora została powyżej omówiona. Pozostaje więc do omówienia synchronizacja w drugą stronę. Jej potrzeba zachodzi wówczas gdy w momencie, w którym mikroprocesor centralny musi skorzystać z wyników równoległe wykonywanej przez koprocator operacji nie ma pewności, czy operacja ta już się zakończyła.

Do synchronizacji tej służy rozkaz WAIT wykonywany przez mikroprocesor centralny. Działanie tego rozkazu jest uzależnione od stanu wejścia TEST mikroprocesora, na które podaje się normalnie sygnał wyjściowy BUSY koprocatora. Sygnał ten ma wartość logiczną 1 od momentu zdekodowania rozkazu ESCAPE przez koprocator do momentu zakończenia wykonywania rozkazu. Jeśli sygnał TEST przyjmuje wartość 0 to działanie rozkazu WAIT jest równoważne działaniu rozkazu "nic nie rób". Natomiast jeśli sygnał TEST przyjmuje wartość 1 to rozkaz WAIT powoduje zawieszenie działania mikroprocesora centralnego aż do zmiany stanu tego sygnału.

Powracając do problemów architektury systemów wielomikroprocesorowych z dzieloną magistralą lokalną należy stwierdzić, że korzystną i zalecaną konfiguracją jest współdzielenie magistrali lokalnej przez trzy mikroprocesory: centralny 8086, we/wy 8089

i koprocesor 8087. /por. rysunek 1/. Wówczas, ze względu na to, że mikroprocesor we/wy prowadzi szybkie transmisje danych uwarunkowane czasowo, nadaje mu się wyższy priorytet dołączając do linii RQ/GT 0 mikroprocesora centralnego. W przypadku jednoczesnego zgłoszenia żądania przydziału magistrali lokalnej przez mikroprocesor we/wy i koprocesor spełnione zostanie w pierwszej kolejności żądanie mikroprocesora we/wy. Jeśli jednak magistrala lokalna jest przydzielona do koprocesora w momencie zgłoszenia żądania przez mikroprocesor we/wy, to musi on czekać na zwolnienie jej. Aby minimalizować to oczekiwanie linię RQ/GT mikroprocesora we/wy łączy się dodatkowo z linią RQ/GT1 koprocesora /por. rysunek 1/. Wówczas, na mocy opisanego uprzednio mechanizmu działania sygnałów RQ/GT w mikroprocesorze centralnym, koprocesor zwalnia magistralę po otrzymaniu żądania od mikroprocesora we/wy natychmiast po zakończeniu bieżącego cyklu maszynowego, nawet jeśli potrzebuje dalszych cykli do dokończenia wykonywania swego rozkazu.

Inną zalecaną konfiguracją mikroprocesorów w omawianej architekturze jest połączenie dwóch mikroprocesorów we/wy w celu zwiększenia liczby kanałów transmisyjnych. W tym przypadku jeden z mikroprocesorów programuje się jako nadrzędny, a drugi podległy. Pozostałe mechanizmy ich wzajemnej współpracy pozostają bez zmian.

3. SYSTEMY WIELOMIKROPROCESOROWE Z DZIELONĄ MAGISTRALĄ SYSTEMOWĄ

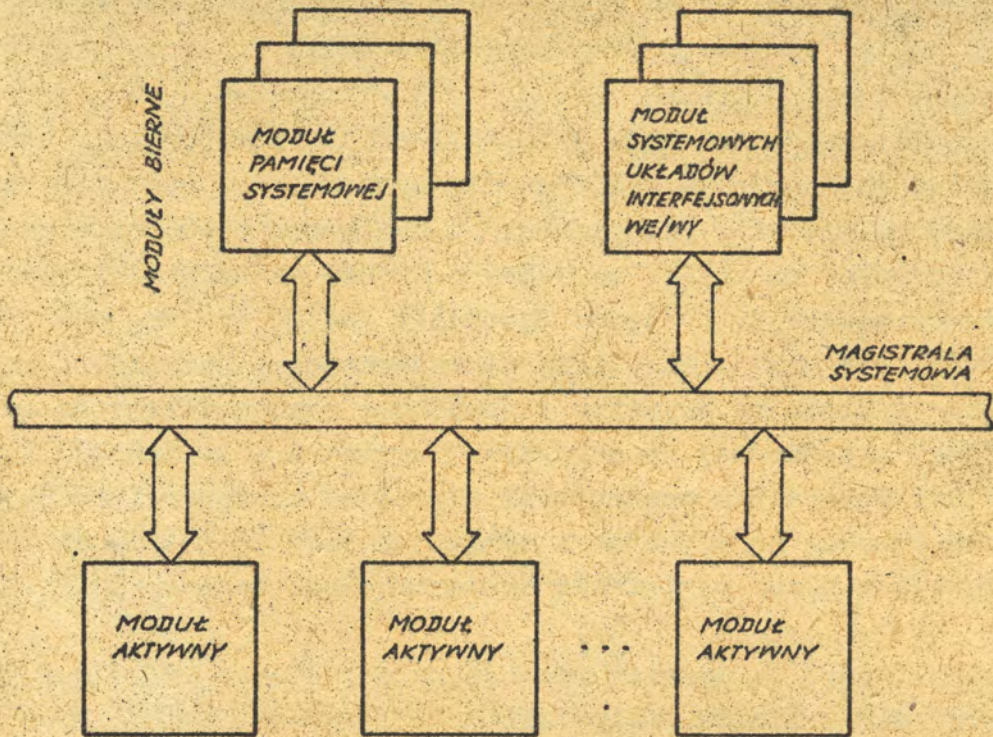
Podstawową zaletą przedstawionej w poprzednim rozdziale architektury systemów wielomikroprocesorowych z dzieloną magistralą lokalną była możliwość budowy takich systemów bez specjalnych układów towarzyszących, przy wykorzystaniu wbudowanych w mikroprocesory mechanizmów. Architekturę tę cechują jednak pewne istotne ograniczenia. Przede wszystkim jest ona przewidziana do łączenia mikroprocesorów specjalizowanych. Przy wykorzystaniu wyłącznie mechanizmów wbudowanych w mikroprocesory rodziny Intel 8086 nie można połączyć w takim systemie kilku mikroprocesorów centralnych. Oczywiście, dobudowując odpowiednią logikę na zewnątrz mikroprocesorów można by skonstruować system wielomikroprocesorowy z dzieloną magistralą lokalną, w którym połączonych byłoby kilka mikroprocesorów central-

nych. System taki byłby jednak mało efektywny z tego względu, że każdy dostęp do sprzętowych zasobów systemu pamięci operacyjnej i układów interfejsowych we/wy, które są wspólne, byłby potencjalnie konfliktowy. Droga do zminimalizowania liczby konfliktów i tym samym do podniesienia efektywności systemu jest rozdział zasobów systemu na prywatne /dedykowane/ i systemowe /współdzielone/. Zasoby prywatne to takie, do których dostęp ma wyłącznie jeden mikroprocesor centralny systemu. Natomiast zasoby systemowe to takie, do których dostęp mają wszystkie mikroprocesory. Oczywiście rozdział zasobów systemu na prywatne i systemowe wymaga rozdzielenia magistrali lokalnej odpowiednio na magistralę prywatną i systemową. Rozdział ten jest dokonany w przestrzeni adresowej mikroprocesorów. Po rozdzieleniu zasobów na prywatne i systemowe jedynie dostęp do zasobów systemowych przez magistralę systemową jest potencjalnie konfliktowy.

Zauważmy, że podział zasobów na prywatne i systemowe jest naturalny dla systemów wielomikroprocesorowych. Poszczególne mikroprocesory wykonują bowiem zazwyczaj autonomiczne, równoległe zadania wykorzystując swoje zasoby prywatne i stosunkowo rzadko komunikują się między sobą, co wymaga wykorzystania zasobów systemowych.

Systemy wielomikroprocesorowe o rozdzielonej w przestrzeni adresowej magistrali lokalnej w sposób opisany powyżej nazywamy systemami z dzieloną /w czasie/ magistralą systemową. Ogólną architekturę takiego systemu przedstawiono na rysunku 5. W systemie takim wyróżnia się moduły aktywne oraz bierne dołączone do magistrali systemowej. Moduły aktywne w przeciwieństwie do modułów biernych, są to takie moduły, które mogą samodzielnie ubiegać się o dostęp do magistrali systemowej. Należy podkreślić, że wyróżnikiem modułów aktywnych jest ich zdolność do ubiegania się o dostęp do magistrali systemowej, a nie fakt wyposażenia w mikroprocesor, który składnikiem jest konieczny do uzyskania tej zdolności. Moduły bierne mogą bowiem również być wyposażone w mikroprocesory, co wcale nie przesądza o możliwości ich samodzielnego dostępu do magistrali.

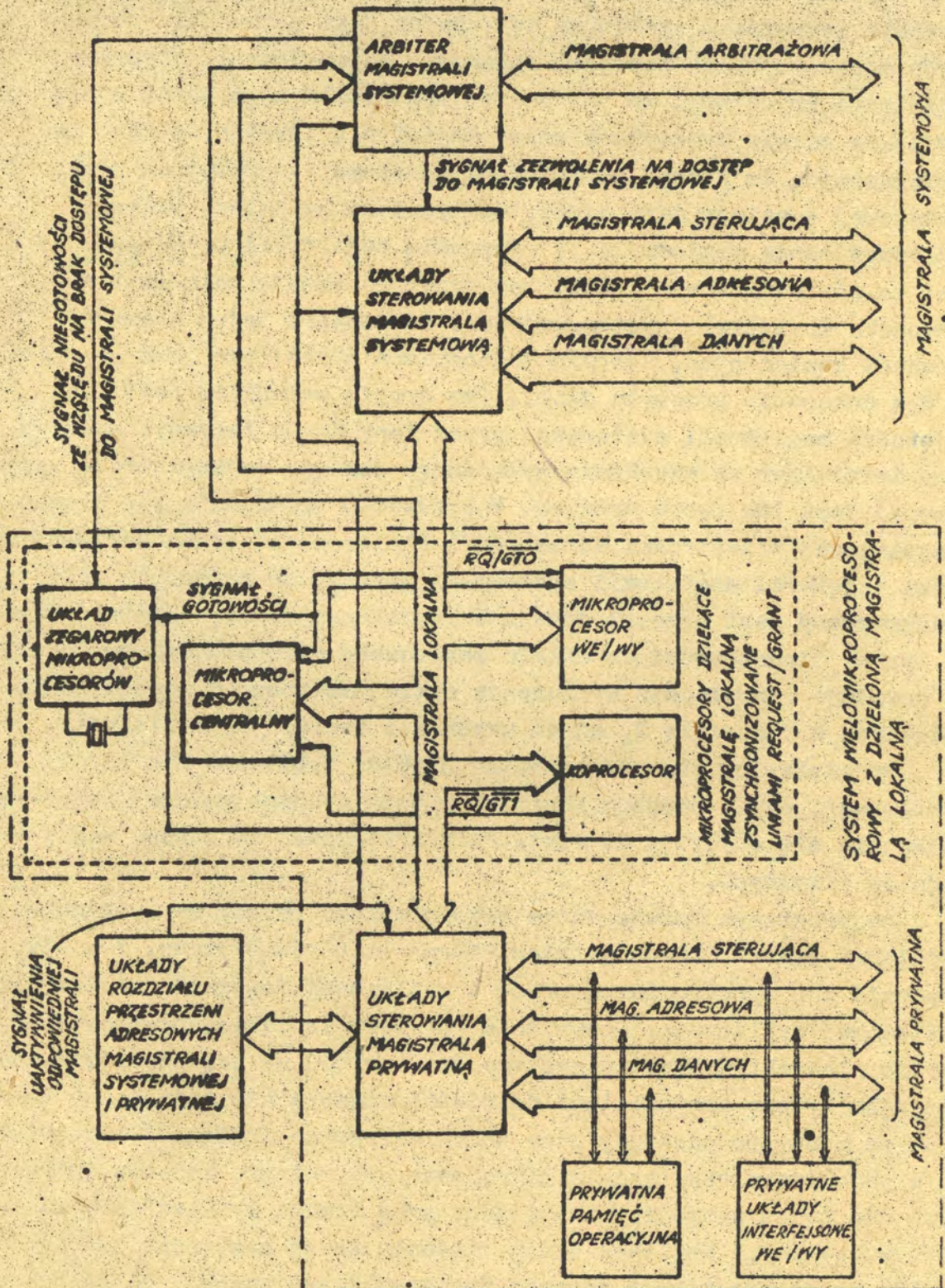
Architektura logiczna systemu wielomikroprocesorowego z dzieloną magistralą systemową przedstawiona na rysunku 5 odpowiada zazwyczaj architekturze fizycznej takiego systemu. Oznacza to, że każdy moduł systemu zawarty jest zazwyczaj na jednej płycie.



Rys. 5. Architektura systemu wielomikroprocesorowego z dzieloną magistralą systemową

Na rysunku 6 przedstawiono architekturę modułu aktywnego. W module tym, w jego konfiguracji maksymalnej, można wyróżnić część będącą systemem wielomikroprocesorowym z dzieloną magistralą lokalną, z trzema specjalizowanymi mikroprocesorami i wspólnym układem zegarowym oraz zasobami prywatnymi - por. rysunek 1. W konfiguracji minimalnej modułu część ta redukuje się do pojedynczego mikroprocesora centralnego lub we/wy, bez dodatkowych mikroprocesorów, zasobów prywatnych, układów sterowania magistralą prywatną i w konsekwencji układów rozdziału przestrzeni adresowej magistrali systemowej i prywatnej. Oczywiście możliwe są konfiguracje pośrednie.

W stosunku do systemu z dzieloną magistralą lokalną moduł aktywny zawiera pewne układy dodatkowe. Przede wszystkim układy sterowania magistralą systemową. Z dokładnością do szczegółów układy te nie różnią się od odpowiednich układów sterowania magis-



Rys. 6. Architektura wewnętrzna modułu aktywnego

trala prywatną.

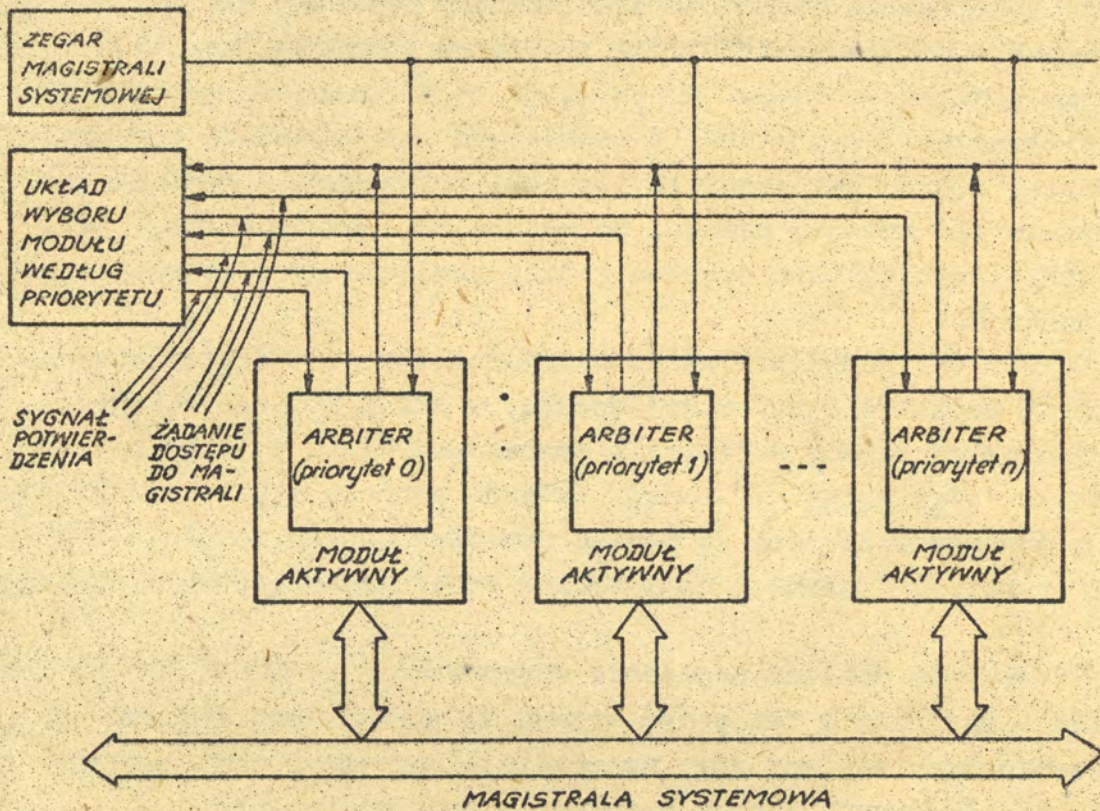
Rozdział przestrzeni adresowej mikroprocesora pomiędzy magistralą prywatną i systemową dokonywany jest przez układ dekodów adresów. Układ ten jest dołączony do magistrali adresowej magistrali prywatnej, na tej bowiem magistrali mogą pojawić się wszystkie adresy przesyłane przez magistralę lokalną modułu. Po stwierdzeniu, do której przestrzeni adresowej - prywatnej czy systemowej - należy dany adres, uaktywniane są układy sterowania odpowiednią magistralą. W przypadku magistrali prywatnej jest to równoznaczne z dokonaniem przesyłu danej. Natomiast w przypadku magistrali systemowej jest to dopiero jeden z dwóch warunków koniecznych, których jednoczesne spełnienie jest wymagane dla dokonania przesyłu danej. Tym drugim warunkiem jest brak zajętości magistrali systemowej przez inny moduł aktywny. Układem odpowiedzialnym za uzyskanie wyłącznego dostępu do magistrali systemowej jest tak zwany arbiter. W przypadku pojawienia się adresu z systemowej przestrzeni adresowej i stwierdzenia przez arbiter braku zajętości magistrali systemowej zezwala on na dostęp do niej i dokonywany jest przesył danej. W przeciwnym razie, arbiter wysyła sygnał niegotowości do układu zegarowego mikroprocesorów modułu, co powoduje odpowiednie wydłużenie cyklu maszynowego według zasad opisanych w rozdziale 2, aż do uzyskania dostępu do magistrali.

Przedstawimy obecnie pokrótce problem arbitrażu magistrali systemowej, czyli problem konfliktów powstających podczas współubiegania się o dostęp do magistrali systemowej ze strony wielu modułów aktywnych.

Najprostszym rozwiązaniem tego problemu byłoby wprowadzenie wspólnego zegara dla wszystkich mikroprocesorów systemu, tak aby zapewnić ich synchroniczną pracę i realizować dostęp do magistrali systemowej cyklicznie. Oczywiście rozwiązanie takie spowodowałoby bardzo nieefektywną pracę systemu, ze względu na to, że często magistrala byłaby przydzielona do mikroprocesora nie żądającego dostępu do niej, podczas gdy inne oczekiwałyby na dostęp. Z tego względu w stosowanym rozwiązaniu mikroprocesory różnych modułów aktywnych nie są zsynchronizowane między sobą i mogą wysyłać żądania dostępu w dowolnych chwilach czasu. Żądania te są następnie zsynchronizowane przez arbitery z zegarem magistrali systemowej. Po zsynchronizowaniu możliwe jest określenie, który z ubiegających się o dos-

tep do magistrali systemowej modułów aktywnych uzyska go. Wybór modułu odbywa się według algorytmu priorytetowego.

Miejsce arbiterów w architekturze systemu wielomikroprocesorowego z dzieloną magistralą systemową przedstawiono na rysunku 7.



Rys. 7. Układ arbitrażu magistrali systemowej.

Mówiąc o problemie arbitrażu należy wspomnieć o pewnej koniecznej własności arbitra magistrali systemowej, a mianowicie zdolności do zapewnienia w niektórych przypadkach ciągłego dostępu do tej magistrali w czasie pełnego cyklu rozkazowego złożonego z kilku cykli maszynowych. Konieczność zagwarantowania takiego dostępu występuje na przykład przy realizacji operacji działających na semaforach służących do synchronizacji zadań na poziomie systemu operacyjnego. Gdyby nie było możliwości nieprzerwanego dostępu do magistrali systemowej w ciągu pełnego rozkazu, to mogłoby dojść do sytuacji, w której rozkazy testowania wartości semafora i jego uaktywniania wykonywane przez dwa różne mikroprocesory prze-

plotłyby się na poziomie cykli maszynowych powodując błędną interpretację wartości semafora. W konsekwencji, niemożliwa byłaby synchronizacja zadań w systemie operacyjnym systemu wielomikroprocesorowego z dzieloną magistralą systemową.

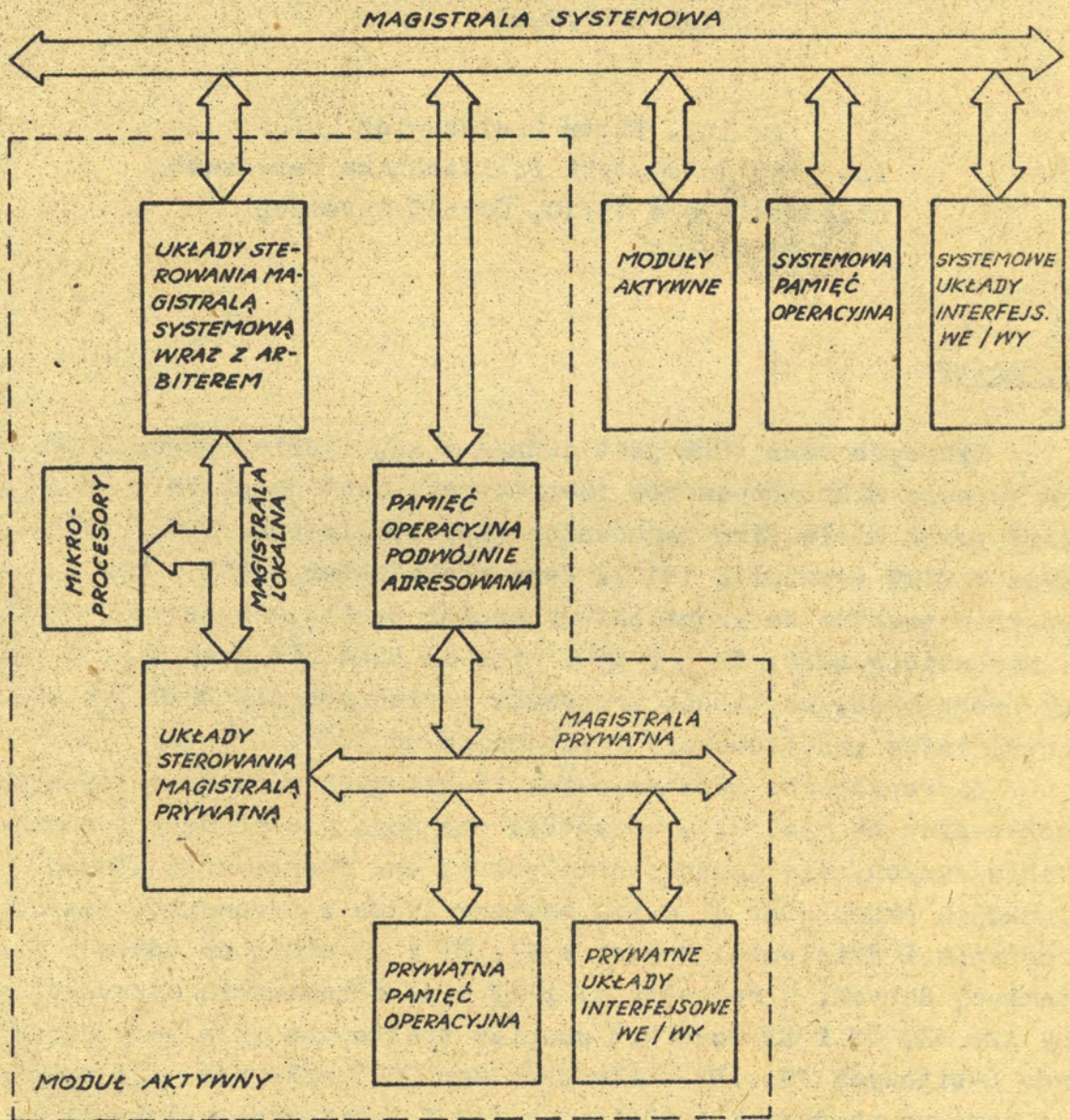
Na zakończenie omawiania architektury systemów wielomikroprocesorowych z dzieloną magistralą systemową powróćmy jeszcze do problemu podziału zasobów. Na początku tego rozdziału wyróżniliśmy zasoby prywatne i systemowe. W niektórych rozwiązaniach pojawiają się zasoby będące jednocześnie zasobami prywatnymi i systemowymi. Dotyczy to tak zwanych pamięci podwójnie adresowanych /ang. Dual Port RAM/. Architektura systemu z taką pamięcią przedstawiona jest na rysunku 8.

Do pamięci podwójnie adresowanej możliwy jest dostęp zarówno od strony magistrali prywatnej modułu, w skład którego wchodzi, jak również od strony magistrali systemowej. Jest to więc pamięć systemowa, gdyż wszystkie moduły aktywne systemu mają do niej dostęp, a jednocześnie jest to pamięć prywatna danego modułu o możliwości szybkiego dostępu z pominięciem arbitrażu magistrali systemowej.

Sterowanie pamięci podwójnie adresowanych redukuje liczbę konfliktów w systemie i tym samym wpływa na wzrost jego efektywności. Stosowanie tych pamięci jest szczególnie korzystne przy wymianie dużych bloków danych między modułami aktywnymi systemu.

4. UWAGI KOŃCOWE

W referacie tym naszkicowaliśmy główne problemy architektury systemów wielomikroprocesorowych trzeciej generacji. Dzięki wbudowanym odpowiednim mechanizmom w mikroprocesory oraz odpowiednim układom rodziny mikroprocesorowej Intel 8086 możliwe jest konstruowanie szerokiej gamy systemów wielomikroprocesorowych specjalizowanych dla różnych zastosowań. Szerokie upowszechnienie tych systemów, również w Polsce, jest, naszym zdaniem, kwestią przewidywalnej przyszłości.



Rys. 8. System wielomikroprocesorowy z pamięcią operacyjną podwójnie adresowaną.

VI Szkoła Mikroprocesorowa CSI/PTI
Łódź, grudzień 1984

MIKROPROCESOR CENTRALNY
INTEL 8086/8088

dr inż. Piotr Misiurewicz
Instytut Automatyki Politechnika Warszawska
ul. Nowowiejska 15/10, 00-665 Warszawa
tel. 21007/862

1. WSTĘP

Mikroprocesor 8086 jest jednym z najbardziej popularnych na świecie mikroprocesorów 16-bitowych. Jest on obecnie produkowany przez wiele firm zachodnich (Intel, Siemens, NEC, Mostek), oraz w ZSRR (symbol K 1810). Jego architektura jest w znacznym stopniu podobna do architektury swoich 8-bitowych poprzedników a mianowicie 8080, 8085 i Z80. Mimo iż 8086 nie jest z nimi zgodny programowo, większość programów napisanych dla 8080 lub Z80 można łatwo przetłumaczyć i uruchomić na 8086.

Mikroprocesor 8086 ma osiem 16-bitowych rejestrów roboczych, pokazanych na rys. 2.1. Wszystkie one mogą służyć do przechowywania danych, ale jednocześnie pełnią one szereg specjalnych funkcji. Akumulator AX m.in. zawiera jeden z argumentów operacji mnożenia i dzielenia. Rejestry SI, DI i BX służą do adresowania pamięci danych, a rejestry SP i BP - do adresowania stosu. Rejestry AX, BX, CX i DX mogą być również traktowane jako pary rejestrów 8-bitowych AH...DL. Mikroprocesor 8086 wykonuje więc operacje zarówno na danych 16-bitowych jak i 8-bitowych. Ośmiobitowe dane mogą reprezentować np. teksty zapisane jako ciągi znaków alfanumerycznych lub liczby dziesiętne zapisane jako ciągi cyfr.

Rozkazy mikroprocesora 8086 mają długość od 1 do 6 bajtów;



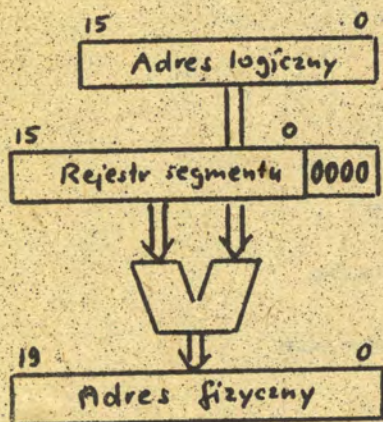
Rys. 2.1. Rejestry mikroprocesora 8086

zasadnicze rozkazy są dwubajtowe. Argumenty większości rozkazów mogą znajdować się w rejestrze roboczym lub w pamięci, przy czym tryby adresowania pamięci są następujące:

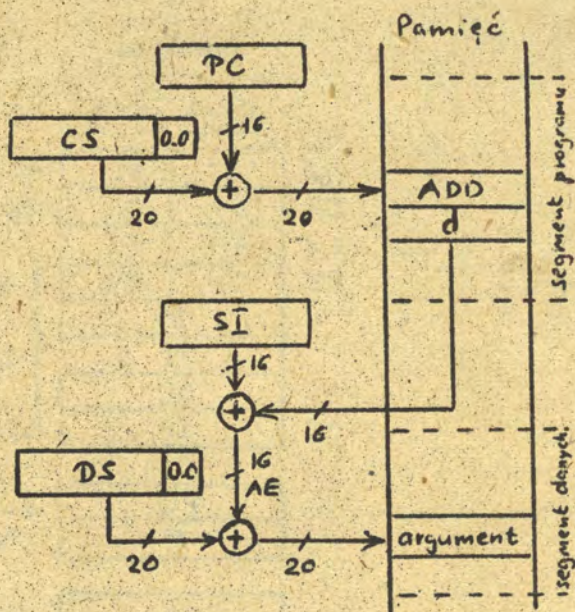
- natychmiastowy
- bezpośredni
- zawartością rejestru indeksowego lub bazowego z ew. dodanym przesunięciem 8 lub 16-bitowym
- indeksowo-bazowe (adres efektywny jest sumą zawartości rejestru indeksowego, bazowego i przesunięcia).

2. ADRESOWANIE PAMIĘCI

Mikroprocesor 8086 operuje na 16-bitowych adresach, gdyż zarówno licznik rozkazów PC, jak i rejestry bazowe oraz indeksowe są szesnastobitowe. Gdyby adresy te były fizycznymi adresami komórek pamięci, można by adresować tylko 64 K bajty pamięci, co w wielu współczesnych systemach mikrokomputerowych jest niewystarczające. Dlatego też 8086 odwzorowuje 16-bitowy adres logiczny w 20-bitowy adres fizyczny, co umożliwi operowanie na pamięci o pojemności 1 Mbajta. Odwzorowanie to jest realizowane przez użycie rejestrów segmentów CS, DS, SS, ES - szesnastobitowe.



Rys. 2.2. Sposób tworzenia adresu fizycznego.



Rys. 2.3. Adresowanie pamięci dla rozkazu ADD [SI+d]. AE - adres efektywny (logiczny)

logiczny jest sumowany z przesuniętą o 4 bity szesnastobitową zawartością odpowiedniego rejestru segmentu, tworząc 20-bitowy adres fizyczny (nie dotyczy to wejścia-wyjścia, gdzie adres logiczny układu we/wy jest od razu adresem fizycznym) - patrz rys. 2.2.

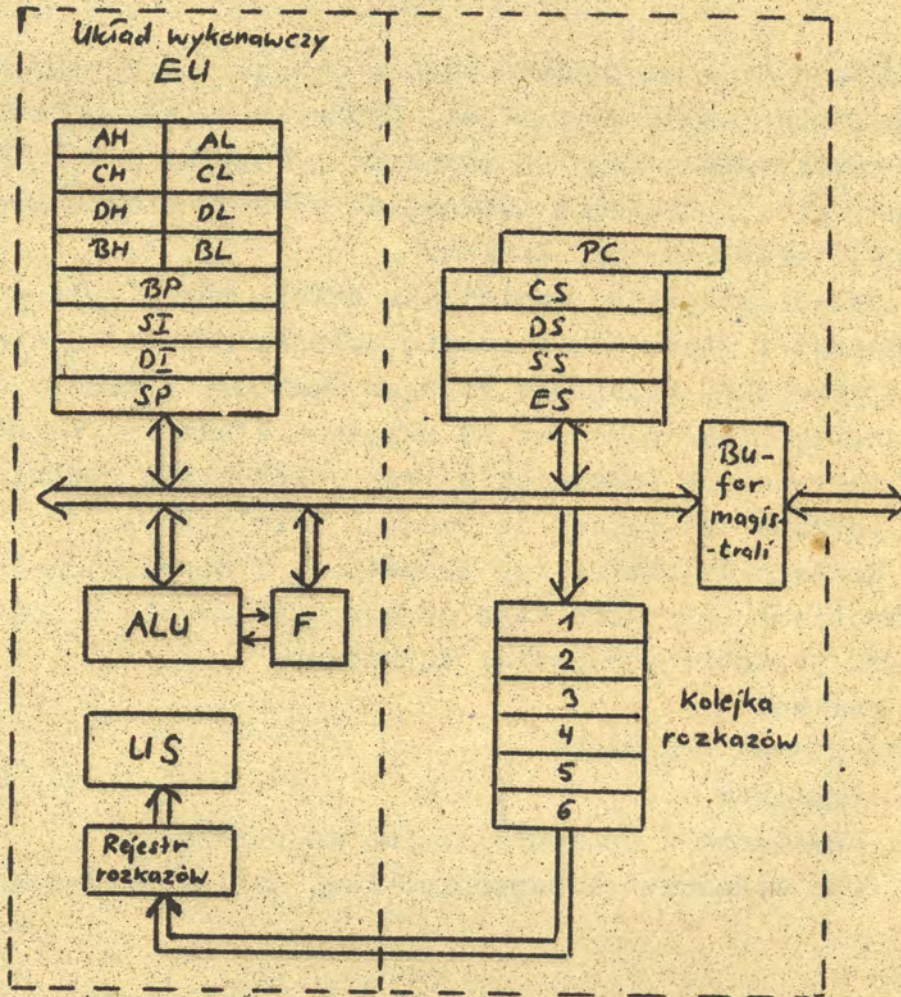
Sposób adresowania pamięci przy pobieraniu i wykonywaniu rozkazu dodawania ADD AL, [SI + d] jest przedstawiony na rys. 2.3. Rozkaz ten realizuje dodanie ośmiobitowej zawartości akumulatora AL do zawartości komórki pamięci o adresie logicznym SI + d.

Rejestr CS zawiera adres początku segmentu pamięci, w którym znajduje się program. W fazie pobrania rozkazu pamięć jest adresowana sumą pomnożonej przez 16 (w wyniku przesunięcia o 4 bity w lewo) zawartości rejestru CS i zawartości licznika rozkazów PC.

Rejestr DS zawiera adres początku segmentu, w którym znajdują się dane. Do pomnożonej przez 16 zawartości tego rejestru dodawany jest adres efektywny argumentu.

Rejestr SS zawiera adres początku segmentu pamięci, w którym znajduje się stos, a ES - adres początku segmentu pomocniczego.

Przedstawiony sposób adresowania pamięci rozwiązuje problem adresowania dużej pamięci przy użyciu 16-bitowych adresów, a ponadto:



Rys. 2.4. Schemat blokowy mikroprocesora 8086

- umożliwia oddzielenie programu od danych i stosu;
- ułatwia relokację (przesuwanie w pamięci) programu i danych;
- ułatwia gospodarke i ochronę pamięci przy pracy wieloprogramowej.

3. WEWNĘTRZNA ORGANIZACJA MIKROPROCESORA

Wewnętrzna budowa mikroprocesora 8086 jest schematycznie przedstawiona na rys. 2.4. Mikroprocesor składa się z dwu części, działających asynchronicznie, a mianowicie:

- układu sprzężenia magistrali BIU (ang. bus interface unit);
- układu wykonawczego EU (ang. execution unit).

Zadaniem układu sprzężenia magistrali BIU jest adresowanie pamięci (a więc przeliczanie adresu logicznego na fizyczny), pobieranie kolejnych rozkazów z pamięci i umieszczenie ich w 6-bajtowej

kolejne rozkazów, oraz - na żądanie układu wykonawczego - pobieranie lub przesłanie danych do pamięci. Zadaniem układu wykonawczego EU jest pobieranie kolejnych rozkazów z kolejki i ich wykonywanie, a więc m.in. obliczanie logicznych adresów danych i realizacja operacji przetwarzania danych.

Ponieważ układ sprzężenia magistrali działa równolegle z układem wykonawczym i stara się zapełnić kolejkę rozkazów, więc zazwyczaj faza wykonania kolejnego rozkazu zachodzi w czasie na fazy pobrania następnych rozkazów, co zwiększa efektywną szybkość pracy mikroprocesora (oczywiście przy wykonywaniu rozkazu skoku rozkazy pobrane "na zapas" są usuwane z kolejki).

Rozważmy dokładniej rozkaz np. dodawania. W konwencjonalnym komputerze pobranie i wykonanie tego rozkazu składa się z następujących kolejno po sobie poniższych czynności:

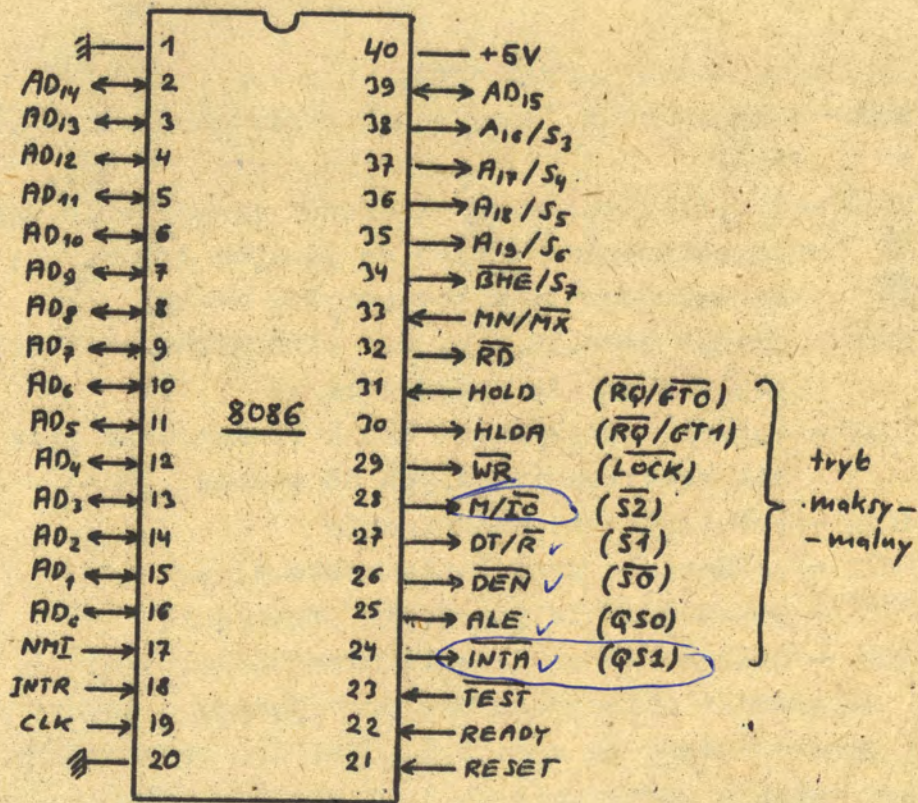
- pobranie rozkazu
- obliczenie adresu argumentu
- pobranie argumentu
- wykonanie dodawania i wpisanie wyniku do rejestru.

Mikroprocesor 8086 wykonuje poszczególne fazy cyklu rozkazowego równolegle:

- | <u>EU</u> | <u>BIU</u> |
|--|--|
| - pobranie rozkazu z kolejki i obliczenie adresu argumentu. Żądanie pobrania argumentu | - pobranie następnego rozkazu z pamięci i wpisanie go do kolejki |
| - układ wykonawczy czeka | - pobranie argumentu z pamięci i przesłanie go do EU |
| - wykonanie dodawania i wpisanie wyniku do rejestru | - ew. pobranie następnego rozkazu z pamięci (o ile w kolejce jest wolne miejsce) |
| - pobranie następnego rozkazu z kolejki. | |

4. SYGNAŁY MIKROPROCESORA

Mikroprocesor 8086 jest umieszczony w obudowie 40-końcówkowej (rys. 2.5), co powoduje konieczność multipleksowania adresu i danych. Po liniach AD_0 i AD_{15} przesyłane są zarówno 16-bitowe



Rys. 2.5. Końcówki mikroprocesora 8086

dane jak i 16 mniej znaczących bitów adresu. Po liniach A₁₆-A₁₉ są przesyłane bardziej znaczące bity adresu oraz bity statusu mikroprocesora. Bity statusu S₃ - S₇ określają rodzaj adresowanego segmentu pamięci, stan przerzutnika zezwalającego na przerwanie itp. - informacja ta w większości przypadków nie jest wykorzystywana.

Funkcje pełnione przez niektóre końcówki mikroprocesora zależą od trybu pracy, określonego przez sygnał podany na wejście MN/MX. Przy podaniu 1-ki na to wejście mikroprocesor pracuje w trybie minimalnym, bezpośrednio generując sygnały potrzebne do sterowania pamięciami i układami wejścia-wyjścia. Przy podaniu zera na wejście MN/MX mikroprocesor pracuje w trybie maksymalnym, właściwym dla systemów wieloprocesorowych. W trybie tym sygnały sterujące układami zewnętrznymi są generowane przez sterownik systemu 8288, opisany w następnych rozdziałach.

W trybie minimalnym mikroprocesor generuje klasyczne sygnały sterujące systemem mikroprocesorowym, a mianowicie:

ALE - sygnał wskazujący, że po magistrali AD jest przesyłany adres (tylne zbocze tego sygnału powinno wpisać adres

do rejestru adresowego);

\overline{DEN} - sygnał wskazujący, że po magistrali AD są przesyłane dane;

DT/\overline{R} - sygnał określający kierunek przesyłania danych;

\overline{WR} - sygnał zapisu danych do pamięci lub we/wy;

\overline{RD} - sygnał odczytu z pamięci lub we/wy;

M/\overline{IO} - sygnał określający, czy mikroprocesor zwraca się do pamięci czy do układów we/wy

HOLD - żądanie wstrzymania cyklu rozkazowego przy bezpośrednim przesyłaniu danych do pamięci (DMA)

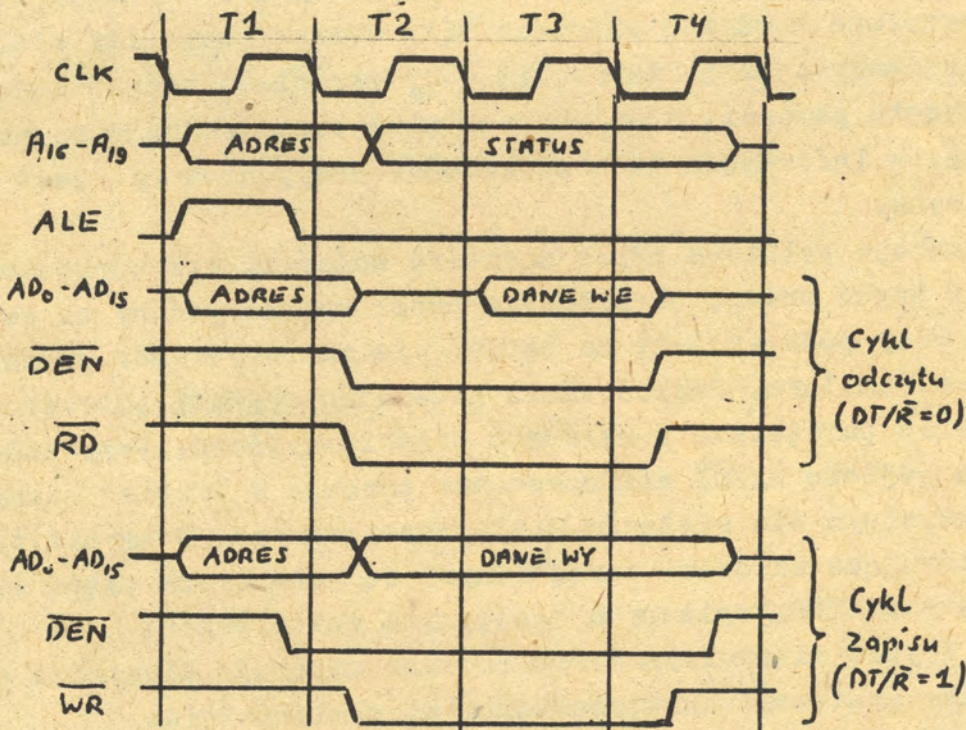
HLDA - potwierdzenie wstrzymania

INTR - zgłoszenie przerwania maskowalnego

\overline{INTA} - potwierdzenie przyjęcia przerwania

NMI - zgłoszenie przerwania niemaskowalnego.

Cykl magistrali mikroprocesora pracującego w trybie minimalnym jest przedstawiony na rys. 2.6. Cykl ten trwa cztery takty zegara (podawanego na wejście CLK). W pierwszym takcie po magistrali adresu/danych/statusu wysyłany jest adres, ustalony w momencie zmiany sygnału ALE z 1 na 0. W drugim takcie magistrala jest



Rys. 2.6. Cykl magistrali w trybie minimalnym

przygotowywana do odczytu lub zapisu danych, który następuje w trzecim taktie. Układy zewnętrzne o niewystarczającej szybkości działania mogą wygenerować sygnał niegotowości READY = 0, powodujący wstawienie dodatkowych taktów zegarowych Tw pomiędzy T3 a T4 (w dużych systemach sygnał gotowości jest normalnie równy zeru, a gdy zaadresowany układ zewnętrzny staje się gotowy do odbioru lub przesłania informacji - generuje sygnał READY = 1).

W trybie maksymalnym mikroprocesor generuje następujące sygnały:

$\overline{S0}$, $\overline{S1}$, $\overline{S2}$ - sygnały statusu, określające rodzaj wykonywanego cyklu magistrali, zgodnie z poniższą tabelką:

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Cykl
0	0	0	potwierdzenie przerwania
0	0	1	odczyt z układu we/wy
0	1	0	zapis do układu we/wy
0	1	1	stan stopu jednostki centralnej
1	0	0	pobranie rozkazu
1	0	1	odczyt pamięci
1	1	0	zapis do pamięci
1	1	1	pusty

Informacja ta jest wykorzystywana przez sterownik systemu do generowania sygnałów sterujących pamięciami i układami wejścia/wyjścia.

$\overline{QS0}$, $\overline{QS1}$ - sygnały określające stan kolejki rozkazów

$\overline{RQ/GT0}$, $\overline{RQ/GT1}$ - sygnały sterujące dostępem do magistrali lokalnej

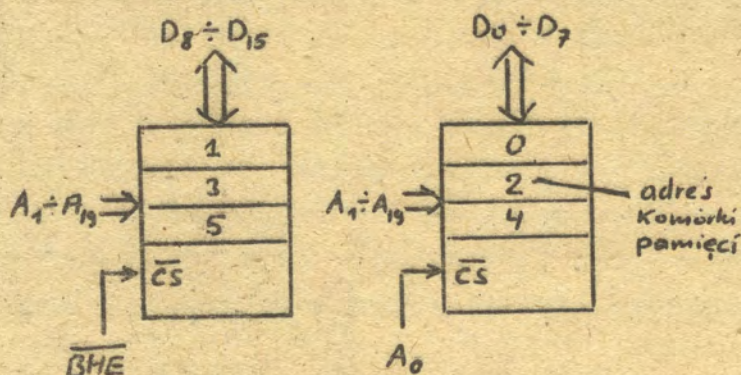
\overline{LOCK} - sygnał zapewniający wyłączność dostępu do magistrali systemowej

Sposób wykorzystywania powyższych sygnałów będzie wyjaśniony w następujących rozdziałach.

Wyjście \overline{RD} oraz wejścia READY, INTR i NMI funkcjonują tak samo w trybie maksymalnym jak i minimalnym.

Wejście RESET umożliwia wyzerowanie systemu i zainicjowanie pracy mikroprocesora. Wejście TEST jest badane rozkazem WAIT - w przypadku wykrycia 1-ki na tym wejściu mikroprocesor czeka na pojawienie się zera.

Mikroprocesor 8086 mimo iż jest mikroprocesorem 16-bitowym, adresuje pojedyncze bajty pamięci. Umożliwia to efektywne wyko-



Rys. 2.7. Sterowanie pamięci ze słowem 8-bitowym

rzystanie pamięci programu (rozkazy o długości od 1 do 6 bajtów) oraz operowanie na 8-bitowych danych (np. ciągach znaków alfanumerycznych). Jednocześnie, aby nie spowalniać działania systemu mikroprocesorowego, wprowadzono dodatkowy sygnał adresowy $\overline{\text{BHE}}$ (ang. bus high enable), umożliwiający przesyłanie w jednym cyklu magistrali słowa 16-bitowego pomiędzy mikroprocesorem a pamięcią.

Sposób sterowania pamięcią jest przedstawiony na rys. 2.7 (założono tu, że magistrale danych i adresu są rozdzielone za pomocą np. rejestru adresowego ładowanego sygnałem ALE). W zależności od wartości sygnałów A_0 i $\overline{\text{BHE}}$ podawanych na wejściu wybierające pamięć następuje przesłanie bajtu lub słowa 16-bitowego, zgodnie z poniższą tabelką:

$\overline{\text{BHE}}$	A_0	Przesłanie
0	0	słowa 16-bitowego (adres parzysty)
0	1	bardziej znaczącego bajtu (adres nieparzysty)
1	0	mniej znaczącego bajtu (adres parzysty)

Chcąc odczytać z pamięci pojedynczy bajt, mikroprocesor generuje sygnały $\overline{\text{BHE}} = A_0$. Wówczas przy $A_0 = 0$ po mniej znaczącej połowie magistrali (linie $D_0 - D_7$) jest przesyłany bajt o adresie parzystym, a przy $A_0 = 1$ po bardziej znaczącej połowie magistrali (linie $D_8 - D_{15}$) jest przesyłany bajt o adresie nieparzystym. Niezależnie od tego po której połowie magistrali danych bajt jest przesyłany, można go przesłać do dowolnego 8-bitowego rejestru procesora.

Jeżeli przy adresie parzystym ($A_0 = 0$) mikroprocesor występuje linię $\overline{\text{BHE}} = 0$, to nastąpi przesłanie całego słowa 16-bitowego. Przesłanie takiego słowa w jednym cyklu przy adresie niepa-

rzystym jest niemożliwe - wymaga to bowiem dwu cykli magistrali z jednoczesnym zwiększeniem adresu o 1.

Produkowana jest również 8-bitowa wersja mikroprocesora 8086, oznaczona symbolem 8088. Ma ona te same rejestry i listę rozkazów co 8086, lecz zewnętrzna magistrala danych jest 8-bitowa. Dane są multipleksowane z ośmioma mniej znaczącymi bitami adresu, a sygnał BHE nie jest generowany. Mikroprocesory 8086 i 8088 różnią się kilkoma innymi drobnymi szczegółami, m.in. kolejka rozkazów w 8088 ma tylko 4 bajty.

5. BIBLIOGRAFIA

1. Misiurewicz P., "Układy mikroprocesorowe", WNT, Warszawa 1983
2. Rector R., Alexy G., "The 8086 book" Osborne - Mc Graw Hill, Berkeley 1980
3. Osborne A., Kane G., "Osborne 16-bit microprocessor handbook" Osborne - Mc Graw Hill 1982.

VI Szkoła Mikroprocesorowa CSI/PTI
Łódź, Grudzień 1984

UKŁADY RODZINY MIKROPROCESOROWEJ INTEL 8086.

mgr inż. Zbigniew Barański
Instytut Technologii Elektronowej
Warszawa, Al. Lotników 32/46, 02-668
tel. 47-12-61

Istnieje duża liczba możliwych do realizacji konfiguracji układowych systemu mikroprocesorowego zbudowanego w oparciu o mikroprocesor 8086. Multipleksowanie sygnałów i podwójne funkcje spełniane przez niektóre z 40 wyprowadzeń CPU 8086 oraz zastosowanie specjalizowanych układów "rodziny" 86 stwarza projektantowi systemu możliwość wyboru rozwiązania optymalnego ze względu na koszty i możliwości funkcjonalne systemu: od rozwiązania najprostszego - konfiguracja minimalna - oszczędnego pod względem ilości użytych elementów ale o ograniczonych możliwościach funkcjonalnych i ograniczonych możliwościach rozbudowy systemu, do rozwiązań o rozbudowanej architekturze - konfiguracja maksymalna - z interfejsem do magistrali Multibus /lub pochodnej/, umożliwiającej pracę w systemie wielomikroprocesorowym i magistralą lokalną, zawierającą np. procesor we-wy 8089, koprocesor arytmetyczny 8087, kontroler przerwań, układy pamięci i we-wy.

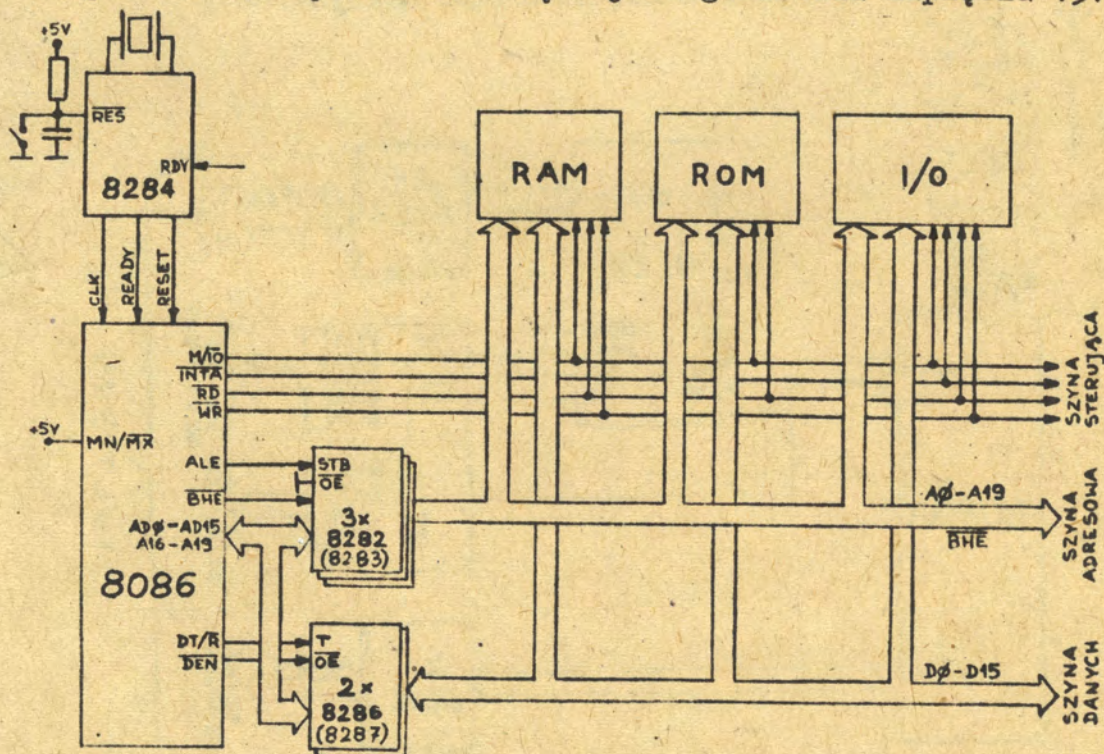
Poniżej wyszczególnione są układy należące do "rodziny" 86:

8086 - jednostka centralna CPU z 16-bitową zewnętrzną szyną danych,

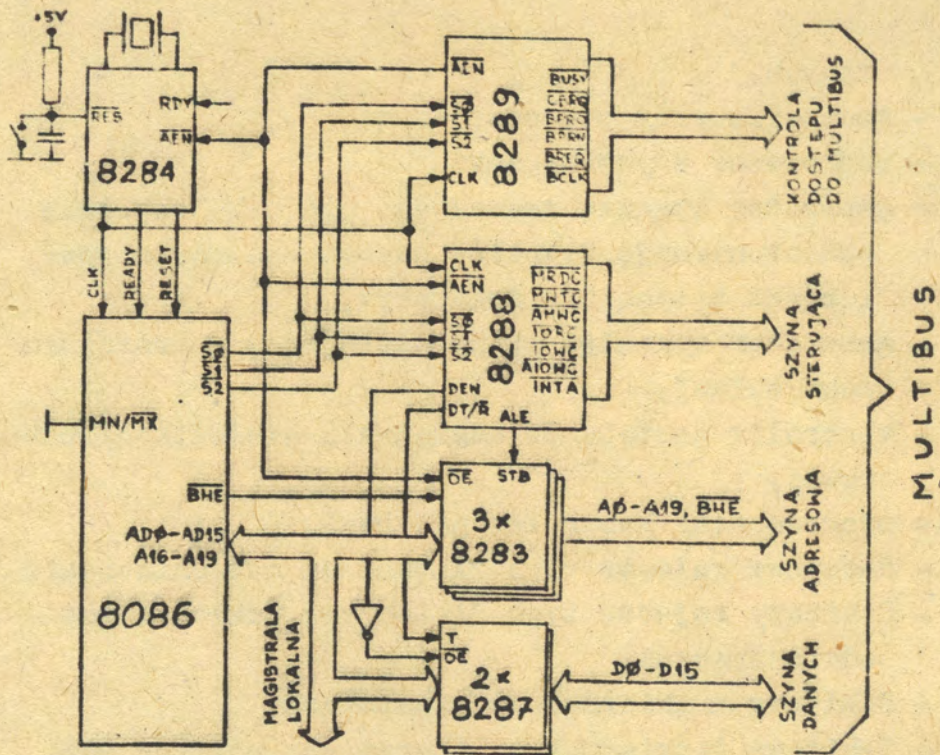
8088 - jednostka centralna CPU z 8-bitową zewnętrzną szyną

danych,

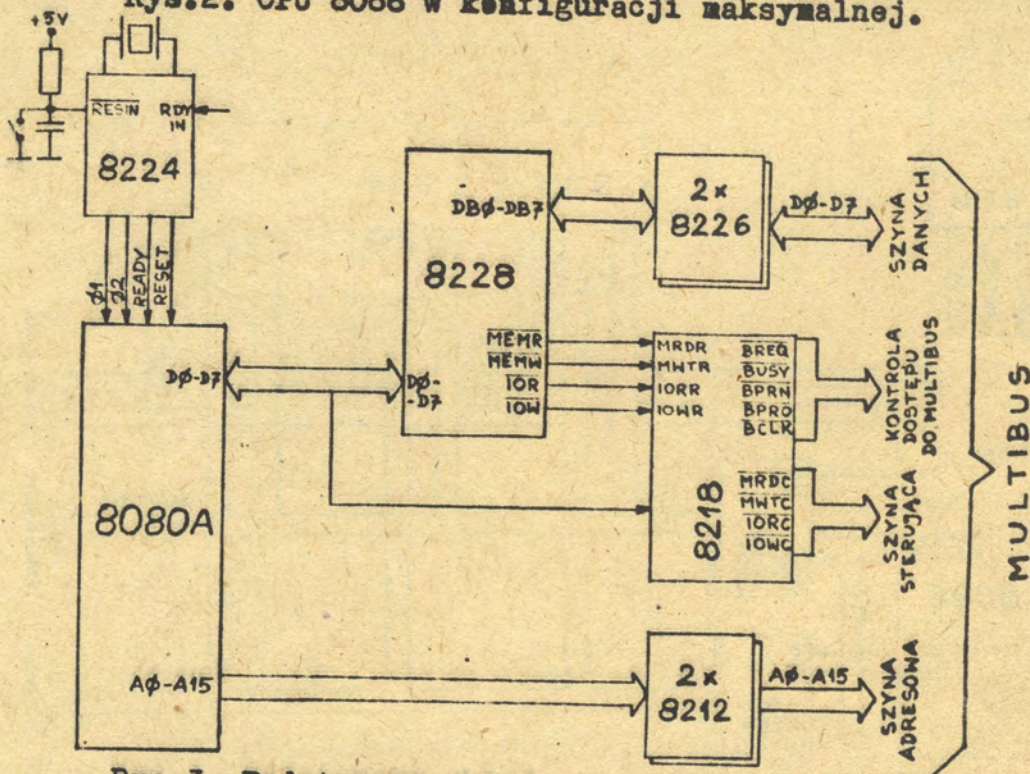
- 8089 - specjalizowany procesor we-wy,
 - 8087 - koprocesor arytmetyczny,
 - 8284 - generator sygnału zegarowego dla CPU, zapewnia synchronizację sygnałów asynchronicznych ready i reset z sygnałem zegarowym,
 - 8288 - generator sygnałów szyny sterującej w konfiguracji maksymalnej,
 - 8289 - kontroler dostępu do magistrali wielomikroprocesorowej,
 - 8259A - programowany kontroler przerwań,
 - 8282 - 8-bitowy rejestr typu "latch" ze wzmacniaczami,
 - 8283 - 8-bitowy rejestr typu "latch" ze wzmacniaczami odwracającymi,
 - 8286 - 8-bitowy dwukierunkowy wzmacniacz,
 - 8287 - 8-bitowy dwukierunkowy wzmacniacz odwracający.
- Wszystkie układy zasilane są z jednego źródła napięcia +5V.



Rys.1. CPU 8086 w konfiguracji minimalnej.



Rys.2. CPU 8086 w konfiguracji maksymalnej.



Rys.3. Podstawowy układ pracy CPU 8080A.

Na rys. 1 i 2 są przedstawione uproszczone schematy dwóch podstawowych konfiguracji pracy CPU 8086 wraz z układami towarzyszącymi: minimalna /rys.1/ i maksymalna z interfejsem do magistrali Multibus /rys.2/. Dla porównania na rys. 3 jest przedstawiony podstawowy układ pracy mikroprocesora 8080.

W obu układach z mikroprocesorem 8086 jako generator sygnału zegarowego wykorzystywany jest układ 8284. Ponadto, układ ten zapewnia że asynchronicznie pojawiające się sygnały ready i reset przed podaniem na odpowiednie wejścia 8086 zostają zsynchronizowane z sygnałem zegarowym CLK. Pozostałe funkcje układu 8284 zostaną dokładniej przedstawione w dalszej części pracy. Układ ten jest funkcjonalnym odpowiednikiem układu 8224, współpracującego z mikroprocesorem 8080 /rys.3/.

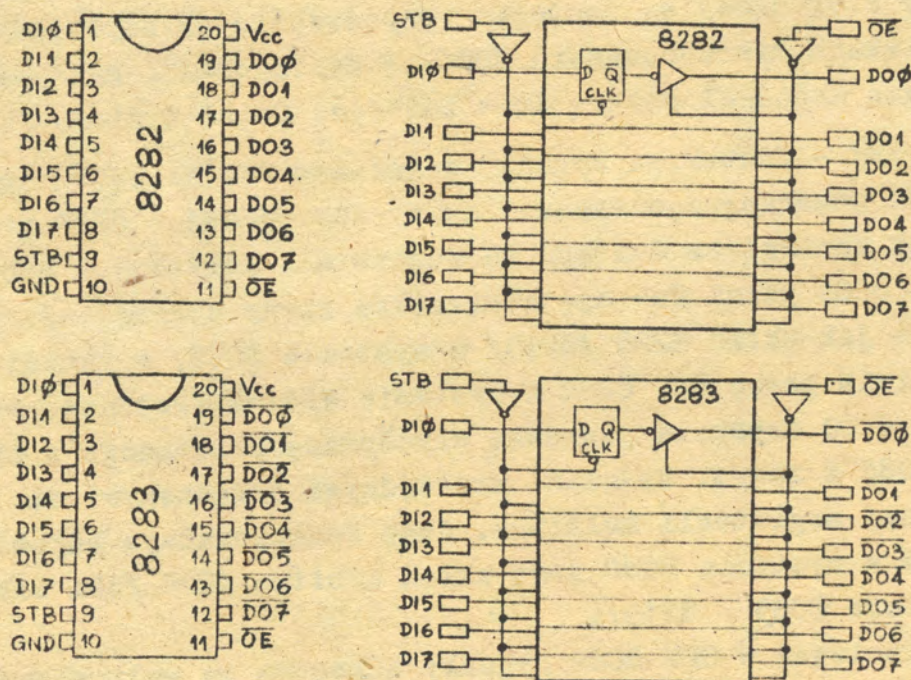
W konfiguracji minimalnej wszystkie potrzebne sygnały szyny sterującej są dostępne bezpośrednio na odpowiednich końcówkach mikroprocesora 8086. W konfiguracji maksymalnej te same końcówki pełnią inne funkcje. Część z nich - S \emptyset /, S1/, S2/ - dostarcza sygnały wejściowe dla zewnętrznego układu generującego sygnały szyny sterującej - 8288 i dla układu kontrolera dostępu do magistrali Multibus - 8289. Układ 8288 pełni funkcję kontrolera szyny sterującej, podobnie jak układ 8228 /8238/ w systemie 8080, a ponadto umożliwia pracę CPU 8086 w systemie wielomikroprocesorowym. Układ 8289 zapewnia poprawną współpracę związanego z nim CPU 8086 z innymi modułami nadrzędnymi dołączonymi do wspólnej magistrali Multibus. Jego funkcjonalnym odpowiednikiem w systemie 8080 jest układ arbitra 8218 /dla mikroprocesora 8085 - 8219/.

Ponieważ w CPU 8086 sygnały adresowe są multipleksowane z sygnałami danych na liniach AD \emptyset -AD15 oraz z sygnałami statusu na liniach A16/S3-A19/S6 i BHE//S7 /podobnie jak w mikroprocesorze 8085/, konieczne jest zatrzaśnięcie adresu pojawiającego się na tych liniach w pierwszym stanie

każdego cyklu maszynowego i zapamiętanie go na cały czas trwania cyklu. Do tego celu służą układy 8-bitowych rejestrów typu "latch" ze wzmacniaczami o trójstanowych wyjściach 8282 i 8283. Jeśli obciążenie szyny danych jest zbyt duże aby mogła ona być wysterowana bezpośrednio z kenców AD $\bar{0}$ -AD15 mikroprocesora 8086, dla zwiększenia wysterowania prądowego można zastosować układy 8-bitowych, dwukierunkowych, trójstanowych wzmacniaczy 8286 lub 8287.

W systemach z mikroprocesorem 8080, jako wzmacniacze szyn danych i adresowej zwykle stosowane są układy starszych typów: 8216 /lub 8226/ i 8212 pracujący jako wzmacniacz /nie ma potrzeby zapamiętywania adresu w rejestrach ponieważ szyny danych i adresowa nie są multipleksowane/.

1. OPIS UKŁADÓW 8282 I 8283.



Rys.4. Rejestry 8282 i 8283.

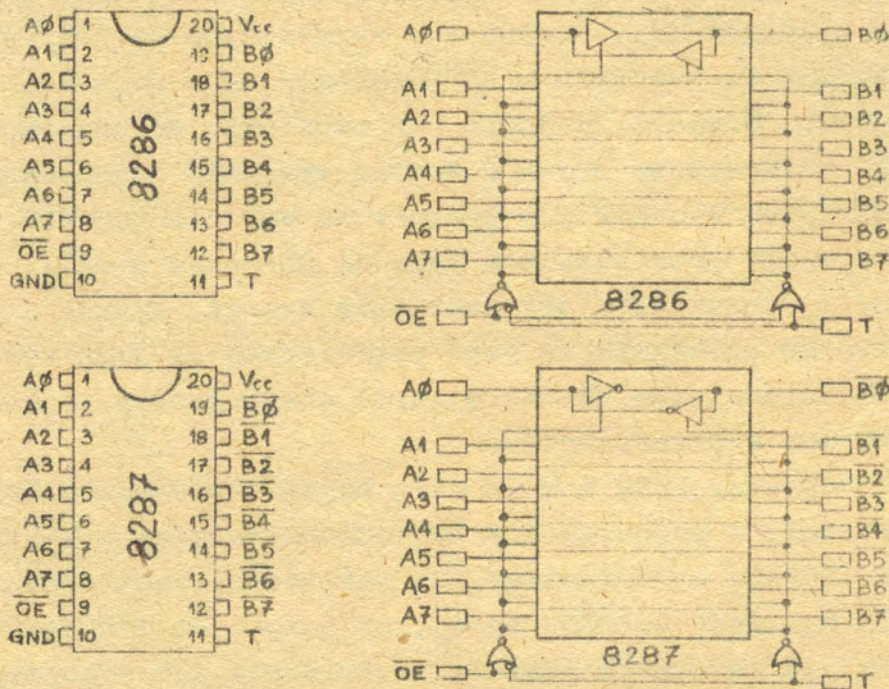
Układy rejestrów 8282 i 8283 zbudowane są z 8 przerzutników typu "latch" i trójstanowych wzmacniaczy prądowych na wyjściu każdego z przerzutników. Każdy z układów ma 8 sygnałów wejściowych i 8 wyjściowych: w 8282 sygnały wyjściowe są proste a w 8283 zanegowane w stosunku do wejściowych. Ponadto, wejście OE/ umożliwia wprowadzenie wyjść wzmacniaczy DO0-DO7 w stan wysokiej impedancji /gdy na OE/ stan wyseki/ lub podanie na wyjścia DO0-DO7 sygnałów z wyjść przerzutników /gdy na OE/ stan niski/. Gdy na wejściu strobuującym STB jest stan wysoki, sygnał z wejścia D każdego przerzutnika jest przenoszony na jego wyjście, a przy spadającym zboczach sygnału strobującego - jest zapamiętywany w przerzutniku.

Rejestry 8282 i 8283 mają mniej rozbudowany układ sterowania w porównaniu z ich odpowiednikiem z systemu 8080 - 8212, co ogranicza ich możliwości zastosowań w systemie np. jako układów we-wy, natomiast do ich zalet należy to, że parametry czasowe umożliwiają ich wykorzystanie w systemie z mikroprocesorem 8086 nawet przy maksymalnej częstotliwości sygnału zegarowego i są w mniejszej obudowie - co jest istotne z uwagi na 20-bitową szynę adresową CPU 8086.

2. OPIS UKŁADÓW 8286 I 8287.

Układy 8286 i 8287 są zbudowane z 8 trójstanowych, dwukierunkowych wzmacniaczy. W układzie 8286 sygnały wyjściowe są proste a w 8287 zanegowane w stosunku do sygnałów wejściowych. Stan wysoki na wejściu OE/ wprowadza wyjścia wszystkich wzmacniaczy w stan wysokiej impedancji, stan niski umożliwia transmisję z wejścia na wyjście. Poziom sygnału na wejściu T decyduje, które z końcówek A czy B są wejściami, a które wyjściami, tzn. decyduje o kierunku transmisji. Gdy na wejściu T jest stan wyseki,

sygnały z wejść A0-A7 są przenieszone na wyjścia B0-B7, gdy na T jest stan niski - sygnały z wejść B0-B7 są przenieszone na wyjścia A0-A7.



Rys.5. Wzmacniacze 8286 i 8287.

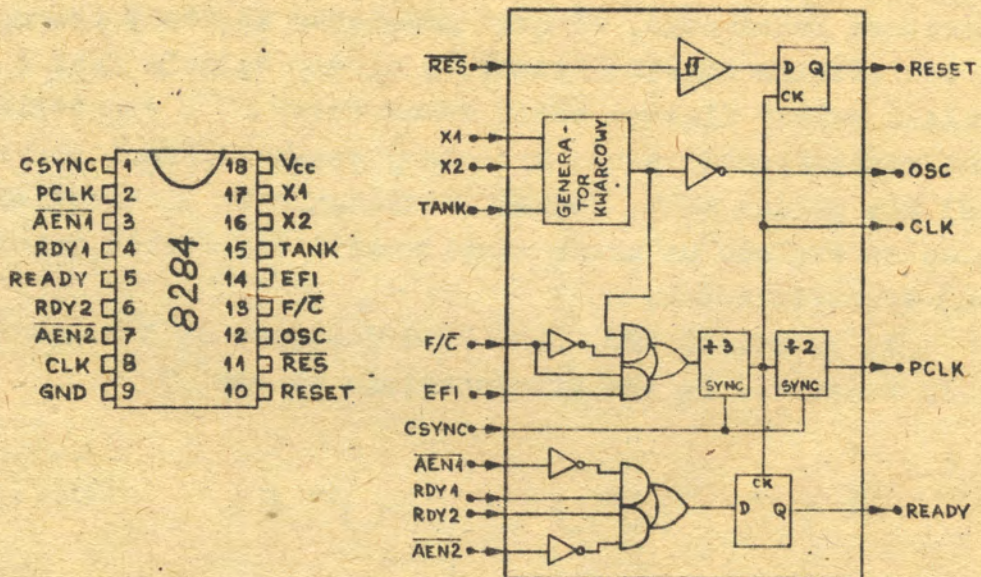
Zaletą układów 8286 i 8287 w stosunku do ich odpowiedników z systemu 8080 - 8216 i 8226 jest to, że zawierają w jednej obudowie większą liczbę wzmacniaczy: 8 zamiast 4. Ich parametry czasowe są dostosowane do współpracy z CPU 8086 przy maksymalnej częstotliwości sygnału zegarowego.

3. OPIS UKŁADU GENERATORA SYGNAŁÓW ZEGAROWYCH 8284.

Mikroprocesor 8086 wymaga sygnału zegarowego na wejściu CLK o ściśle określonych parametrach: napięcie w stanie niskim w zakresie -0.5V - +0.6V, w stanie wysokim +3.9V - - Vcc+1V, czasy narastania i opadania zbocz sygnału mniejsze od 10ns, maksymalna częstotliwość 5MHz /dla wersji 8086-2 8MHz/, przy częstotliwości maksymalnej współczynnik

wypełnienia równy 0.33. Ponieważ w strukturze mikroprocesora 8086 są zastosowane komórki pamięci dynamicznej wymagające odświeżania, częstotliwość sygnału zegarowego nie może być mniejsza niż 2MHz. Uniemożliwia to także wykonywanie pracy krokowej lub wstrzymywanie wykonywania programu przez odłączenie sygnału zegarowego.

Sygnału zegarowego o powyższych parametrach dostarcza układ 8284, którego schemat blokowy oraz oznaczenia końcówek są przedstawione na rys. 6.



Rys.6. Układ 8284.

Źródłem częstotliwości sterującej pracą układu 8284 może być: albo jego wewnętrzny generator kwarcowy - wówczas wejście F/C/ należy połączyć z masą układu, a między końcówki X1 i X2 dołączyć rezonator kwarcowy, albo zewnętrzne źródło częstotliwości dołączone do wejścia EFI - wówczas wejście F/C/ należy połączyć z +5V przez opornik 1k Ω ,

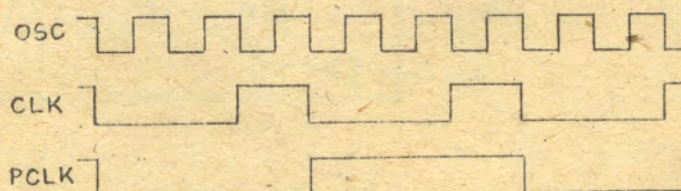
a sygnał podawany na wejście EFI powinien mieć poziomy logiczne wymagane dla układów TTL i współczynnik wypełnienia równy 0.5.

W obu przypadkach częstotliwość sygnału sterującego powinna być 3 razy większa niż częstotliwość sygnału zegarowego dla CPU 8086 pobieranego z wyjścia CLK.

Szeregowe z rezonatorem kwarcowym dołączonym do wejść X1-X2 powinien być włączony kondensator o pojemności 12-15pF w celu zredukowania wpływu indukcyjności wejściowej generatora kwarcowego oraz zabezpieczenia rezonatora kwarcowego przed polaryzacją napięciem stałym występującą między wejściami X1 i X2.

Istnieje też możliwość wykorzystania overtoneowego rezonatora kwarcowego, wówczas generator kwarcowy pracuje z powielaniem częstotliwości, a między wejście TANK i masę układu należy włączyć obwód rezonansowy LC o częstotliwości rezonansowej równej wymaganej częstotliwości sterującej dla układu 8284. Jednak rozwiązanie to nie jest zalecane ze względu na niską stabilność i dokładność generowanej częstotliwości.

Na wyjściach OSC, CLK, PCLK dostępne są 3 różne sygnały zegarowe. Są one przedstawione na rys. 7.

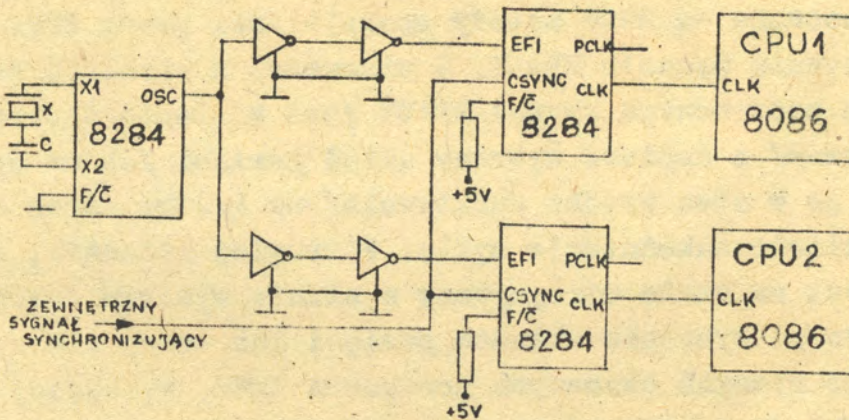


Rys.7. Sygnały zegarowe generowane przez 8284.

Częstotliwość sygnału na wyjściu OSC równa jest częstotliwości wyjściowej z generatora kwarcowego. Jeśli sterowanie 8284 odbywa się z wejścia EFI to sygnał na wyjściu OSC jest nieokreślony. CPU 8086 sterowane jest sygnałem zegarowym pobieranym z wyjścia CLK. Wyjście PCLK do-

staroza sygnału o częstotliwości równej połowie częstotliwości zegarowej CPU. Może on być dowolnie wykorzystany w układach peryferyjnych systemu.

Zgodnie ze schematem na rys. 6 sygnał na wyjściu OSC jest odwrócony w fazie w stosunku do sygnału wejściowego dzielnika przez 3, z którego uzyskuje się sygnał CLK dla CPU, tak więc w systemie, w którym kilka mikroprocesorów 8086 ma pracować synchronicznie nie można wykorzystywać wyjścia OSC układu 8284 jednego z nich, do sterowania wejścia EFI innych 8284. Synchroniczną pracę kilku CPU 8086 umożliwia natomiast układ przedstawiony na rys. 8.



Rys.8. Synchroniczna współpraca kilku układów 8284.

Jednak aby oba mikroprocesory CPU1 i CPU2 pracowały synchronicznie należy je najpierw zsynchronizować sygnałem zewnętrznym. Do tego celu służy wejście CSYNC układu 8284. Po włączeniu zasilania ustawia się przypadkowy stan początkowy dzielników przez 3 i przez 2 /rys.6/ i fazy sygnałów CLK i PCLK będą przypadkowe. Podanie poziomu wysokiego na wejście CSYNC wymusza stan wysoki na wyjściach CLK i PCLK. Po pojawieniu się stanu niskiego na CSYNC, oba układy 8284 /rys.8/ rozpoczną generowanie sygnałów CLK i PCLK o takiej samej fazie.

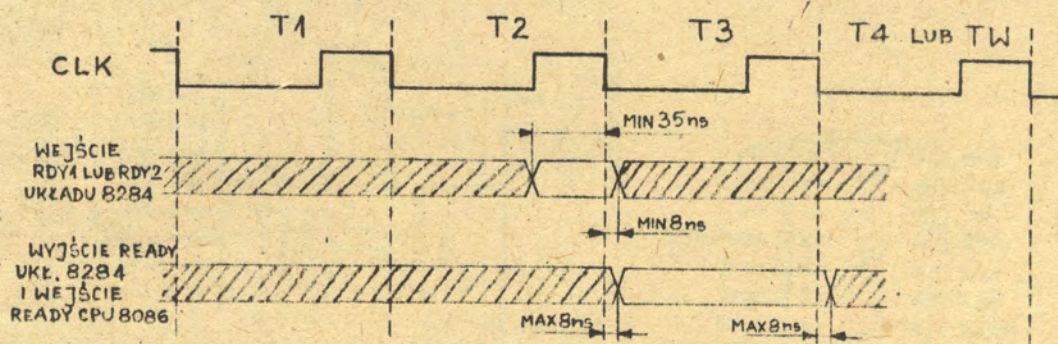
Wejście RES/ i wyjście RESET układu 8284 umożliwiają generację sygnału RESET wymaganego przez CPU 8086. Sygnał z wejścia RES/, po ukształtowaniu przez przerzutnik Schmitta jest synchronizowany sygnałem zegarowym CLK przed podaniem go na wejście RESET CPU 8086.

Sygnał wejściowy READY CPU 8086 wykorzystywany jest do dwóch celów: do spowalniania pracy procesora gdy współpracuje on z układami pamięci lub we-wy wymagającymi długiego czasu dostępu oraz do wstrzymywania pracy procesora gdy żąda on dostępu do magistrali systemowej w momencie gdy jest ona kontrolowana przez inny moduł. Spowalnianie i wstrzymywanie pracy procesora odbywa się przez wprowadzanie dodatkowych stanów wait.

Stosowane są dwie metody spowalniania pracy 8086 przy wykorzystaniu sygnału READY. W pierwszej z nich, od początku cyklu maszynowego sygnał READY jest w stanie niskim /nieaktywny/ a dopiero wybrany układ pamięci lub we-wy ustawia go w stan wysoki /uaktywnia/ na krótki okres czasu, aby umożliwić zakończenie cyklu. W drugiej metodzie, sygnał READY jest na stałe utrzymywany w stanie wysokim /aktywny/, a dopiero po wybraniu układów pamięci lub we-wy nie spełniających wymagań czasowych procesora 8086, wymuszany jest stan niski /nieaktywny/ na czas potrzebny do wprowadzenia wymaganej liczby stanów wait.

Dla każdej z powyższych metod są inne wymagania czasowe procesora 8086 dotyczące momentu pojawienia się i czasów trwania stanów niskiego i wysokiego na linii READY. Spełnienie tych wymagań zapewnia fragment układu 8284 generujący sygnał READY dla CPU 8086. Wejścia RDY1 i RDY2 układu 8284 umożliwiają kontrolę linii READY CPU 8086 z dwóch źródeł, np. jednym mogą być układy na magistrali systemowej, a drugim układy na magistrali lokalnej. Poziomy sygnałów na wejściach AEN/ decydują, który z sygnałów - RDY1 czy RDY2 - steruje linię READY /rys.6/. Gdy wykorzystywany jest tylko jeden z sygnałów RDY, na odpowiadającym mu wejściu

AEN/ powinien być stan niski, a na drugim wejściu AEN/ -
- stan wysoki, wówczas drugi sygnał RDY jest odłączony.
Sygnał READY podawany na CPU 8086 jest zsynchronizowany
z sygnałem zegarowym CLK. Jest to przedstawione na rys.9.



Rys.9. Synchronizacja sygnału RDY z sygnałem CLK.

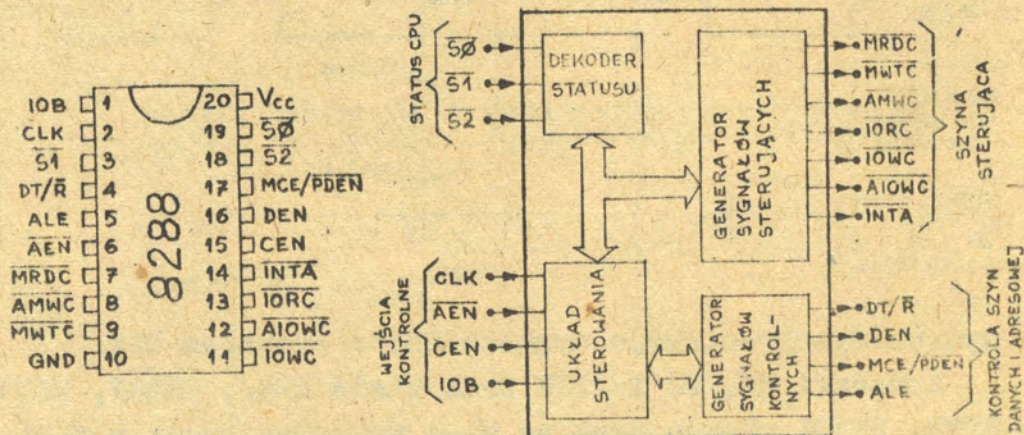
Końcówki X1, X2 i TANK oraz linie RES/, RESET, RDY1 i RDY2, READY układu 8284 są więc wykorzystywane w taki sam sposób jak odpowiadające im linie układu 8228 w systemie 8080, lecz obowiązują inne zależności czasowe.

4. KONTROLER SZYNY STERUJĄCEJ 8288.

Aby mikroprocesor 8086 mógł pracować w systemie wielo-
mikroprocesorowym za pośrednictwem magistrali Multibus
/lub pochodnej/ konieczne jest zastosowanie układu 8288
do generacji sygnałów szyny sterującej i praca CPU 8086
w konfiguracji maksymalnej.

W systemie z mikroprocesorem 8080, informacja o aktu-
alnie wykonywanym cyklu maszynowym przekazywana jest w po-
staci słowa statusu do kontrolera 8228 na szynie danych
w pierwszym stanie każdego cyklu maszynowego. Na podstawie
tej informacji układ 8228 generuje odpowiednie sygnały ste-
rujące w dalszej części cyklu. W systemie z mikroprocesorem
8086 również konieczne jest przekazywanie informacji o ro-
dzaju wykonywanego cyklu maszynowego, z CPU 8086 do układu
kontrolera 8288. Do tego celu służą sygnały statusu: S0/,

S1/ i S2/ dostępne na odpowiednich końcówkach CPU 8086 gdy pracuje on w konfiguracji maksymalnej. Mikroprocesor 8086 może współpracować z kontrolerem 8288 tylko w konfiguracji maksymalnej.



Rys.10. Kontroler 8288.

Redzaje cykli maszynowych wykonywanych przez CPU 8086, odpowiadające im sygnały statusu oraz sygnały sterujące generowane przez 8288 w danym cyklu zostały przedstawione w tabeli 1.

Ponieważ sygnały statusu generowane przez procesor we-wy 8089 są zgodne z sygnałami statusu CPU 8086, układ 8288 może też współpracować z procesorem 8089, gdy ten kontroluje niezależny moduł nadrzędny dołączony do magistrali Multibus.

Opis sygnałów układu 8288:

S0/,S1/,S2/ - sygnały statusu otrzymywane z mikroprocesora 8086 lub 8089 /tabela 1/.

CLK - sygnał zegarowy otrzymywany z układu 8284, który zapewnia synchroniczną współpracę 8288 i współpracującego z nim mikroprocesora.

ALE - sygnał generowany przez 8288 gdy na liniach AD0-AD15, A16/S3-A19/S6, BHE//S7 jest ważny adres, wykorzystywany

jako sygnał strobulujący do zapamiętania adresu w rejestrach 8282 /lub 8283/.

TABELA 1.

S2/ S1/ S0/	RODZAJ CYKLU	SYGNAŁ Z 8288
0 0 0	potwierdzenie przyjęcia przerwania	INTA/
0 0 1	czytanie z układu we	IORC/
0 1 0	zapis do układu wy	IOWC/,AIOWC/
0 1 1	stan HALT	żaden sygnał nie jest generowany
1 0 0	czytanie pamięci programu /FETCH/	MRDC/
1 0 1	czytanie pamięci	MRDC/
1 1 0	zapis do pamięci	MWTC/,AMWC/
1 1 1	stan bezczynności, żaden cykl nie jest wykonywany	żaden sygnał nie jest generowany

MRDC/ - sygnał odczytu danych z pamięci.

MWTC/ - sygnał zapisu danych do pamięci.

IORC/ - sygnał odczytu danych z układu wejścia.

IOWC/ - sygnał zapisu danych do układu wyjścia.

AIOWC/ - sygnał zapisu danych do układu wyjścia o wydłużonym czasie trwania, umożliwiającą współpracę z układami we-wy o czasie dostępu dłuższym niż wymagany przez CPU 8086, bez wprowadzania dodatkowego stanu wait /rys.11/.

AMWC/ - sygnał zapisu danych do pamięci o wydłużonym czasie trwania, umożliwiającą zapis danych do układów pamięci o czasie dostępu dłuższym niż wymagany przez CPU 8086 bez wprowadzania dodatkowego stanu wait /rys.11/.

INTA/ - sygnał potwierdzenia przyjęcia przerwania przez CPU 8086, wysyłany przez 8288 do kontrolera przerwania zgłaszającego przerwanie /8259A/.

DEN - sygnał wyjściowy z 8288 "otwierający" wzmacniacze szyny danych /8286 lub 8287/ i umożliwiającą transmisję danych z lub do CPU 8086.

DT/R/ - sygnał z 8288 kontrolujący kierunek transmisji danych przez układy wzmacniaczy 8286 lub 8287.

IOB - umożliwia wybór jednego z dwóch trybów pracy układu 8288: gdy na wejściu IOB jest stan wysoki /połączone z +5V/ układ pracuje w trybie z lokalną magistralą we-wy, gdy na wejściu IOB jest stan niski /połączone z masą/ - w trybie z magistralą systemową. W trybie pracy z magistralą systemową w każdym cyklu maszynowym 8288 generuje odpowiednie do rodzaju cyklu sygnały szyny sterującej /jeśli pozwala na to stan sygnału na wejściu AEN//. Tryb pracy z lokalną magistralą we-wy wykorzystywany jest wówczas, gdy na magistrali lokalnej CPU 8086 znajdują się tylko układy we-wy, a wszystkie odwołania do pamięci powodują żądanie dostępu do magistrali systemowej Multibus.

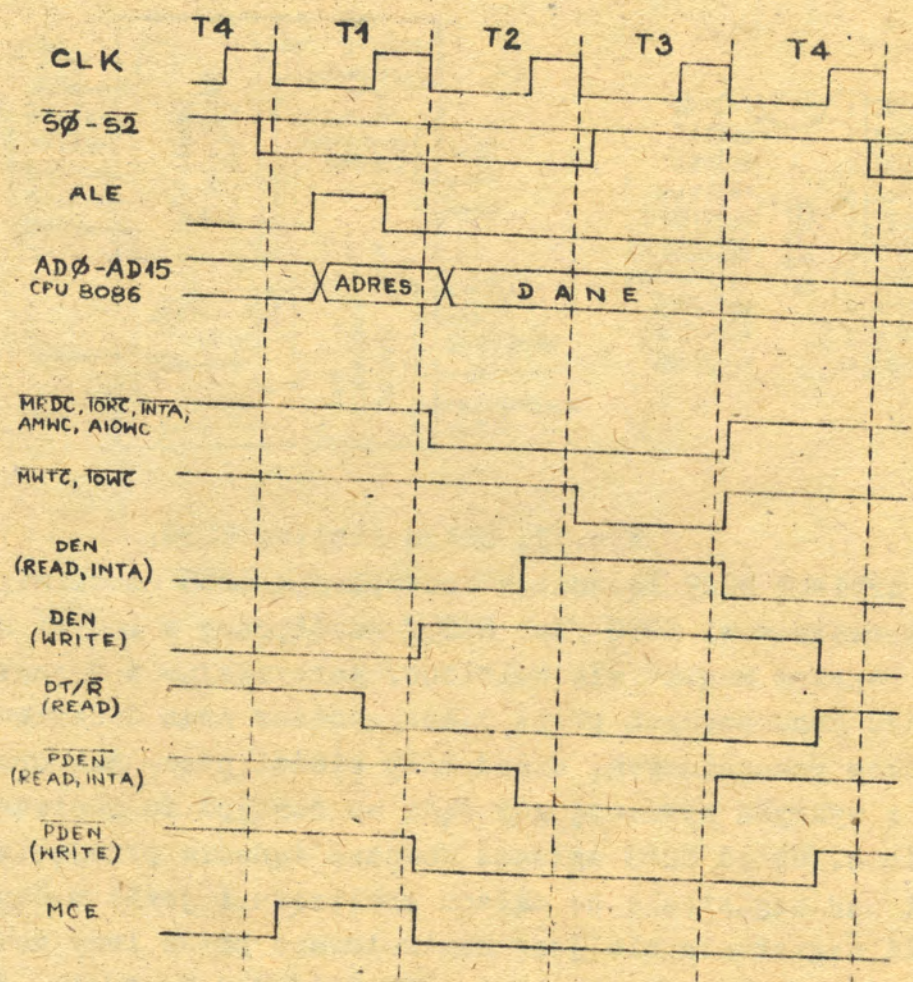
AEN/ - stan niski na wejściu AEN/ umożliwia generowanie sygnałów szyny sterującej przez 8288, stan wysoki - wprowadza wyjścia sygnałów szyny sterującej w stan wysokiej impedancji. Jeśli IOB jest w stanie wysokim /tryb pracy z magistralą we-wy/ i na AEN/ jest stan wysoki, w stan wysokiej impedancji nie są wprowadzane wyjścia 8288 generujące sygnały sterujące dla układów we-wy, tzn. IORC/, IOWC/, AIOWC/, INTA/.

MCE/PDEN/ - wyjście pełniące jedną z dwóch różnych funkcji, inną dla każdego trybu pracy 8288. W trybie pracy z lokalną magistralą we-wy, wyjście to generuje sygnał PDEN/ otwierający wzmacniacze lokalnej szyny danych i umożliwiający transmisję danych między CPU 8086 i układami we-wy dołączonymi do tej szyny. W trybie pracy z magistralą systemową sygnał MCE aktywny pojawia się tylko w czasie cyklu potwierdzenia przerwania i umożliwia kontrolerowi przerwania 8259A Master, znajdującemu się na szynie lokalnej CPU 8086, przesłanie adresu z linii CAS0-CAS2 poprzez szynę adresową systemu do jednego z kontrolerów przerwania 8259A Slave dołączonemu do magistrali Multibus, który zgłosił żądanie przerwania.

CEN - stan niski na tym wejściu wymusza stany nieaktywne wszystkich sygnałów szyny sterującej oraz DEN i PDEN/

układu 8288, niezależnie od rodzaju cyklu maszynowego wykonywanego przez CPU 8086. Stan wysoki umożliwia generację sygnałów szyny sterującej. Wykorzystywany jest, kiedy CPU 8086 współpracuje z dwoma kontrolerami 8288, z których jeden steruje układami pamięci i we-wy na magistrali Multibus, a drugi układami pamięci i we-wy na magistrali lokalnej.

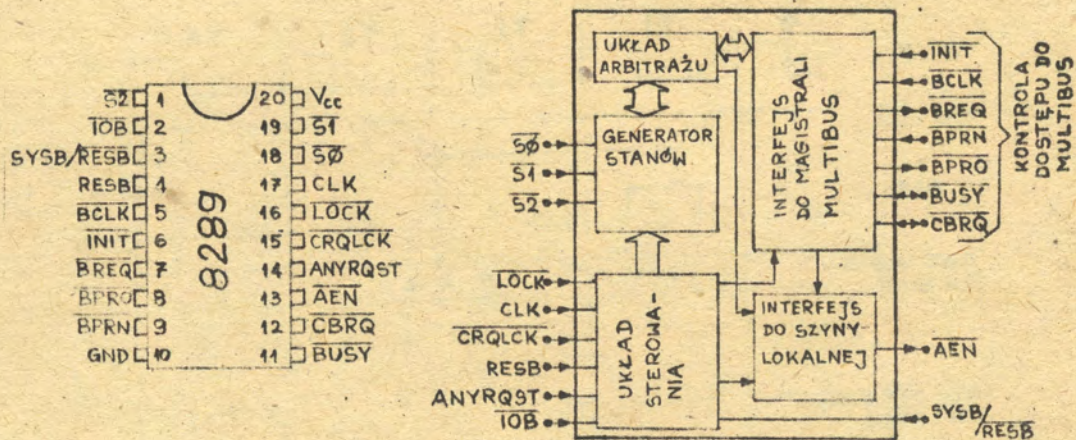
Sygnaly generowane przez 8288 przedstawione są na rys.11.



Rys.11. Sygnaly generowane przez 8288.

Sposób wykorzystania poszczególnych sygnałów układu 8288 zostanie dokładniej przedstawiony w dalszej części. Opisującej współpracę układów 8288 i 8289 z magistralą Multibus.

5. ARBITER MAGISTRALI MULTIBUS 8289.



Rys.12. Układ arbitra 8289.

Arbiter 8289 łącznie z kontrolerem 8288 umożliwia mikroprocesorowi 8086 /lub 8089/ współpracę z innymi modułami poprzez magistralę Multibus. Korzystając z sygnałów statusu generowanych przez mikroprocesor oraz dodatkowych sygnałów pomocniczych, układ 8289 śledzi pracę mikroprocesora i wykrywa sytuację gdy żąda on dostępu do magistrali Multibus. Układ 8289 zgłasza wówczas żądanie przejęcia kontroli nad magistralą do układu arbitrażu i jeśli w danej chwili magistrala nie jest kontrolowana przez inny moduł nadrzędny /master/, umożliwia CPU 8086 jej kontrolę. Jeśli jest kontrolowana - wstrzymuje pracę CPU 8086 przez wprowadzenie stanów wait do momentu aż zostanie zwolniona i do-

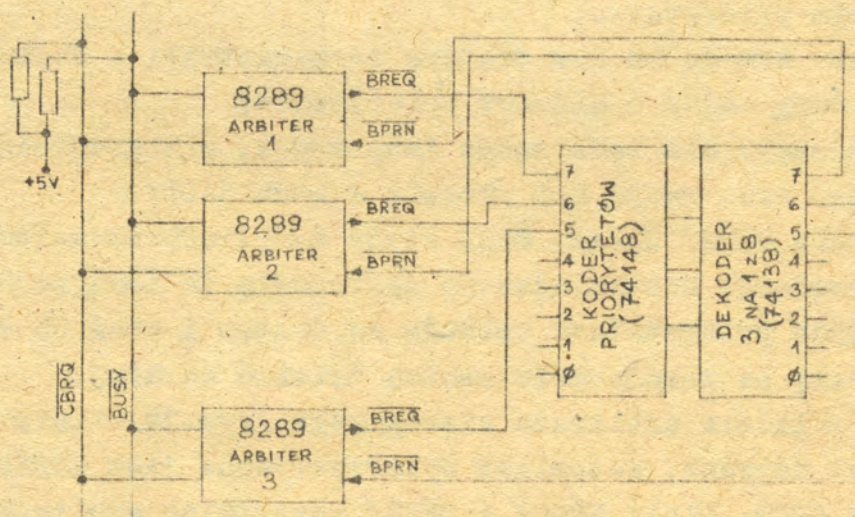
piero wówczas zezwala CPU 8086 przejęcie jej kontroli.

Układ arbitrażu służy do rozstrzygnięcia konfliktu powstającego w przypadku gdy kilka modułów nadrzędnych jednocześnie żąda dostępu do magistrali Multibus. Ponieważ każdy z modułów nadrzędnych ma określony priorytet dostępu do magistrali /każdy moduł ma inny priorytet/, układ arbitrażu przyznaje kontrolę nad magistralą temu spośród zgłaszających żądanie modułów, który ma najwyższy priorytet.

Stosowane są 3 rodzaje układów arbitrażu:

- układ arbitrażu równoległego,
- układ arbitrażu szeregowego,
- układ arbitrażu rotacyjnego.

Układ arbitrażu równoległego /rys.13/ wymaga zastosowania dwóch dodatkowych elementów: kodera priorytetów /74148/, do wejść którego dołączone są linie BREQ/ arbitrów 8289 każdego z modułów nadrzędnych oraz dekodera 3 na 1 z 8 /np. 74138/, którego wyjścia dołączone są do wejść BPRN/ odpowiednich arbitrów.



Rys.13. Układ arbitrażu równoległego.

Układ 8289 zgłasza żądanie dostępu do magistrali przez ustawienie linii BREQ/ w stan aktywny /niski/. Gdy jednocześnie na kilku wejściach kodera priorytetów są stany aktywne, na jego wyjściu pojawia się zakodowany binarnie

najwyższy numer wejścia o stanie aktywnym, a zatem tylko na jednym z wyjść dekodera /i wejściu BPRN/ połączonego z nim arbitra 8289/ pojawi się stan aktywny. Stan aktywny na wejściu BPRN/ arbitra 8289 oznacza, że moduł nadrzędny z nim współpracujący będzie mógł przejąć kontrolę magistrali po zwolnieniu jej przez inny moduł nadrzędny, który aktualnie ją kontroluje. Zostanie to zasygnalizowane pojawieniem się stanu nieaktywnego /wysokiego/ na linii BUSY/. Kiedy nowy moduł nadrzędny przejmie kontrolę magistrali, ponownie wprowadzi on linię BUSY/ w stan aktywny /niski/, sygnalizując innym modułom że jest ona zajęta. Priorytety modułów nadrzędnych są więc ustalone przez połączenie linii BREQ/ i BPRN/ ich arbitrów 8289 z wejściami i wyjściami o odpowiednich numerach układów: kodera priorytetów i dekodera.

Moduł nadrzędny kontrolujący magistralę zwolni ją tylko w trzech przypadkach:

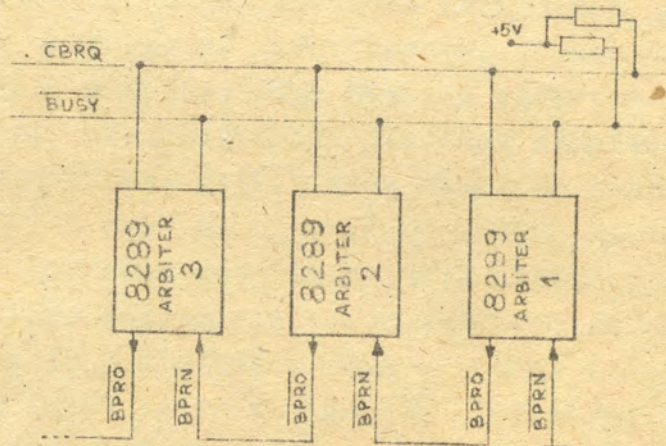
- gdy dostępu do magistrali żąda inny moduł nadrzędny o wyższym priorytecie,
- gdy nie wymaga on już dostępu do magistrali a dostępu żąda inny moduł o niższym priorytecie,
- gdy dostępu nie żąda żaden inny moduł, a mikroprocesor wykonał instrukcję HLT /wsadził w stan HALT/.

Linia CBRQ/ układu 8289 służy do poinformowania modułu nadrzędnego kontrolującego magistralę że inne moduły o niższych priorytetach czekają na dostęp i musi on zwolnić magistralę po zakończeniu swoich działań na niej.

W układzie arbitrażu szeregowego /rys.14/ linia BREQ/ nie jest używana, natomiast wykorzystywana jest linia BPRO/ arbitra 8289. Linie BPRO/ i BPRN/ wszystkich arbitrów są połączone szeregowo. Stan niski na linii BPRN/ układu 8289 informuje go, że żaden inny moduł nadrzędny o wyższym priorytecie nie kontroluje ani nie żąda dostępu do magistrali. Linia BPRN/ modułu nadrzędnego o najwyższym priorytecie jest więc połączona z masą. Jeśli jeden z modułów przejmuje

kontrolę magistrali, ustawia swoją linię BPRO/ w stan wysoki. Stan ten jest przenoszony kolejno na wejścia BPRN/ wszystkich innych modułów o niższych priorytetach, uniemożliwiając im przejęcie kontroli nad magistralą. Ponieważ czas propagacji sygnału z wejścia BPRN/ do wyjścia BPRO/ ma określoną wartość, a suma tych czasów dla wszystkich modułów nie może być większa niż okres zegara systemowego BCLK/ więc ogranicza to liczbę modułów nadrzędnych, które mogą być dołączone do magistrali Multibus. Przy częstotliwości maksymalnej BCLK/ /10MHz/ i zastosowaniu układu arbitrażu szeregowego liczba modułów nadrzędnych jest ograniczona do 3.

Układ arbitrażu rotacyjnego jest podobny do układu arbitrażu równoległego, a różnica polega na tym, że priorytety dostępu do magistrali poszczególnych modułów nadrzędnych nie są im przyporządkowane na stałe lecz są dynamicznie zmieniane po każdym cyklu dostępu. Spesób modyfikacji priorytetów może być dowolny, stosownie do potrzeb użytkownika.



Rys.14. Układ arbitrażu szeregowego.

Na wejście INIT/ układu 8289 jest podawany sygnał inicjalizujący /reset/ magistrali Multibus.

Sygnał wejściowy LOCK/ jest pobierany z wyjścia LOCK/ CPU 8086. Przez czas trwania stanu aktywnego /niskiego/ na linii LOCK/, arbiter 8289 nie straci kontroli nad magistralą na rzecz innego modułu nadrzędnego, nawet jeśli dostępu

do magistrali będzie żądał moduł o wyższym priorytecie.

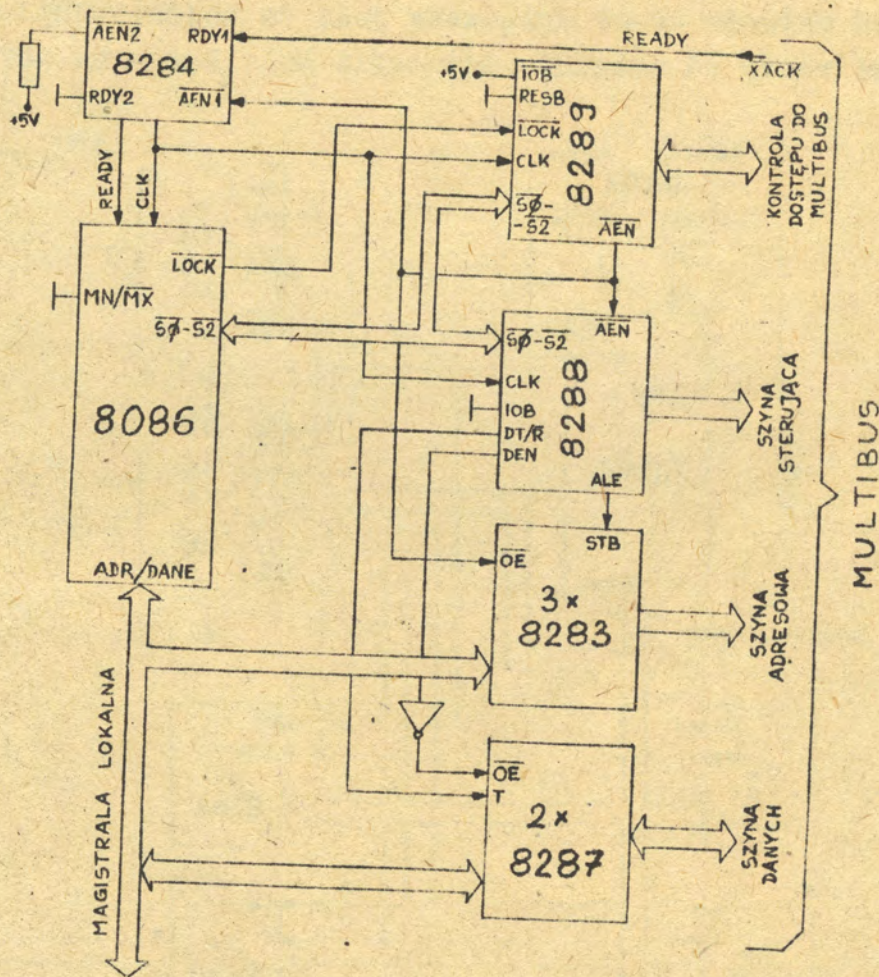
Na wejście CLK jest podawany sygnał zegarowy z układu 8284, ten sam, który steruje CPU 8086 i 8288.

Sygnał wyjściowy AEN/ wykorzystywany jest do wstrzymywania pracy CPU w czasie oczekiwania na dostęp do magistrali Multibus. Stan nieaktywny linii AEN/ powoduje zablokowanie sygnału RDY w układzie 8284 i wprowadzenie CPU w stan wait.

Pozostałe wejścia kontrolne arbitra 8289, tzn. CRQLCK/, RESB, ANYRQST, IOB/ są na stałe połączone z masą lub +5V i umożliwiają wybór jednego z trybów pracy układu. Stan aktywny /niski/ na wejściu CRQLCK/ powoduje że 8289 nie odda kontroli nad magistralą innym modułom, zgłaszającym żądanie dostępu na linii CBRQ/. Gdy na wejściu ANYRQST jest stan aktywny /wysoki/, na CBRQ/ jest stan niski a 8289 ma kontrolę nad magistralą Multibus, to po wykonaniu każdego cyklu maszynowego CPU 8086, arbiter 8289 traci kontrolę nad magistralą. Następnie, kontrola magistrali jest przyznawana na czas trwania jednego cyklu maszynowego modułowi o najwyższym priorytecie spośród zgłaszających żądanie. Ten sposób przydzielania kontroli nad magistralą jest zgodny ze sposobem pracy arbitrów 8218 i 8219 współpracujących z mikroprocesorami 8080 i 8085. Wejścia RESB i IOB/ określają konfigurację układu pracy mikroprocesora współpracującego z 8289. Gdy mikroprocesor posiada magistralę lokalną z układami pamięci i we-wy to RESB powinno być połączone z +5V, gdy takiej magistrali nie ma - z masą. Jeśli jest magistrala lokalna /na RESB +5V/ to stan sygnału na wejściu SYSB/RESB/ układu 8289 decyduje czy zażąda on dostępu do magistrali Multibus, gdy stan jest wysoki - tak, gdy niski - nie. Gdy mikroprocesor ma magistralę lokalną tylko z układami we-wy wówczas IOB/ jest połączone z masą, gdy takiej magistrali nie ma - na IOB/ jest +5V.

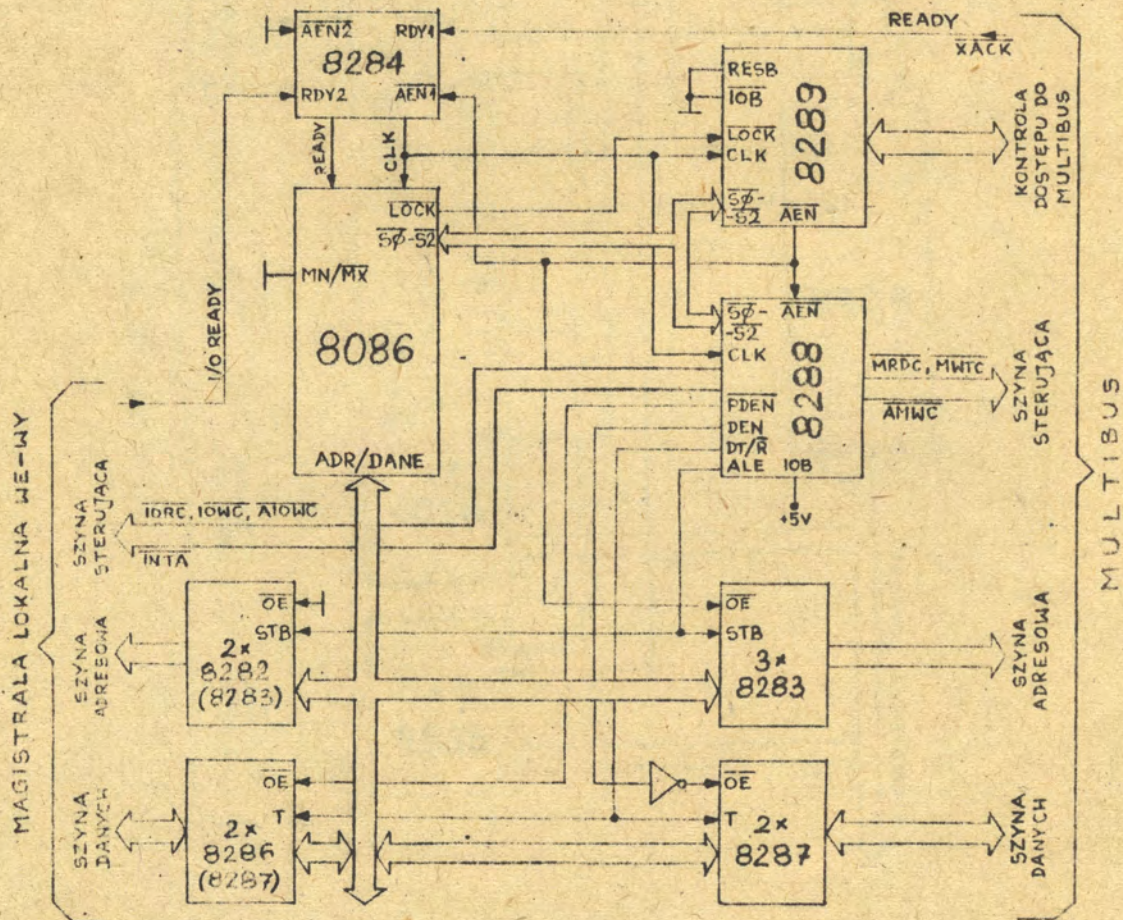
Różne warianty wykorzystania arbitra 8289 oraz pozostałych opisanych układów "rodziny" 86 przedstawiają poniższe przykłady.

Rys. 15 przedstawia najprostszy układ współpracy CPU 8086 z magistralą Multibus. Na magistrali lokalnej CPU 8086 nie ma ani pamięci, ani układów we-wy: wejście IOB/ układu 8289 połączone jest z +5V, a RESB i IOB 8288 z masą. Gdy arbiter 8289 kontroluje magistralę, niski stan na jego wyjściu AEN/ uaktywnia wyjścia rejestrów adresowych 8283 i umożliwia kontrolerowi 8288 generację sygnałów szyny sterującej oraz DT/R/ i DEN - kontrolujących transmisję danych przez wzmacniacze 8287.

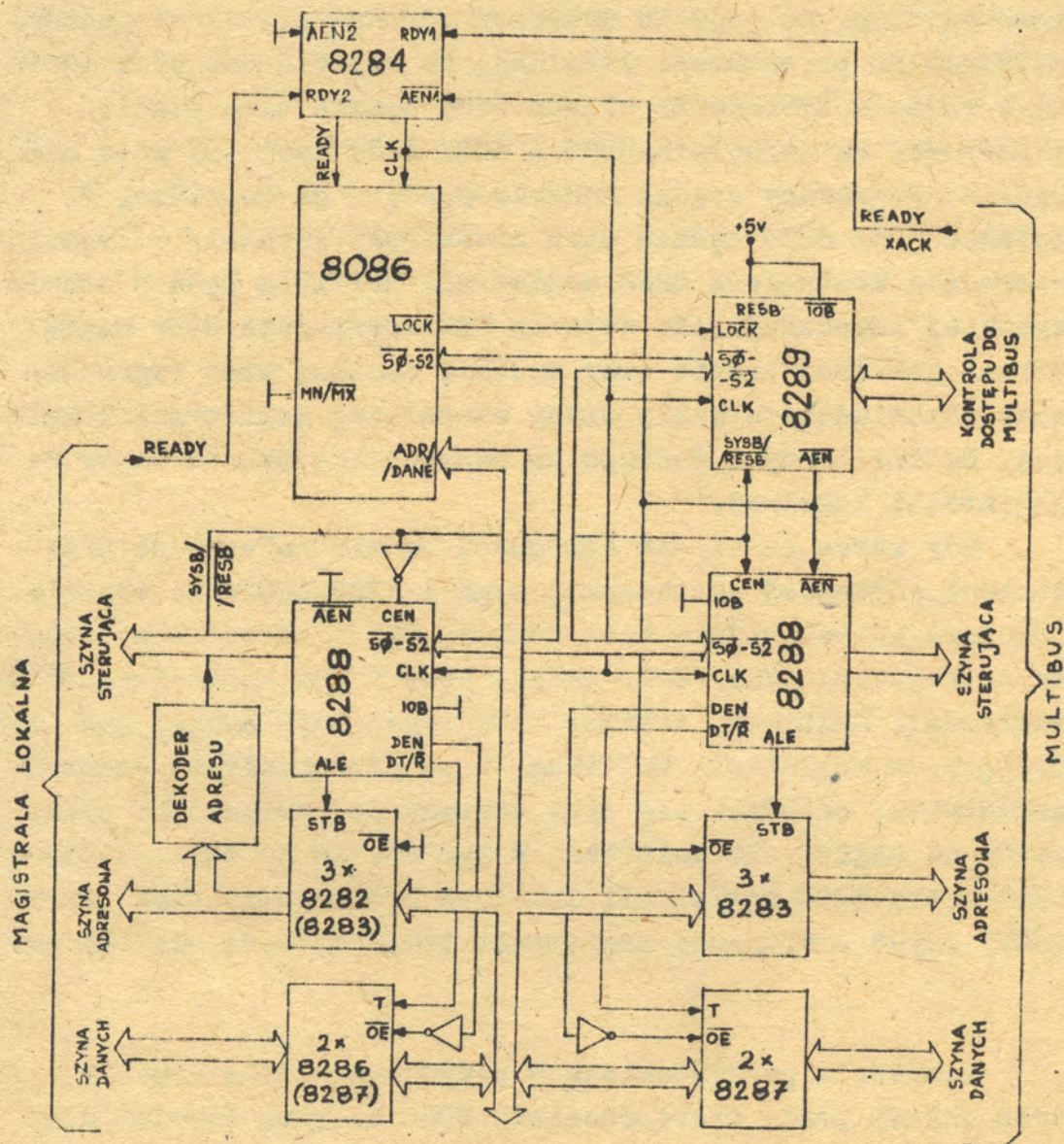


Rys.15. CPU 8086 współpracuje tylko z Multibus.

W układzie na rys. 16 CPU 8086 posiada szynę lokalną, do której dołączone są tylko układy we-wy. Wejścia arbitra 8289: RESB i IOB/ połączone są z masą, a wejście IOB układu 8288 z +5V. Wszystkie cykle maszynowe CPU odwołujące się do pamięci generują żądanie dostępu do Multibus, natomiast w cyklach zapisu lub odczytu układów we-wy żądanie dostępu do Multibus nie jest generowane. W czasie zapisu lub odczytu układów we-wy na wyjściu AEN/ 8289 jest stan nieaktywny /wysoki/, ale ponieważ na wejściu IOB 8288 jest +5V więc generowane są sygnały sterujące dla układów we-wy /IORC/, IOWC/, AIOWC/, INTA// znajdujących się na magistrali lokalnej. Sygnał ready od układów we-wy dołączony jest do wejścia RDY2 8284, a sygnał ready od układów magistrali Multibus - do RDY1.



Rys.16. CPU 8086 z lokalną magistralą we-wy.



Rys.17. CPU 8086 z magistralą lokalną zawierającą pamięć i układy we-wy.

W konfiguracji przedstawionej na rys. 17 CPU 8086 współpracuje z dwoma magistralami: Multibus i lokalną. Na obu magistralach mogą się znajdować zarówno układy pamięci jak i we-wy. Cała przestrzeń adresowa CPU 8086 jest podzielona na dwie części: systemową i lokalną. Jeśli w czasie

cyklu maszynowego wykonywanego przez CPU 8086 na szynie adresowej /tzn. na wejściu dekodera adresu/ pojawi się adres należący do przestrzeni lokalnej, na wyjściu dekodera adresu i wejściu SYSB/RESB/ układu 8289 będzie stan niski, a ponieważ na wejściach RESB i IOB/ 8289 jest +5V więc nie będzie generowany sygnał żądania dostępu do Multibus. Na wyjściu AEN/ 8289 będzie stan nieaktywny /wysoki/ i wyjścia sterujące kontrolera 8288 magistrali Multibus będą w stanie wysokiej impedancji. Na wejściu CEN kontrolera 8288 magistrali lokalnej będzie stan wysoki, zostaną więc wygenerowane odpowiednie sygnały szyny sterującej magistrali lokalnej. Będzie to cykl dostępu do pamięci lub układu we-wy na magistrali lokalnej.

Gdy adres na szynie adresowej będzie należał do przestrzeni adresowej systemowej, sygnał SYSB/RESB/ na wyjściu dekodera adresu będzie miał stan wysoki i zostanie on podany na wejścia: SYSB/RESB/ układu 8289 i CEN kontrolera 8288 magistrali Multibus. Arbiter 8289 wygeneruje sygnał żądania dostępu do magistrali Multibus, a po przyznaniu mu kontroli magistrali, odbędzie się cykl dostępu do pamięci lub układu we-wy na magistrali Multibus. W tym czasie na wejściu CEN kontrolera 8288 magistrali lokalnej będzie stan niski i sygnały szyny sterującej magistrali lokalnej będą nieaktywne.

Wszystkie przedstawione w powyższym opisie sygnały oraz układy pracy mikroprocesora 8086 dotyczą również mikroprocesora 8088, z następującymi wyjątkami:

- w CPU 8086 zewnętrzna szyna danych jest 16-bitowa, tzn. dane pojawiają się na liniach AD \emptyset -AD15, natomiast w CPU 8088 zewnętrzna szyna danych jest 8-bitowa i dane pojawiają się tylko na liniach AD \emptyset -AD7. Na liniach A8-A15 występują tylko sygnały adresowe,
- w układzie pracy w konfiguracji minimalnej na końcówce nr 28 CPU 8086 jest sygnał M/IO/, natomiast w CPU 8088

jest IO/M/.

BIBLIOGRAFIA:

1. MCS-86 User's Manual. Intel Corporation 1978.
2. The 8086 Family User's Manual. Intel Corporation 1979.
3. Rector R., Alexy G.: The 8086 book. Berkeley, OSBORNE/
/McGraw-Hill 1980.
4. Misiurewicz P.: Układy mikroprocesorowe. Warszawa, WNT
1983.

VI Szkoła Mikroprocesorowa CSI/PTI
Łódź, grudzień 1984

STANDARD MAGISTRALI MULTIBUS I STANDARDY POCHODNE

dr inż. Marek Tadeusz Jankowski
Przemysłowy Instytut Elektroniki
ul. Długa 44/50
00-241 Warszawa, tel. 47-12-52

1. WSTĘP

1.1 Definicje

INTERFEJS - /ang. interface/ granica między współpracującymi modułami, a ściślej definiująca tę granicę zasady współpracy i protokoły transmisji danych, a także zasady łączenia tych modułów oraz wykazy i parametry wymienianych między nimi sygnałów.

INTERFEJS WEWNĄTRZKASETOWY - /ang. backplane bus/ interfejs modułów stanowiących konstrukcyjnie pakiety umieszczone w jednej kasecie. Standard i.w. wymaga zdefiniowania wymiarów pakietów, typu/ów/ łączówek oraz rozmieszczenia na łączówkach sygnałów i napięć zasilających.

SZYNA - zespół linii danych, lub linii adresowych, lub linii kontrolno-sterujących, służących do przesyłania sygnałów między modułami połączonymi równolegle.

MAGISTRALA - 1. zespół szyn danych, adresowych oraz kontrolno-sterujących, a także linii zasilania. 2. dowolny w zasadzie interfejs o strukturze szynowej /definicja dotycząca również szeregowych interfejsów między kasetami bądź między urządzeniami/.

MASTER - moduł /pakiet/ nadrzędny, zdolny do kontrolowania magistrali i inicjowania transmisji danych przez magistralę.

SLAVE - moduł /pakiet/ podrzędny, zdolny do reagowania na polecenie transmisji wysłane przez moduł master.

1.2 Uwagi ogólne

Budowa współczesnych systemów mikroprocesorowych charakteryzuje się znaczną standaryzacją interfejsów. Są one opracowywane przez wiodące firmy-producentów sprzętu, lub przez profesjonalne organizacje takie jak IEC oraz IEEE, a w mniejszym stopniu przez ośrodki typu uniwersyteckiego. Z konstrukcyjnego punktu widzenia są to zarówno interfejsy wewnątrz-kasetowe, o typowych odległościach między pakietami do czterdziestu kilku cm. /co odpowiada kasiecie 19"/, jak i interfejsy między kasetami bądź między urządzeniami, o typowych odległościach od kilku do 2000 m.

Do pierwszej grupy interfejsów należą /w nawiasach podano źródło/ m.in.: S-100 /MITS/, Multibus i Multibus 2 /Intel/, AMS /Siemens/, VME /Motorola, Mostek, Signetics-Philips/, CAMAC i E3S /ESONE/ oraz P896 /IEEE/. Wspólną cechą tych interfejsów jest struktura szynowa i do wyjątków należą linie dochodzące do pakietów indywidualnie - jak np. LAM w standardzie CAMAC oraz linie arbitrażu BPRN/, BREQ/, BPRO/ w standardzie Multibus. Większość nowszych interfejsów stosuje przy tym wymiary pakietów i łączówki zgodne ze standardem mechanicznym Eurokarty.

Do drugiej grupy należą m.in. interfejsy: urządzeń pomiarowych IEEE 488 /Hewlett-Packard/, połączeń "point-to-point" lub gwiazdzystych - RS 232C i V-24 oraz takie magistrale jak SAN /ang. Small Area Network/ - I²C oraz D²B, a także LAN /ang. Local Area Network/ - PROWAY, IEEE 802 i Ethernet /DEC, Intel, Xerox/. Interfejsy te mogą się różnić konfiguracjami tworzonych sieci, a w przypadku magistrali - zasadami dostępu do niej /systemy hierarchiczne, z detekcją kolizji, "token passing"/. Niemniej jednak, niektóre protokoły - jak np. HDLC - mogą się w nich powtarzać.

W zasadzie interfejsy drugiej grupy, poza równoległym standardem IEEE 488, są interfejsami szeregowymi. Z kolei interfejsy pierwszej grupy mają strukturę równoległą, aczkolwiek najnowsze są już wyposażane w dodatkowe linie magistrali szeregowej.

Omawiany poniżej standard Multibus jest już klasycznym /o ile tak można mówić dla horyzontu czasowego ok. 10 lat/ i jednym z najbardziej rozpowszechnionych interfejsów wewnątrz-kasetowych. Spośród wymienionych wyżej standardów, dla porównania: CAMAC i S-100 są starsze i gorsze /aczkolwiek dobrze osadzone na rynkach/, AMS i VME są tej samej klasy, a E3S, P896 i Multibus 2 są bardziej nowoczesne. Ostatecznie o skoncentrowaniu się na standardzie Multibus zdecydowały więc czynniki do pewnego stopnia poza-techniczne:

- a/ produkowanie w Polsce /i szerzej - w RWPG/ układów mikroprocesorowych będących przede wszystkim odpowiednikami mikroukładów firmy Intel. Multibus funkcjonalnie jest zorientowany na te układy, a z drugiej strony Intel produkuje dla tego standardu specjalne adaptory "mikroprocesor-magistrala": 8218 dla mikroprocesora 8080, 8219 dla mikroprocesora 8085 oraz 8288 i 8289 dla mikroprocesorów 8086 i 8088.
- b/ w odniesieniu do również "intelowskiego" i nowszego standardu Multibus 2: większa prostota interfejsu Multibus, a zatem i większa ekonomia - przy zapewnieniu jednak możliwości tworzenia architektur wieloprocessorowych oraz wyprowadzania drugiej magistrali na łączówki /por. niżej/. Przy tym główne ograniczenie Multibus, jakim jest niemożność realizacji systemów 32-bitowych, w warunkach krajowych nie wydaje się być ważne do końca lat osiemdziesiątych, a zapewne i później.
- c/ stanie się tego standardu bazą dla prac normalizacyjnych zarówno w IEEE /projekt P796/, jak i w IEC /podkomisja 47B, Sekretariat 19 i 20/, a także na terenie RWPG /interfejs I-41, ustalenia komitetu SS-1/.

2. Standard magistrali Multibus

2.1 Wiadomości podstawowe

Koncepcja magistrali Multibus opiera się na rozróżnieniu dwu rodzajów modułów /pakietów/: master i slave. Pakiety master - np. procesorów albo kontrolerów DMA - są przede wszystkim źródłem sygnałów sterujących i adresów /i danych dla zapisu/. Natomiast pakiety slave - np. pamięci albo wejścia/wyjścia - są tylko źród-

łem lub odbiornikiem przesyłanych danych. Transmisja danych po magistrali odbywa się asynchronicznie, z inicjatywy pakietu master i zawsze wymaga potwierdzenia przez pakiet slave. Umożliwia to współpracę modułów o różnej szybkości, przy czym maksymalną szybkość transmisji może dochodzić do 5 milionów /bajtów albo słów 16-bitowych/ na sekundę. Magistrala dysponuje przy tym dwoma przestrzeniami adresowymi: dla pamięci o pojemności 16 Mbajtów oraz dla wejścia/wyjścia o pojemności 64 Kbajtów /odpowiednio 24 i 16 bitów adresu/.

W systemie wykorzystującym magistralę Multibus może być równocześnie kilka albo więcej modułów master. Niemniej jednak kontrolę i dostęp do szyn w danej chwili może posiadać tylko jeden moduł master - o czym decydują układy arbitrażu /por. p.2.4/.

Pakiety standardu Multibus mają wymiary 12" x 6,75" /304,8 x x 171,5 mm./. Pakiety te w płaszczyźnie magistrali /plater/ są wyposażone w dwie łączówki krawędziowe: P1 /2 x 43 kontakty, odstęp 0,156"/ oraz P2 /2 x 30 kontaktów, odstęp 0,1"/. Na łączówce P1 są rozmieszczone podstawowe sygnały interfejsu, wraz z napięciami zasilającymi. Na pomocniczej łączówce P2 bywają rozmieszczane alternatywnie:

- dodatkowe sygnały opcjonalne oraz najbardziej znaczące cztery bity adresowe ADR14/-ADR17/
- sygnały drugiej, lokalnej magistrali /np. interfejs ILBX/
- nietypowe, tzn. spoza standardu interfejsu, połączenia między sąsiednimi pakietami, stanowiącymi razem bardziej rozbudowany moduł /np. w systemie MDS-800 kontroler dysków albo emulator ICE/

Poniżej jest przedstawiony wykaz sygnałów interfejsu, a także podstawowe zasady transmisji danych oraz arbitrażu. Standard Multibus specyfikuje ponadto - w odniesieniu do sygnałów - wymagania elektryczne: charakter źródeł sygnałów /TTL, OC albo tzw. trójstanowe/, ich obciążalność, zasady polaryzacji i dopasowywania określonych linii /w większości przypadków należy stosować rezystory "pullup" 2,2 k Ω / itp. Zgodnie z przyjętą konwencją sygnały są aktywne lub oznaczają logiczną jedynkę dla poziomu niskiego -

- oznaczanego w dokumentacji jako /.

2.2 Elementy magistrali /sygnały łączówki P1/

Sygnały magistrali P1 o łącznej ilości 62, których rozmieszczenie jest pokazane w Dodatku 1, można podzielić na następujące grupy /o ile nie jest powiedziane inaczej, sygnały te są rozprowadzane szynami/:

A. Sygnały kontrolno-sterujące

- CCLK/** Zegar - sygnał okresowy ogólnego użytku, generowany przez jedno i tylko jedno źródło w systemie. Minimalny okres 100 ns., a wypełnienie 35 - 65 procent.
- MRDC/** Sygnał odczytu z pamięci
- MWTC/** Sygnał zapisu do pamięci
- IORC/** Sygnał odczytu z portu
- IOWC/** Sygnał zapisu do portu
- XACK/** Sygnał potwierdzenia transmisji przez moduł slave: wskazuje, że dane są "umieszczone" albo "odebrane" z szyny danych
- INIT/** Sygnał inicjalizacji bądź "zerowania": zazwyczaj służy do ustawiania systemu w określonym stanie przed rozpoczęciem pracy /przez dowolny moduł master bądź np. przez operatora/
- LOCK/** Sygnał generowany przez moduł master kontrolujący magistralę, stosowany najczęściej w operacjach semaforowych typu odczyt-modyfikacja-zapis, aby zapobiec dostępowi innego procesora do pamięci RAM /zwłaszcza dla tzw. multiported RAM/

B. Sygnały adresowe i inhibicji

- ADR \emptyset /-ADR13/** Sygnały adresacji pamięci bądź portów. Z 24 linii adresowych standardu Multibus - jak już było powiedziane - 4 znajdują się na łączówce P2. W przypadku portów wykorzystywanych jest 8 albo 16 linii, tzn. ADR \emptyset /-ADR7/ albo -ADRF/
- BHEN/** Sygnał służący do wybierania bardziej znaczącego bajtu, tzn. bity DAT8/-DATF/, słów 16-bitowych. Stosowany tylko w systemach zawierających moduły slave 16-bitowe
- INH1/-INH2/** Sygnały generowane przez moduły slave w celu uniemożliwienia działania innych modułów slave. INH1/ służy do "wyłączenia" RAM /np. przez ROM o tych samych adresach/, a INH2/ służy do "wyłączenia" ROM przez inny ROM /np. bootstrap/

C. Sygnały danych

- DAT \emptyset /-DATF/** Sygnały przesyłania danych do/z pamięci albo portów. W systemach 8-bitowych są stosowane tylko linie DAT \emptyset /-DAT7/

D. Sygnały przerwań

INT0/-INT7/ Sygnały przerwań - INT0/ ma najwyższy priorytet
INTA/ Sygnał potwierdzenia generowany przez moduł master kontrolujący magistralę - służy do umieszczania specyfikacji przerwania /ang. vector address/ na szynie danych

E. Sygnały arbitrażu

- BCLK/ Zegar - sygnał okresowy, którego ujemne zbocze synchronizuje inne sygnały arbitrażu, generowany przez jedno i tylko jedno źródło w systemie. Minimalny okres 100 ns., a wypełnienie 35 do 65 procent. Asynchroniczny w stosunku do CPU
- BUSY/ Sygnał generowany przez moduł master sprawujący aktualnie kontrolę magistrali - wskazuje, że jest ona zajęta
- BREQ/ Sygnał indywidualny /nie szyna/ stosowany przez moduły master - tylko w przypadku równoległego arbitrażu - do zgłoszenia żądania objęcia kontroli magistrali
- BPRN/ Sygnał indywidualny /nie szyna/ wskazujący danemu modułowi master, że żaden master o wyższym priorytecie nie żąda kontroli nad magistralą. Stosowany zarówno w szeregowym jak i w równoległym arbitrażu
- BPRO/ Sygnał indywidualny /nie szyna/ wskazujący modułowi o niższym priorytecie, przez zwarcie z jego BPRN/, że może kontrolować magistralę. Stosowany tylko przy szeregowym arbitrażu
- CBRQ/ Sygnał generowany na linii typu OC przez wszystkie moduły master żądające kontroli magistrali. Poziom wysoki CBRQ/ umożliwia modułowi aktualnie kontrolującemu magistralę utrzymanie kontroli bez powtarzania procedury arbitrażu przy każdej transmisji /co zabiera ok. 3 okresy BCLK/ - por. p.2.4/

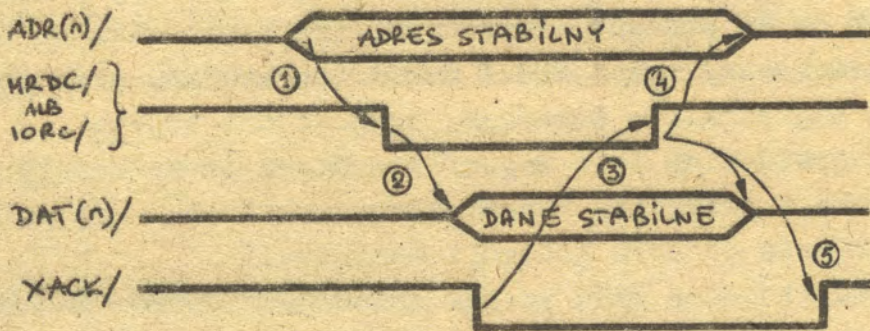
2.3 Transmisja danych

Transmisja danych, która stanowi główne zadanie interfejsu, przebiega w sposób następujący /por. rysunki 1 i 2/:

- A. Master kontrolujący magistralę /zgodnie z zasadami arbitrażu jest to jeden moduł/ wysyła adres pamięci albo portu
- B. Dla operacji zapisu master wysyła w tym czasie również dane
- C. Po upływie conajmniej 50 ns. master wysyła sygnał polecenia

wykonania operacji - zapisu albo odczytu - który to sygnał synchronizuje procesy w odpowiednim module slave

- D. Slave przyjmuje dane /dla zapisu/ albo wysyła dane /dla odczytu/
- E. Slave wysyła sygnał potwierdzenia, który powoduje koniec operacji w module master: usunięcie sygnału sterującego, a następnie adresu i danych



- ①. Czas ustalenia adresu, min. wyprzedzenie 50 ns.
- ②. Czas ustalenia danych, XACK/ może być wysłany gdy tylko dane są na szynie
- ③. Czas wymagany przez moduł master na usunięcie polecenia
- ④. Czas przetrzymywania adresu, min. opóźnienie 50 ns.
- ⑤. XACK/ i dane usunięte po max. 65 ns. od usunięcia polecenia

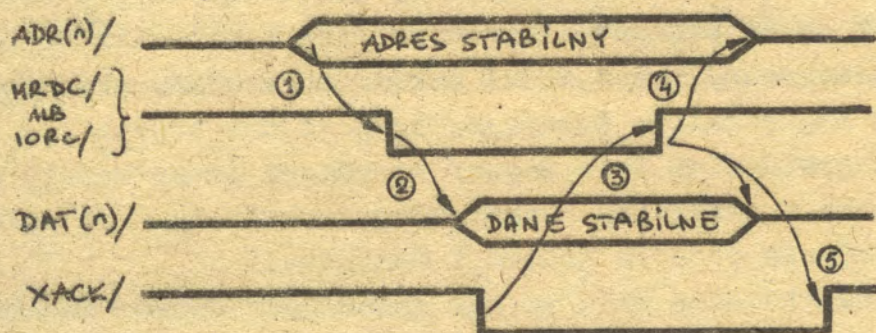
Rys.1 Wymagania czasowe dla odczytu /z pamięci albo z portu/

Przewidziane są cztery tryby transmisji danych, umożliwiające współpracę modułów 8- i 16-bitowych:

- I. Transmisja bajtów parzystych po liniach DAT \emptyset /-DAT7/, nieaktywne są zarówno BHEN/ jak i ADR \emptyset /
- II. Transmisja bajtów nieparzystych po liniach DAT \emptyset /-DAT7/, sygnał BHEN/ jest nieaktywny, a ADR \emptyset / jest aktywny
- III. Transmisja słów 16-bitowych: bajt parzysty po liniach DAT \emptyset /-DAT7/ a bajt nieparzysty po liniach DAT8/-DATF/, sygnał BHEN/ jest aktywny a ADR \emptyset / jest nieaktywny
- IV. Transmisja bajtów nieparzystych po liniach DAT8/-DATF/, aktywne są zarówno BHEN/ jak i ADR \emptyset / - tryb dotyczący tylko systemów 16-bitowych, niezalecany

wykonania operacji - zapisu albo odczytu - który to sygnał synchronizuje procesy w odpowiednim module slave

- D. Slave przyjmuje dane /dla zapisu/ albo wysyła dane /dla odczytu/
- E. Slave wysyła sygnał potwierdzenia, który powoduje koniec operacji w module master: usunięcie sygnału sterującego, a następnie adresu i danych

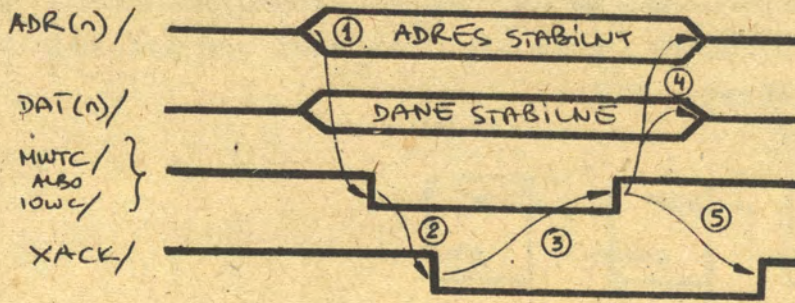


- ①. Czas ustalenia adresu, min. wyprzedzenie 50 ns.
- ②. Czas ustalenia danych, XACK/ może być wysłany gdy tylko dane są na szynie
- ③. Czas wymagany przez moduł master na usunięcie polecenia
- ④. Czas przetrzymywania adresu, min. opóźnienie 50 ns.
- ⑤. XACK/ i dane usunięte po max. 65 ns. od usunięcia polecenia

Rys.1 Wymagania czasowe dla odczytu /z pamięci albo z portu/

Przewidziane są cztery tryby transmisji danych, umożliwiające współpracę modułów 8- i 16-bitowych:

- I. Transmisja bajtów parzystych po liniach DAT \emptyset /-DAT7/, nieaktywne są zarówno BHEN/ jak i ADR \emptyset /
- II. Transmisja bajtów nieparzystych po liniach DAT \emptyset /-DAT7/, sygnał BHEN/ jest nieaktywny, a ADR \emptyset / jest aktywny
- III. Transmisja słów 16-bitowych: bajt parzysty po liniach DAT \emptyset /-DAT7/ a bajt nieparzysty po liniach DAT8/-DATF/, sygnał BHEN/ jest aktywny a ADR \emptyset / jest nieaktywny
- IV. Transmisja bajtów nieparzystych po liniach DAT8/-DATF/, aktywne są zarówno BHEN/ jak i ADR \emptyset / - tryb dotyczący tylko systemów 16-bitowych, niezalecany



- ①. Czas ustalenia adresu i danych, min. wyprzedzenie 50 ns.
- ②. Czas niezbędny dla modułu slave dla przyjęcia danych
- ③. Czas wymagany przez moduł master dla usunięcia polecenia
- ④. Czas przetrzymywania adresu i danych, min. opóźnienie 50 ns.
- ⑤. XACK/ musi być usunięty po max. 65 ns. od usunięcia polecenia

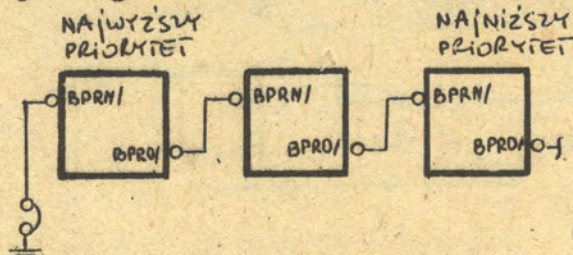
Rys.2 Wymagania czasowe dla zapisu /do pamięci albo do portu/

Odczyt z pamięci może być uniemożliwiony przez sygnał inhibicji INH1/ albo INH2/. Sygnał inhibicji musi być generowany nie później niż 100 ns. od momentu dysponowania stabilnym adresem - - sygnał ten powinien wyłączyć sygnały potwierdzenia i danych w modułach podlegających inhibicji. Natomiast wpływ inhibicji na proces zapisu w standardzie Multibus nie jest zdefiniowany.

2.4 Praca wieloprocessorowa - arbitraż

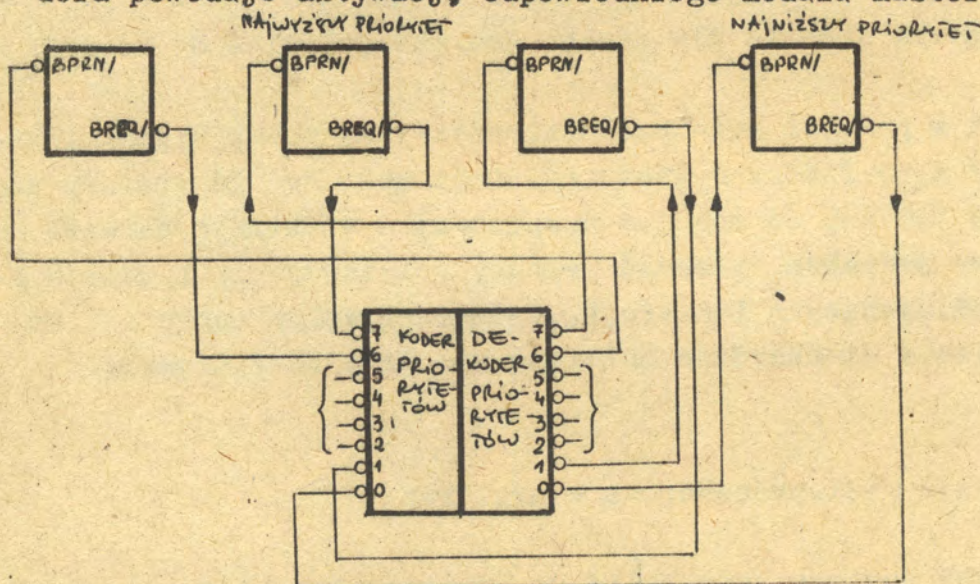
Arbitraż dostępu /kontroli/ do magistrali jest realizowany szeregowo albo równoległe. W obu przypadkach mamy do czynienia z modułami master pracującymi asynchronicznie i niezależnie, ale o sygnałach arbitrażu synchronizowanych przy pomocy zegara BCLK/. a/ Szeregowy arbitraż wykorzystuje technikę "daisy-chain" - por. rys.3. Wejście BPRN/ modułu master o najwyższym priorytecie jest dołączone na stałe do ziemi, jego wyjście BPRO/ jest dołączone do wejścia BPRN/ następnego modułu o niższym priorytecie itd. Dowolny master zgłaszający żądanie kontroli magistrali odbiera tu uprawnienia wszystkim modułom o niższym prio-

rytecie. Ilość modułów w łańcuchu szeregowego arbitrażu bywa jednak dość ograniczona: zależy ona od okresu BCLK/ oraz zboczy i opóźnień sygnałów, gdyż arbitraż musi być wykonany w czasie jednego okresu BCLK/.



Rys.3 Szeregową realizacją arbitrażu

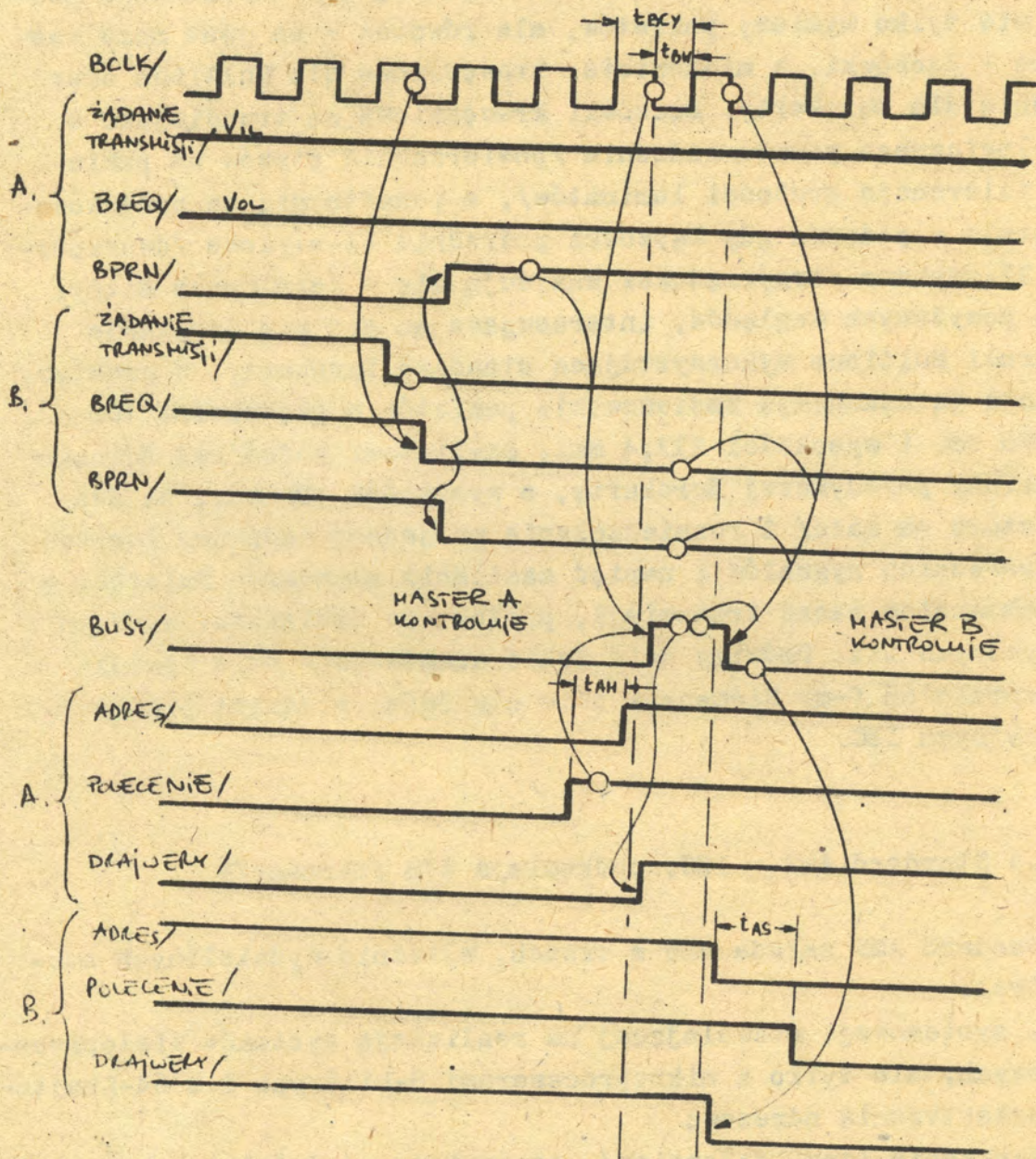
b/ Równoległy arbitraż niejako zamiast sygnałów BPRO/ wykorzystuje sygnały BREQ/. Sygnały te są doprowadzone do specjalnego układu priorytetu, na którego wyjściu jest zakodowane dojście sygnału BREQ/ o najwyższym priorytecie. Następnie układ dekodera powoduje aktywację odpowiedniego modułu master - por.rys.4.



Rys.4 Równoległą realizacją arbitrażu

Proces przejścia kontroli magistrali - na przykładzie arbitrażu równoległego - jest pokazany na rys.5. W przykładzie tym master A. posiada niższy priorytet niż master B. Przejęcie kontroli magistrali ma miejsce ponieważ master B. zgłasza żądanie objęcia kontroli w trakcie dostępu do magistrali przez moduł master A. Proces przejścia kontroli rozpoczyna się od wygenerowania sygnału BREQ/ przez moduł B., synchronicznie z ujemnym zboczem zegara BCLK/, a następnie zostają wygenerowane sygnały BPRN/ dla obu

modułów. Niemniej jednak, przed właściwym przejęciem kontroli przez moduł B, niezbędne jest zakończenie operacji aktualnie wykonywanej przez moduł A. i zwolnienie przez niego magistrali. Jest to sygnalizowane przez przejście sygnału BUSY/ z poziomu niskiego na wysoki - po czym, w fazie z następnym ujemnym zboczem zegara BCLK/ następuje objęcie kontroli przez moduł B.



Rys. 5 Przebiegi czasowe przy przejęciu kontroli magistrali /przykład arbitrażu równoległego/

3. Standardy pochodne

Wadą magistrali Multibus jest jej niezgodność z Eurokartą - standardem mechanicznym rozpowszechnionym w Europie i zatwierdzonym do stosowania na terenie RWPG /w Polsce konstrukcje mechaniczne Eurokarty są już produkowane w conajmniej dwu zakładach/. Różne są nie tylko wymiary pakietów, ale również - co jest może ważniejsze - łączówki, a mianowicie: krawędziowe dla Multibus oraz pośrednie dla Eurokarty. Łączówki krawędziowe są trudniejsze z technologicznego punktu widzenia /powierzchnia styków na pakietach, tolerancje grubości laminatów/, a ponadto nie są produkowane w kraju - podczas gdy łączówki pośrednie 64-stykowe /dwurzędowe/ i 96-stykowe /trzyrzędowe/ znajdują się w katalogach Eltry.

Z powyższych względów, interesujące są dla nas adaptacje magistrali Multibus wykorzystujące standard Eurokarty. W praktyce adaptacje te oznaczają zastosowanie pakietów o głębokości 160 albo 220 mm. i wysokości 233,4 mm., czyli tzw. podwójnej Eurokarty /pakiety pojedynczej Eurokarty, o wysokości 100 mm., są zdaniem autora za małe/ i rozmieszczenie na jednej bądź dwu łączówkach pośrednich sygnałów i napięć zasilania standardu Multibus - z zachowaniem zasad transmisji, protokołów arbitrażu, wymagań elektrycznych itd. Omówimy dwie takie adaptacje - obie wywodzą się z opracowań f-my Siemens i obie stanowią przedmiot prac normalizacyjnych IEC.

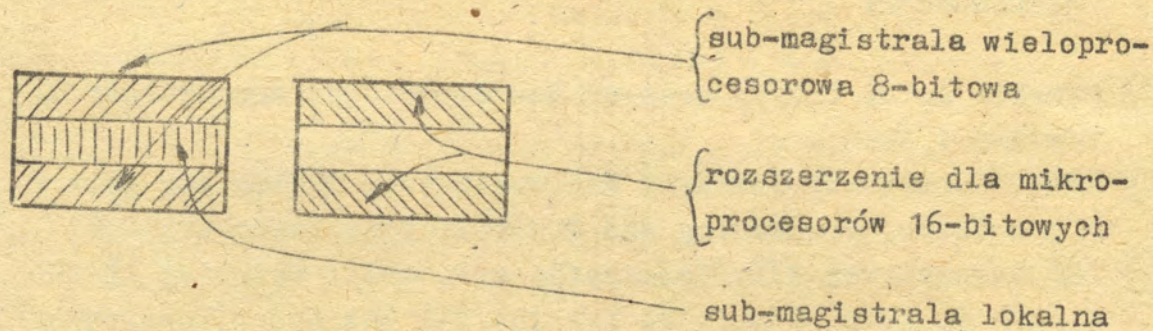
3.1 Standard AMS - IEC, podkomisja 47B /Germany/5

Standard AMS składa się z trzech, wyraźnie wydzielonych sub-magistral:

- a/ tzw. systemowej, pozwalającej na realizację systemów wieloprocesorowych, ale tylko z mikroprocesorami 8-bitowymi i z 64-Kbajtową przestrzenią adresową
- b/ rozszerzenia /ang. extension/, zapewniającej dodatkowych 8 linii danych dla mikroprocesorów 16-bitowych i 7 linii adresowych dla przestrzeni adresowej 8 Mbajtów, a także pewne sygnały pomocnicze, tzn. spoza łączówki P1 interfejsu Multibus
- c/ lokalnej /prywatnej, ang. resident/, przystosowanej do pracy z pojedynczym mikroprocesorem 16-bitowym o przestrzeni adresowej 1 Mbajta, z multipleksowaną szyną danych i adresów

Tak pomyślana magistrala /por. rys. 6/ stwarza możliwość tworzenia systemów o architekturze dwumagistralowej. Pozwala to na swobodną, autonomiczną pracę modułów master nie mających aktualnie dostępu do magistrali systemowej - moduły te mogą tworzyć subsystemy wyposażone np. w kilka pakietów slave, do których mają dostęp przez sub-magistralę lokalną. Ponadto jest tu możliwa oszczędna realizacja systemów bazujących tylko na mikroprocesorach 8-bitowych. W tym celu można wykorzystać jedynie sub-magistralę systemową rozmieszczoną na jednej łączówce 64-stykowej, po pewnych modyfikacjach rozmieszczenia linii zasilania.

Zalety standardu AMS sprawiły, że stał się on podstawą do opracowania krajowej normy branżowej interfejsu wewnątrzkaletowego dla automatyki. Niemniej jednak, konstrukcyjną wadą magistrali AMS jest umieszczenie sub-magistrali lokalnej na tej samej łączówce co systemowej - wady tej już nie ma opracowany rok później standard AMS-M.



Rys.6 Rozmieszczenie sygnałów AMS na łączówkach

3.2 Standard AMS-M - IEC, podkomisja 47B /Secretariat/ 21

Standard interfejsu AMS-M stanowi prawie dokładne przetransformowanie sygnałów łączówki P1 interfejsu Multibus na jedną łączówkę pośrednią 96-kontaktową pakietu Eurokarty. Umieszczono na niej dodatkowo linie adresowe ADR14/-ADR17/ oraz dwie linie nie zdefiniowanej jeszcze magistrali szeregowej, natomiast usunięto mało potrzebną linię zasilania -5V /por. Dodatek 2/. Tak więc stworzono możliwość swobodnego wykorzystywania drugiej łączówki pakietu - analogicznie do łączówki P2 pakietów Multibus - a w szczególności do poprowadzenia wydzielonej sub-magistrali lokalnej. W skrajnym przypadku istnieje możliwość zastosowania tego standardu również dla pakietów pojedynczej Eurokarty.

4. Uwagi końcowe

Standaryzacja interfejsu wewnątrzkasetowego w Polsce - poważnie utrudniona przez brak do tej pory mocnego producenta sprzętu mikrokomputerowego, zdolnego do narzucenia swoich konstrukcji - sprowadzi się najprawdopodobniej do wyboru między AMS, AMS-M oraz ewentualnie Multibus 2. Za AMS przemawia możliwość konstruowania tanich systemów na bazie mikroprocesorów 8-bitowych, a także wybranie go już jako bazy dla normy branżowej interfejsu dla automatyki. Za AMS-M przemawia zgrupowanie sygnałów wieloprocessorowego systemu 16-bitowego na jednej łączówce. Za Multibus 2 przemawia możliwość realizacji w przyszłości złożonych systemów 32-bitowych.

BIBLIOGRAFIA

- [1] Borrill L.P.: Microprocessor Bus Structures and Standards, EUROMICRO 1980
- [2] ESONE Committee: Standard Microprocessor System For Data Acquisition, Processing and Control, E3S, Luxemburg 1981
- [3] IEC Subcommittee 47B /Germany/3, Apr. 1982, Proposal of the German National Committee for the VME Bus Specification
- [4] IEC Subcommittee 47B /Germany/5, May 1982, Proposal of the German National Committee for the AMS Bus System
- [5] IEC Subcommittee 47B /Secretariat/19,20,21, July 1983, Microprocessor System Bus I, 8-bit and 16-bit Data
- [6] IEEE Computer Society: P896 - A Proposed Standard Backplane Bus Specification For Advanced Microcomputer Systems, Draft 4.1.2, Febr. 1982
- [7] Intel Multibus Specification, No 9800683-04
- [8] Intel Multibus Interfacing, Application Note AP-28A
- [9] Standard SEW tema 17.082.38-79, Pridłżenie 5.1, Tipowyje konstrukcii, typy i osnovnyje razmiery
- [10] IEC Subcommittee 48D: Dimensions of Subracs and Associated Plug-in Units for Mounting in Electronic Equipment, According to IEC Publication 297
- [11] Jankowski M.T.: KRABUS - sprzęg wewnątrz-kasetowy dla systemów mikrokomputerowych, Informatyka nr 1 1983

Dodatek 1

Rozmieszczenie sygnałów na łączówce P1 magistrali MULTIBUS

PIN	Strona elementów	PIN	Strona druku
1	GND	2	GND
3	+5V	4	+5V
5	+5V	6	+5V
7	+12V	8	+12V
9		10	
11	GND	12	GND
13	BCLK/	14	INIT/
15	BPRN/	16	BPRO/
17	BUSY/	18	BREQ/
19	MRDC/	20	MWTC/
21	IORC	22	IOWC/
23	XACK/	24	INH1/
25	LOCK/	26	INH2/
27	BHEN/	28	AD10/
29	CBRQ/	30	AD11/
31	CCLK/	32	AD12/
33	INTA/	34	AD13/
35	INT6/	36	INT7/
37	INT4/	38	INT5/
39	INT2/	40	INT3/
41	INT0/	42	INT1/
43	ADRE/	44	ADRE/
45	ADRC/	46	ADRD/
47	ADRA/	48	ADRB/
49	ADR8/	50	ADR9/
51	ADR6/	52	ADR7/
53	ADR4/	54	ADR5/
55	ADR2/	56	ADR3/
57	ADR0/	58	ADR1/
59	DATE/	60	DATE/
61	DATC/	62	DATD/
63	DATA/	64	DATB/
65	DAT8/	66	DAT9/
67	DAT6/	68	DAT7/
69	DAT4/	70	DAT5/
71	DAT2/	72	DAT3/
73	DAT0/	74	DAT1/
75	GND	76	GND
77		78	
79	-12V	80	-12V
81	+5V	82	+5V
83	+5V	84	+5V
85	GND	86	GND

IEC SUB-COMMITTEE 47B /SECRETARIAT/ 21
Eurocard Bus Connector P1 /96 Pin/

CONNECTOR PIN	a	b	c
1	MRDC/	BCLK/	IORC/
2	MWTC/	GND	IOWC/
3	BPRN/	BUSY/	INTA/
4	BPRO/	CBRQ/	BREQ/
5	NOT BUSSED	NOT BUSSED	NOT BUSSED
6	RESERVED /BUSSED/	GND	RESERVED /BUSSED/
7	INT 0/	INT 1/	INT 2/
8	INT 3/	INT 4/	INT 5/
9	INT 6/	GND	INT 7/
10	NOT BUSSED	XACK/	INIT/
11	+ 5V	NOT BUSSED	NOT BUSSED
12	+ 5V	+ 12V	+ 12V
13	{ RESERVED FOR SERIAL LINK /BUSSED/	GND	{ RESERVED FOR SERIAL LINK /BUSSED/
14	+ 5V	- 12V	- 12V
15	+ 5V	INH 1/	INH 2/
16	ADR 0/	ADR 1/	ADR 2/
17	ADR 3/	GND	ADR 4/
18	ADR 5/	ADR 6/	ADR 7/
19	ADR 8/	ADR 9/	ADR A/
20	ADR B/	GND	ADR C/
21	ADR D/	ADR E/	ADR F/
22	ADR 10/	ADR 11/	ADR 12/
23	ADR 13/	GND	ADR 14/
24	ADR 15/	ADR 16/	+ 5V
25	ADR 17/	BHEN/	+ 5V
26	LOCK/	GND	CCLK/
27	DAT 0/	DAT 1/	DAT 2/
28	DAT 3/	DAT 4/	DAT 5/
29	DAT 6/	GND	DAT 7/
30	DAT 8/	DAT 9/	DAT A/
31	DAT B/	DAT C/	DAT D/
32	DAT E/	GND	DAT F/

"/" indicates that the signal is active low

VI Szkoła Mikroprocesorowa CSI/PTI
Łódź, grudzień 1984

HIERARCHICZNY SYSTEM PRZERWAŃ W SYSTEMACH
WIELOMIKROPROCESOROWYCH

mgr inż. Jerzy Kręglewski
Instytut Automatyki
Politechnika Poznańska
ul. Piotrowo 3 A
60-965 Poznań, tel. 78-23-70

1. WSTĘP

Głównym obszarem zastosowań systemów mikroprocesorowych są systemy czasu rzeczywistego, a w szczególności systemy sterowania. W takich systemach, cechą mikroprocesora nabierającą szczególnego znaczenia staje się zdolność do szybkiego reagowania na zdarzenia zachodzące w otoczeniu mikroprocesora. Zdolność taką mikroprocesory posiadają dzięki wbudowanemu mechanizmowi przerwania. Wraz z rozwojem konstrukcji mikroprocesorów zmiane ulegają również wbudowane w nich systemy przerwania. W stosunkowo prostych mikroprocesorach 8-bitowych systemy przerwania zazwyczaj obejmują wektorowe przerwania maskowalne oraz niekiedy jednopoziomowe przerwania niemaskowalne.

W mikroprocesorach 16-bitowych, mikroprocesorach trzeciej generacji, pojawiają się bardziej doskonałe systemy przerwania obejmujące nie tylko przerwania maskowalne i niemaskowalne z otoczenia mikro-

procesora, ale również przerwania sygnalizujące zdarzenia zachodzące w wewnętrznych układach mikroprocesora. Dodatkowo, systemy te uwzględniają w swojej organizacji możliwość pracy w środowisku wielomikroprocesorowym.

W niniejszym artykule zajmiemy się omówieniem systemu przerwań w mikroprocesorze Intel 8086.

Mikroprocesor Intel 8086 jest przedstawicielem mikroprocesorów trzeciej generacji, mikroprocesorem 16-bitowym zawierającym w swojej architekturze liczne elementy niezbędne do pracy w systemie wielomikroprocesorowym. Omawiając poszczególne elementy systemu przerwań tego mikroprocesora zwracać będziemy uwagę na jego cechy, które uwzględniają wielomikroprocesorowość.

W drugiej części tego artykułu zajmiemy się programowalnym sterownikiem przerwań typu Intel 8259A omawiając szczegółowo te jego cechy, których nie posiadał jego poprzednik, tj. sterownik typu Intel 8259.

2. PRZERWANIA W MIKROPROCESORZE 8086

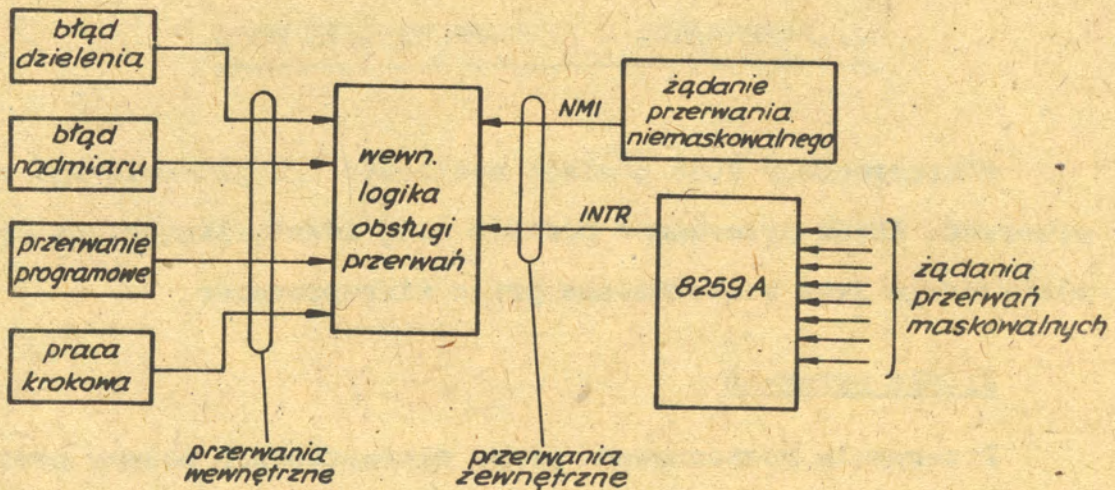
Mikroprocesor 8086 posiada możliwość przyjmowania 256 różnych przerw. Każde przerwanie posiada swój identyfikator, na podstawie którego jest rozpoznawane przez mikroprocesor.

Źródła przerw

Przerwania rozpoznawane przez mikroprocesor możemy podzielić na dwie grupy: przerwania zewnętrzne i przerwania wewnętrzne. Podział ten wynika z różnych źródeł przerw /rys. 1/.

Źródłami przerw zewnętrznych są zdarzenia związane z pracą innych niż mikroprocesor układów systemu mikroprocesorowego. Źródłami tymi mogą być urządzenia wejścia-wyjścia, układy kontroli pracy systemu mikroprocesorowego itp. Przerwania zewnętrzne mogą być zgłaszane do mikroprocesora poprzez linię NMI lub linię INTR. Linia NMI służy do generowania przerwania niemaskowalnego, które wykorzystywane jest zazwyczaj do sygnalizacji zdarzeń zagrażających poprawnej pracy systemu jak np.: zanik zasilania, błąd parytetu na magistrali lub błąd pamięci operacyjnej. Linią INTR jest linia zgłoszenia przerwania, którego przyjęcie uzależnione jest od stanu wskaźnika uaktywnienia przerw IF /ang interrupt enable flag/ słowa stanu mikroprocesora.

Przyjęcie przerwania zgłaszanego poprzez linię INTR powoduje przejście mikroprocesora do sekwencji potwierdzenia przyjęcia przerwania /INTA/, w trakcie której układ sterownika przerw /typowo układ 8259A/ musi przekazać mikroprocesorowi identyfikator przerwania. Sekwencja INTA będzie szczegółowo omówiona w dalszej części tego rozdziału.



Rys.1. Źródła przerw w mikroprocesorze 8086.

Osobną grupę stanowią przerwy wewnętrzne, których źródłami są zdarzenia w wewnętrznych układach mikroprocesora lub specjalne rozkazy z wykonywanego programu. Do przerw wewnętrznych należą:

- przerwanie sygnalizujące błąd operacji dzielenia,
- przerwanie sygnalizujące błąd nadmiaru,
- przerwanie programowe z wyborem identyfikatora,
- przerwanie dla pracy krokowej.

Przerwanie sygnalizujące błąd dzielenia generowane jest zawsze gdy wynik dzielenia /rozkazy DIV / $IDIV$ / przekracza dopuszczalny zakres wartości. W szczególności błąd ten pojawia się przy próbie dzielenia przez zero. Przerwanie błędu dzielenia posiada identyfikator równy 0.

Przerwanie sygnalizujące błąd nadmiaru generowane jest w wyniku wykonania rozkazu $INTO$ /ang. Interrupt on Overflow/ przy zapalonym wskaźniku nadmiaru OF /ang. Overflow Flag/ słowa stanu mikroprocesora. Przerwanie to posiada identyfikator równy 4.

Przerwanie programowe z wyborem identyfikatora generowane jest w wyniku wykonania rozkazu INT n, w którym argument n określa identyfikator przerwania. Przerwanie to może być wykorzystywane do testowania procedur obsługi przerwania zewnętrznych, do wywoływania procedur systemu operacyjnego itp.

Szczególnym przypadkiem rozkazu: przerwania programowego INT n jest rozkaz z identyfikatorem przerwania równym 3 /INT 3/, który posiada jednobajtowy kod rozkazu. Może on być zatem w prosty sposób używany jako "punkt zatrzymania" /ang breakpoint/ testowanego programu lub służyć do zastępowania na etapie uruchamiania programu, błędnego rozkazu innym rozkazem.

Przerwanie dla pracy krokowej generowane jest po każdym rozkazie, w trakcie wykonania którego wskaźnik pracy krokowej TF /ang. Trap Flag/ w słowie stanu mikroprocesora był zapalony. Należy w tym miejscu zaznaczyć, że istnieją przypadki, w których przerwanie zostanie przyjęte dopiero po zakończeniu następnego rozkazu. Przypadki te omówimy w dalszej części tego rozdziału.

Sekwencja przyjęcia przerwania

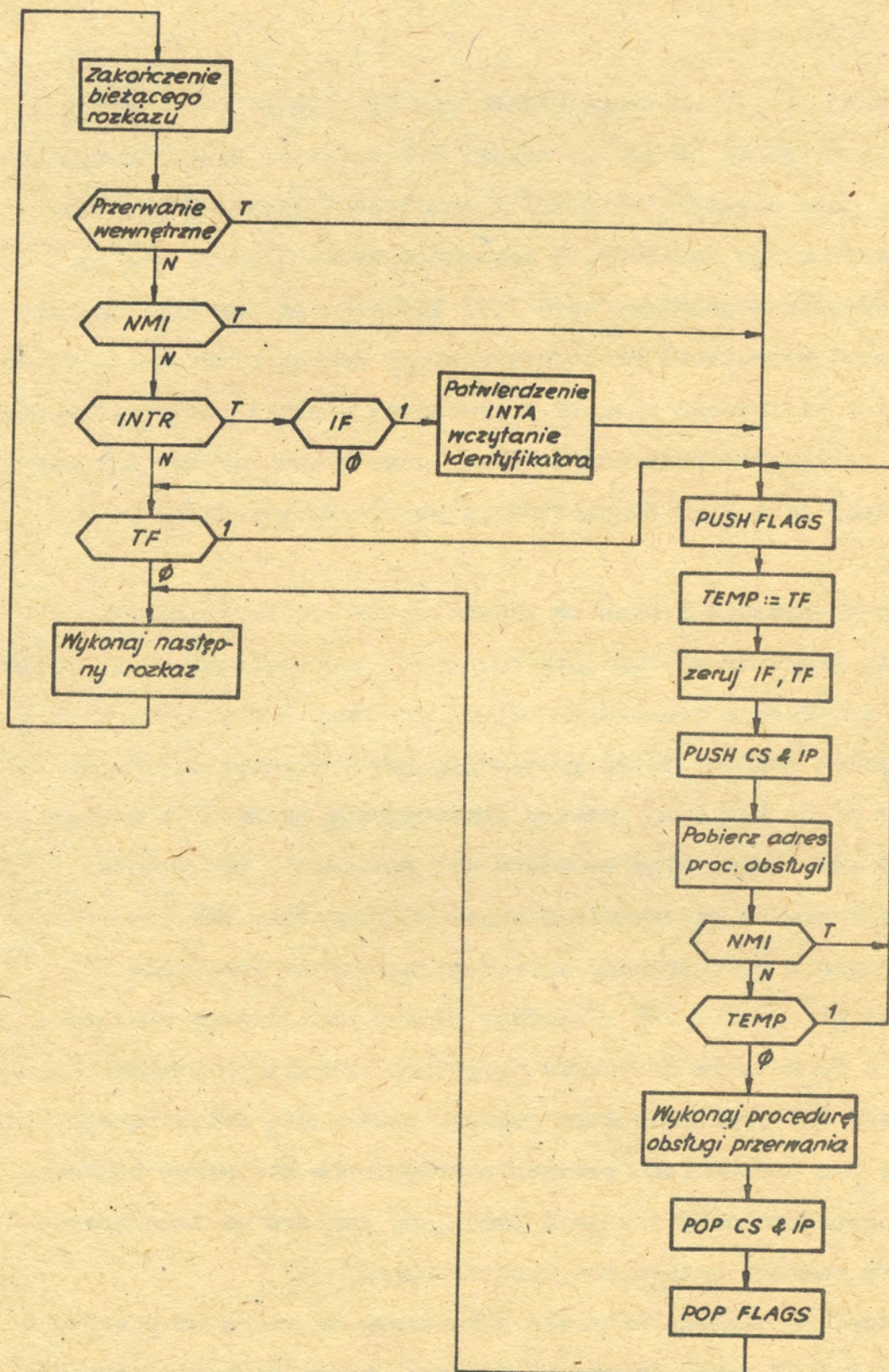
Zajmijmy się teraz przedstawieniem sekwencji rozpoznania, przyjęcia i obsługi przerwania przez mikroprocesor 8086. Sieć działań przedstawiająca tę sekwencję przedstawiona jest na rys. 2.

Zgłoszenia przerwania analizowane są każdorazowo po zakończeniu wykonywania kolejnego rozkazu. Istnieje jednak kilka wyjątków od tej reguły. Przede wszystkim, przedrostki /ang. prefix/ powtórzenia /ang. repeat/, blokady /ang. LOCK/ i zmiany segmentu /ang. segment override/ są traktowane jako część rozkazu i w związku z tym przerwanie nie są przyjmowane pomiędzy przedrostkiem a zasadniczym rozkazem. Ponadto, rozkazy przesłania /MOV i POP/ do rejestru

segmentowego są traktowane podobnie i przerwania nie są przyjmowane aż do zakończenia wykonania następnego po wyżej wymienionych rozkazu. Ma to szczególne znaczenie w przypadku zmiany rejestru segmentowego stanu /ang. Stack Segment register/ i wskaźnika stosu /ang. Stack Pointer/ określających położenie stosu w pamięci. Dla adresowania stosu uaktualnienie tych dwóch rejestrów musi być operacją niepodzielną, gdyż w przeciwnym przypadku mikroprocesor mógłby dokonać zapisu na stos umieszczony w przypadkowym obszarze pamięci. Należy zwrócić uwagę na fakt, że niepodzielność operacji uaktualnienia rejestru segmentowego i wskaźnika stosu będzie zapewniona tylko wówczas, gdy pierwszym w sekwencji będzie rozkaz uaktualnienia rejestru segmentowego, a drugim rozkaz uaktualnienia wskaźnika stosu. Istnieją również dwa przypadki, w których przerwania mogą być przyjmowane w trakcie wykonywania rozkazu. Są to rozkazy WAIT oraz powtarzane operacje łańcuchowe /ang. repeated string instructions/. W rozkazach tych przerwania mogą być przyjmowane po każdym elementarnym cyklu danego rozkazu.

Przedstawiona na rys. 2 kolejność w jakiej badane są poszczególne zgłoszenia przerwania określa priorytety tych przerwania. Najwyższy priorytet mają przerwania wewnętrzne z wyłączeniem przerwania dla pracy krokowej, następnie przerwanie MNI, przerwanie INTR, a najniższy priorytet posiada przerwanie dla pracy krokowej. W przypadku przerwania INTR jego przyjęcie uzależnione jest dodatkowo od stanu wskaźnika IF, a jego obsługa poprzedzona jest przejściem mikroprocesora do sekwencji potwierdzenia przyjęcia przerwania /INTA/, w trakcie której wczytywany jest identyfikator przyjętego przerwania.

Obsługa przerwania rozpoczyna się od złożenia na stosie słowa stanu mikroprocesora /ang. flags/, zapamiętania stanu wskaźnika pra-



Rys. 2. Sekwencja przyjęcia przerwania.

cy krokowej TF, wyzerowania wskaźników IF oraz TF słowa stanu, a następnie złożenia na stosie zawartości rejestru segmentowego kodu /ang. Code Segment Register/ i zawartości licznika rozkazów /ang. Instruction Pointer/. W następnym kroku licznik rozkazów /IP/ i rejestr segmentowy kodu /CS/ ładowane są wartościami odpowiadającymi początkowi procedury obsługi danego przerwania. Przed wykonaniem pierwszego rozkazu z procedury obsługi powtórnie badane jest zgłoszenie przerwania NMI, a przy braku takiego zgłoszenia, badany jest stan zapamiętanego wcześniej wskaźnika pracy krokowej.

Zapamiętywanie i badanie wskaźnika TF umożliwia śledzenie wykonywania procedur obsługi przerwania przy jednoczesnym wykonywaniu z pełną szybkością procedury obsługi przerwania dla pracy krokowej. Wyjątek stanowi tutaj przypadek, gdy w trakcie wykonania w trybie pracy krokowej rozkazu generującego przerwanie wewnętrzne /błąd dzielenia, błąd nadmiaru lub przerwanie programowe/ pojawi się zgłoszenie przerwania niemaskowalnego NMI. Wówczas przyjęcie przerwania wewnętrznego spowoduje zgaszenie wskaźnika TF, a przyjęcie przerwania NMI spowoduje zmianę pamiętanego wskaźnika TF z 1 na \emptyset . W takim przypadku procedura obsługi przerwania niemaskowalnego NMI oraz procedura obsługi przerwania wewnętrznego będą wykonane bez uprzedniego przyjęcia przerwania dla pracy krokowej, a tryb pracy krokowej zostanie wznowiony dopiero po zakończeniu procedury obsługi przerwania wewnętrznego.

Procedury obsługi przerwania zakończone są rozkazem powrotu z przerwania IRET /ang. Interrupt Return/, który powoduje pobranie ze stosu zawartości rejestru segmentowego kodu, zawartości licznika rozkazów oraz słowa stanu mikroprocesora odnawiając w ten stan mikroprocesora sprzed przyjęcia przerwania.

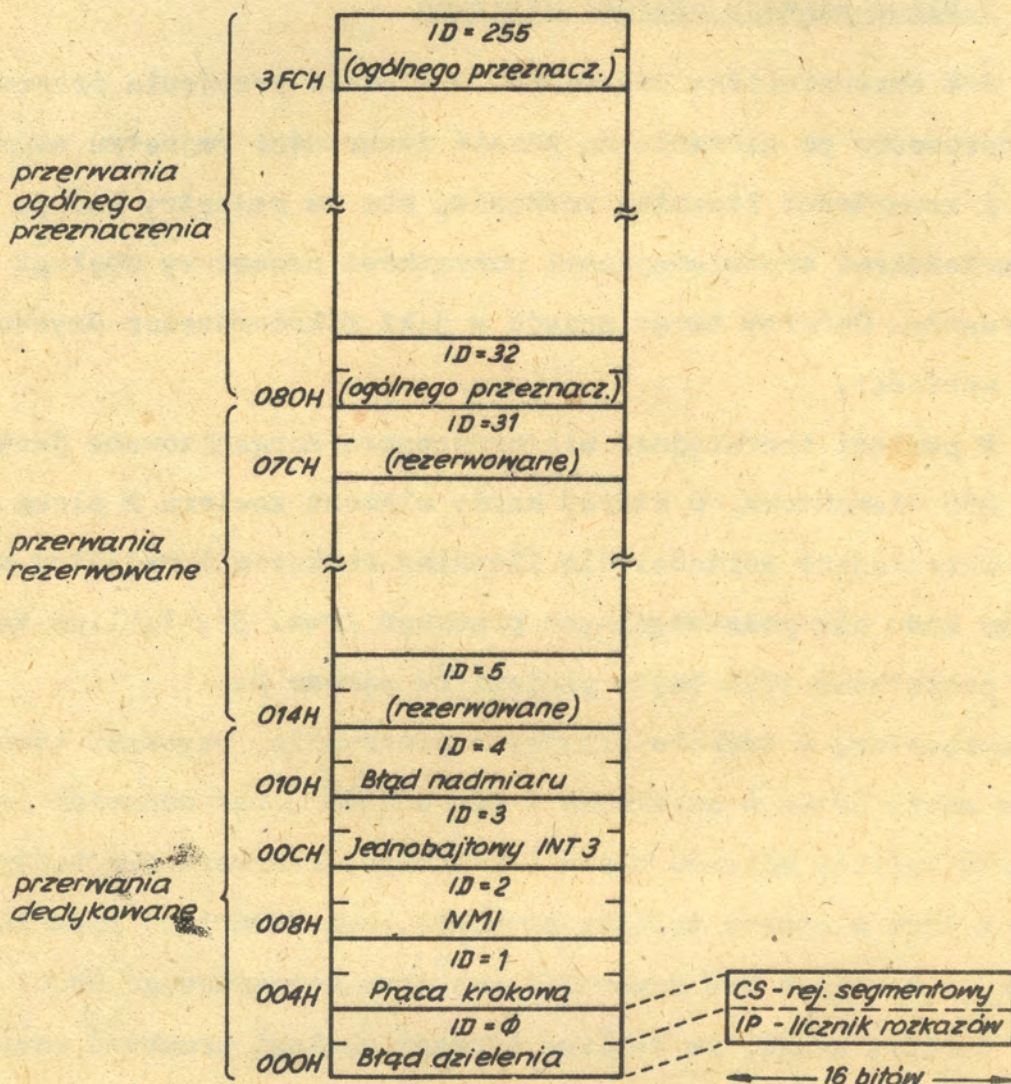
Tablica adresów obsługi przerwania

Jak wspomnieliśmy uprzednio, w trakcie przyjęcia przerwania, mikroprocesor po złożeniu na stosie zawartości rejestru segmentowego i zawartości licznika rozkazów, oba te rejestry ładuje nowymi wartościami odpowiadającymi początkowi procedury obsługi danego przerwania. Omówimy teraz sposób w jaki mikroprocesor uzyskuje te dwie wartości.

W pamięci operacyjnej mikroprocesora zorganizowana jest tablica 256-elementowa, w której każdy element zawiera 2 słowa 16-bitowe określające wartości dla licznika rozkazów i rejestru segmentowego kodu dla poszczególnych przerwania /rys. 3/. Tablica ta zajmuje początkowe 1024 bajty pamięci od adresu 0.

Mikroprocesor, w trakcie przyjęcia przerwania, uzyskany identyfikator mnoży przez 4 uzyskując w ten sposób adres odpowiedniego elementu tablicy adresów obsługi przerwania. Pierwsze dwa bajty wzięte z tego elementu tablicy stanowią nową zawartość licznika rozkazów, a następne dwa zawartość rejestru segmentowego kodu.

Zwróćmy uwagę, że tablica adresów obsługi przerwania rozpoczyna się zawsze od adresu 0 pamięci operacyjnej. Wynika z tego, że w systemie wielomikroprocesorowym te obszary pamięci poszczególnych mikroprocesorów powinny być pamięciami prywatnymi, a nie pamięcią systemową. W przeciwnym przypadku adresy obsługi przerwania dla poszczególnych mikroprocesorów byłyby identyczne.



Rys. 3. Tablica adresów obsługi przerwani.

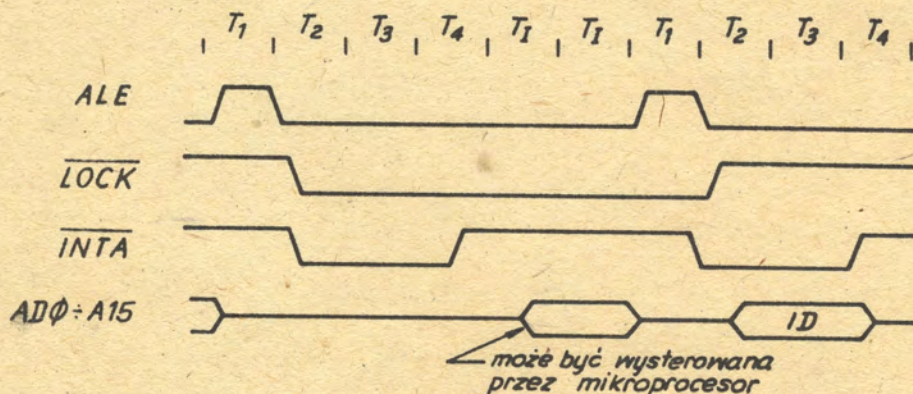
Sekwencja INTA potwierdzenia przyjęcia przerwania

Z przedstawionej na rys. 2 sieci działań wynika, że jedynym przerwaniem, które wymaga określenia przez zewnętrzne układy mikroprocesora identyfikatora przerwania jest przerwanie INTA.

Przyjęcie przerwania INTR powoduje przejście mikroprocesora do wykonania sekwencji potwierdzenia przyjęcia przerwania. Sekwen-

cja potwierdzenia przyjęcia przerwania składa się z dwóch kolejnych cykli INTA. Pierwszy cykl INTA służy do poinformowania układu sterownika przerwania o rozpoczęciu sekwencji potwierdzenia przyjęcia przerwania. W drugim, cyklu INTA mikroprocesor wczytuje z magistrali danych identyfikator przerwania przesłany ze sterownika przerwania.

Przebiegi sygnałów magistrali dla obu cykli INTA przedstawiono na rys. 4. Zwróćmy uwagę, że w czasie obu cykli INTA mikroprocesor nie wysyła żadnego adresu na szyny AD pozostawiając je w stanie wysokiej impedancji, a równocześnie sygnał ALE jest nadal generowany. Oznacza to, że rejestry adresu sterowane sygnałem ALE zostaną zapisane przypadkowymi adresami. W związku z tym, należy zwrócić uwagę, żeby żadne układy w systemie mikroprocesorowym nie uaktywniały swoich wyjść na magistralę jedynie na podstawie adresu, bez odpowiedniego sygnału sterującego /np. MRDC/. Przedstawione na rys. 4 przebiegi pokazują również sygnał $\overline{\text{LOCK}}$, który jest wewnętrznym sygnałem $\overline{\text{LOCK}}$ mikroprocesora.



Rys. 4. Sekwencja potwierdzenia przyjęcia przerwania.

Aktywny stan sygnału \overline{LOCK} pomiędzy kolejnymi cyklami INTA zabezpiecza mikroprocesor przed zwolnieniem magistrali w trakcie wykonywania sekwencji potwierdzenia przyjęcia przerwania. Sygnał ten, jak już wspomniano, jest wewnętrznym sygnałem mikroprocesora i spełnia swoją rolę niezależnie od trybu pracy mikroprocesora. W trybie minimum blokuje żądania HOLD, a w trybie maksimum blokuje żądania $\overline{RQ}/\overline{GT}$ oraz poprzez wyjście \overline{LOCK} , blokuje układy arbitrażu magistrali systemu wielomikroprocesorowego.

3. PROGRAMOWALNY STEROWNIK PRZERWAŃ TYPU 8259A

Układ sterownika przerwań typu 8259A jest zmodyfikowaną wersją sterownika 8259 stosowanego w systemach z mikroprocesorami 8080 i 8085. Układ 8259A został wzbogacony o nowe tryby pracy związane z pracą sterownika w systemie z mikroprocesorem 8086.

W niniejszym rozdziale przedstawimy tryby pracy sterownika 8259A, które stanowią rozszerzenia w stosunku do sterownika 8259.

W sterowniku 8259A jest pięć nowych trybów pracy:

- tryb pracy dla systemu 8086/8088 /ang. MCS-86/88 Mode/,
- tryb automatycznego końca obsługi /ang. Automatic End of Interrupt Mode - AEOI/,
- tryb zgłaszania przerwań poziomem /ang. Level Triggered Mode/,
- tryb specjalnego pełnego zagnieżdżenia przerwań /ang. Special Fully Nested Mode/,
- tryb pracy buforowanej /ang. Buffered Mode/.

Tryb pracy dla systemu 8086/8088 dostosowuje działanie sterownika 8259A do sekwencji potwierdzenia przyjęcia przerwania realizowanej w postaci dwóch cykli INTA. W czasie pierwszego cyklu INTA sterownik 8259A określa, które ze zgłoszeń będzie bieżąco obsługiwane i jeżeli zgłoszenie pochodzi z wejścia, do którego dołączony jest sterownik podrzędny /Slave/, na wyjścia kaskadowe CAS0-CAS2 podaje kod identyfikujący odpowiedni sterownik podrzędny. W drugim cyklu INTA, sterownik uaktywnia wyjścia danych na magistralę przekazując do mikroprocesora identyfikator bieżąco obsługiwanego przerwania. Należy tutaj zaznaczyć, że w przypadku systemu kaskadowego sterowników przerwań identyfikator przekazywany jest jedynie ze sterownika, do którego przyszło zgłoszenie przerwania. Wysyłany

identyfikator tworzony jest na podstawie 5-bitowego identyfikatora sterownika zdefiniowanym w drugim słowie inicjującym /ICW2/ i 3-bitowego numeru przerwania danego sterownika.

Automatyczny koniec obsługi przerwania /AEOI/ jest trybem pracy sterownika, w którym odpowiadający odebranemu przerwaniu bit w rejestrze bieżącej obsługi /ang. In Service Register/ jest gaszony natychmiast po zakończeniu sekwencji potwierdzenia przyjęcia przerwania. Tryb ten eliminuje konieczność wysyłania komendy zakończenia obsługi przerwania /ang. End Of Interrupt/ na końcu każdej procedury obsługi przerwania upraszczając w ten sposób kod tej procedury.

Tryb zgłaszania przerwania poziomem dotyczy zasady działania układów wejściowych dla wejść zgłoszeń przerwania. W sterowniku 8259 zgłoszenie przerwania było przyjmowane tylko wtedy, gdy na wejściu pojawiło się zbocze narastające i poziom wysoki był utrzymany aż do zakończenia pierwszego cyklu INTA. W sterowniku 8259A można za pomocą pierwszego słowa inicjującego /ICW1/ zmienić działanie układów wejściowych tak, żeby nie było wymagane narastające zbocze dla zgłoszenia przerwania.

Następnym z wymienionych nowych trybów pracy jest tryb specjalnego pełnego zagnieżdżenia przerwania /SFNM/, który przeznaczony jest dla sterownika nadrzędnego /Master/ w systemie kaskadowym. W normalnym trybie pełnego zagnieżdżenia zgłoszenia przerwania były przyjmowane jedynie z wejść o wyższym priorytecie. Tryb ten dotyczy również sterownika nadrzędnego, a zatem, gdy w trakcie obsługi przerwania ze sterownika podrzędnego przyszło nowe zgłoszenie o wyższym priorytecie w ramach tego samego sterownika, nie było ono przyjmowane, gdyż sterownik nadrzędny nie mógł przyjąć drugiego przerwania na tym samym wejściu. Przerwanie to było przyjmowane dopiero po

zakończeniu obsługi przerwania o niższym priorytecie. Tryb specjalnego pełnego zagnieżdżenia przerw powoduje, że sterownik w trakcie obsługi jednego przerwania może przyjmować nie tylko przerwania o wyższym priorytecie, ale również przerwania o tym samym priorytecie co obecnie obsługiwane. W ten sposób, wykorzystując omawiany tryb pracy w sterowniku nadrzędnym, możliwe jest zrealizowanie pełnego systemu zagnieżdżonych przerw przy kaskadowym połączeniu sterowników.

Ostatni z omawianych trybów pracy, tryb pracy buforowanej przeznaczony jest dla systemów, w których sterowniki podrzędne znajdują się na innych niż sterownik nadrzędny pakietach i w związku z tym muszą być łączone z magistralą systemu poprzez bufory. W sterowniku 8259A nóżka 16 układu oprócz funkcji \overline{SP} /ang. Slave Program/ definiującej czy sterownik jest nadrzędnym czy podrzędnym, otrzymała drugą funkcję \overline{EN} /ang. Enable Buffer/ umożliwiającą sterowanie buforem łączącym sterownik z magistralą. Tak więc w momencie zaprogramowania sterownika na tryb pracy buforowanej nóżka 16 $\overline{SP}/\overline{EN}$ staje się wyjściem, na którym pojawia się stan niski ilekroć sterownik uaktywnia wyjścia danych na magistralę w celu przesłania identyfikatora w cyklu INTA. Zmiana funkcji nóżki $\overline{SP}/\overline{EN}$ powoduje, że określenie czy sterownik ma pracować jako nadrzędny czy jako podrzędny musi być dokonane na drodze programowej. Określenia tego dokonuje się łącznie z wyborem pracy buforowanej w czwartym słowie inicjującym /ICW4/.

4. PODSUMOWANIE

W artykule tym przedstawiono organizację systemu przerwań w mikroprocesorze 8086. W opisach poszczególnych elementów tego systemu zwracano uwagę na przypadki odstępstwa od reguł charakteryzujących działanie systemu. Przedstawiono również mechanizmy systemu przerwań uwzględniające pracę w środowisku wielomikroprocesorowym. Opisane zostały również nowe tryby pracy sterownika przerwań typu 8259A.

Niestety ograniczona objętość tego artykułu nie pozwoliła zilustrować przedstawionego tematu przykładami różnych konfiguracji połączeń mikroprocesora 8086 ze sterownikami przerwań 8259A w systemach jedno- i wielomikroprocesorowych.

5. BIBLIOGRAFIA

1. The 8086 Family User's Manual, Intel Corporation 1979.
2. George Alexy, 8086 System Design, Application Note AP-67, Intel Corporation 1979.
3. Designing 8086, 8088, 8089 Multiprocessing Systems with the 8289 Bus Arbiter, Application Note AP-51, Intel Corporation 1979.
4. Robin Jigour, Using the 8259A Programmable Interrupt Controller, Application Note AP-59, Intel Corporation 1979.
5. Joe Barthmaier, Intel MULTIBUS Interfacing, Application Note AP-28A, Intel Corporation 1979.

VI Szkoła mikroprocesorowa CSI/PTI
Łódź, grudzień 1984

KOPROCESOR ARYTMETYCZNY
INTEL 8087

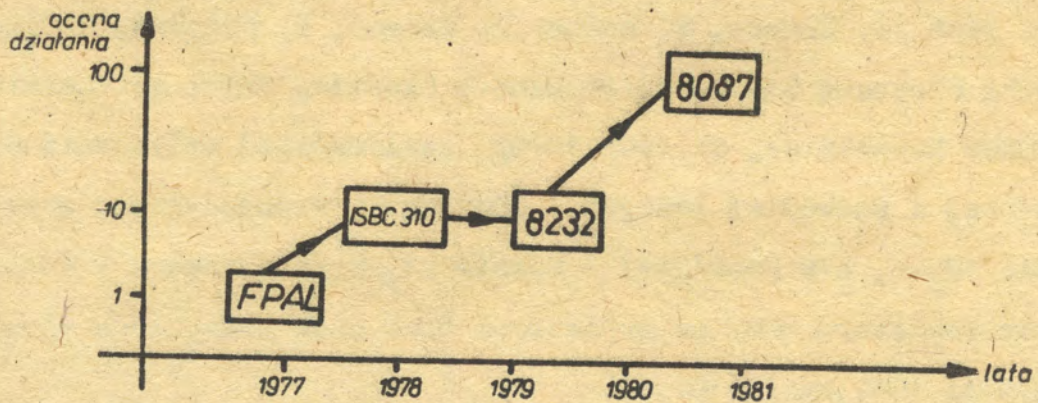
mgr inż. Mieczysław Drabowski
mgr Leszek Szklarczyk
Krakowska Fabryka Aparatów Pomiarowych
MERA-KFAP
ul. Zapolskiej 38
30-126 Kraków
tel. 374-222 w.493

STRESZCZENIE

W artykule przedstawiono koprocesor arytmetyczny Intel 8087, który jest przeznaczony do współpracy z mikroprocesorami centralnymi Intel 8086 i 8088. Zastosowanie tego układu pozwala na poszerzenie listy rozkazów wymienionych mikroprocesorów centralnych, a tym samym ich specjalizację w kierunku obliczeń numerycznych. Architektura wewnętrzna koprocesora arytmetycznego, a w szczególności sześćdziesięcioczworobitowa wewnętrzna magistrala danych, grupy osiemdziesięciobitowych rejestrów oraz sześćdziesięcioośmiobitowa jednostka arytmetyczno-logiczna pozwala na uzyskanie dużej precyzji obliczeń i znaczne skrócenie czasu ich wykonywania. Koprocesor arytmetyczny pracuje równolegle z mikroprocesorem centralnym i wykonuje nakazane przez niego instrukcje. W artykule szczególną uwagę poświęcono zasadom współpracy mikroprocesora centralnego z koprocesorem arytmetycznym oraz scharakteryzowaniu jego możliwości obliczeniowych.

1. Wstęp

Koprocesor /coprocessor/ Intel 8087 należy do rodziny procesorów iAPX86, 88 i jest przeznaczony do realizacji operacji arytmetycznych. Jest on częścią procesorów typu iAPX86/88//20, iAPX86/88//21. Koprocesor 8087 jest procesorem numerycznym /NDP - Numeric Data Processor/. Rysunek 1 pokazuje rozwój środków do obliczeń numerycznych w firmie Intel.



Rys. 1. Ewolucja procesorów numerycznych

W roku 1977 firma Intel przyjmuje standard obliczeń zmiennoprzecinkowych /floating point format/. Pierwszą implementacją tego standardu był FPAL /Floating Point Arithmetic Library/ opracowany dla procesorów 8080/8085. Format danych rzeczywistych jest 32-bitowy, a czas mnożenia 1,5 ms /dla zegara 1,6 MHz/. Następnym produktem jest pakiet iSBC 310 redukujący czas mnożenia do około 100 μ s. W roku 1979 ukazuje się procesor arytmetyczny 8232, który operuje także w podwójnej precyzji i jest pojedynczym układem scalonym realizującym operacje zmiennoprzecinkowe dla procesorów 8080/8085. Z punktu widzenia procesora CPU procesor arytmetyczny jest modulem zewnętrznym zawierającym dwie adresowe komórki tj. wierzchołek stosu oraz rejestr stanu i poleceń. Współpraca CPU z procesorem arytmetycznym odbywa się następująco:

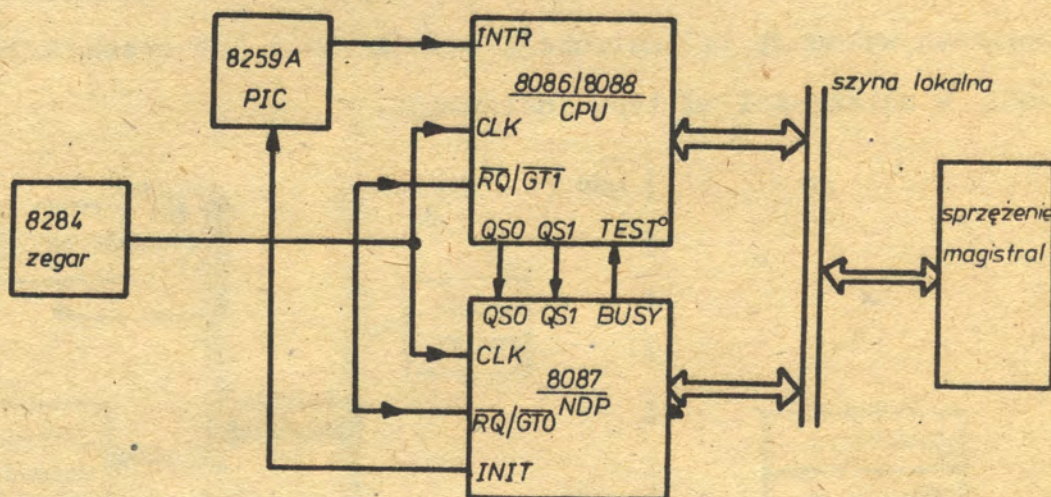
- CPU wpisuje dane na stos procesora arytmetycznego,
- CPU wysyła polecenie wykonania operacji na tych danych, po czym odczytuje wynik ze stosu lub wysyła następne polecenie.

Procesor 8232 mnoży w pojedynczej precyzji w czasie $100\mu\text{s}$, a w podwójnej w czasie $875\mu\text{s}$ /dla zegara 2 MHz/. W roku 1980 pojawia się koprocesor 8087 będący sprzętową implementacją w pojedynczym układzie scalonym arytmetyki zmiennoprzecinkowej opracowanej przez komitet IEEE /J. Coonen, W. Kahan, J. Palmer, T. Pittman, D. Stevenson "A Proposed Standard for Binary Floating Point Arithmetic" ACM SIGNUM Newsletter, October 1979/, zawierającej obliczenia dla pojedynczej i podwójnej precyzji. 8087 mnoży w pojedynczej precyzji w czasie $19\mu\text{s}$, a w podwójnej w czasie $27\mu\text{s}$ /dla zegara 5 MHz/. Używanie emulatora 8087 na procesorze 8086 daje czasy tych mnożeń odpowiednio $1600\mu\text{s}$ i $2100\mu\text{s}$.

2. Programowe i sprzętowe połączenie 8087 z CPU

Dla programisty połączenie 8086 /8088/ z 8087 daje jeden wspólny procesor. 8087 dodaje nowe typy danych, nowe rejestry i nowe instrukcje. Dla języków programowania powstaje interprocesor o automatycznie skoordynowanych funkcjach. Koprocesor 8087 pracuje równolegle z procesorem głównym, pobierając z pamięci ten sam ciąg rozkazów, lecz wykonuje tylko te rozkazy, które są dla niego przeznaczone. Sposób połączenia koprocesora do procesora przedstawiono na rys. 2. Mikroprocesor 8086/88/ pracuje w trybie maksymalnym generując linie kolejki statusowej /QS0 i QS1/ synchronizujące pobieranie i dekodowanie instrukcji NDP z CPU. Sygnał BUSY informuje CPU o wykonywaniu operacji przez NDP; instrukcja WAIT CPU testuje ten sygnał. NDP może generować przerwanie do CPU poprzez kontroler przerwania 8259A. Procesor arytmetyczny może przejąć sterowanie

magistralą lokalną dla ładowania i pamiętania danych w pamięci. Służą do tego wewnętrzne układy arbitrażu sterowane linią RQ/GT. Oba procesory używają tego samego zegara, kontrolera szyn i arbitra magistrali.

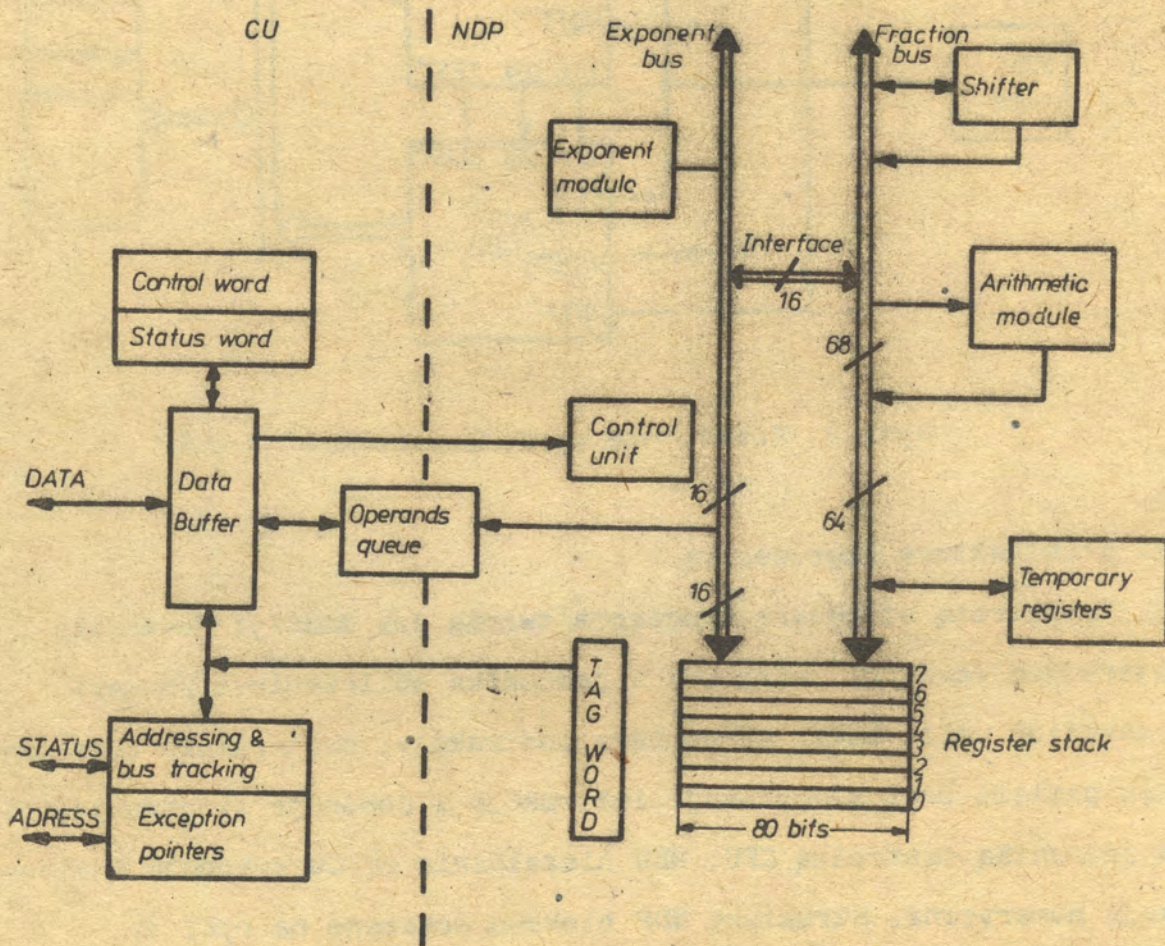


Rys. 2. Połączenie sprzętowe 8086/88 i 8087

3. Architektura koprocatora

Wewnętrzna struktura procesora tworzą dwa moduły: jednostka sterująca /control unit; CU/ i jednostka obliczeniowa /numeric exeaction unit; NEU/. CU pobiera instrukcje, czyta/pisze operandy z/w pamięci oraz klasyfikuje instrukcje i dokonuje synchronizacji z jednostką centralną CPU. NEU niezależnie od CU wykonuje instrukcje numeryczne. Strukturę NDP blokowo pokazano na rys. 3. Instrukcje 8087 są pomieszczone w strumieniu instrukcji pobieranych przez CPU. CU NDP otrzymuje od CPU informację statusową określającą kiedy następuje pobieranie /fetch/ instrukcji. Kiedy instrukcja jest dostępna /bajt lub słowo/ na szynie lokalnej CU równolegle z CPU pobiera tę instrukcję. CU posiada podobnie jak CPU kolejkę instrukcji. W rezultacie synchronizacji z CPU oba procesory

pobierają i dekodują strumień instrukcji równolegle. Pierwsze pięć bitów we wszystkich instrukcjach 8087 jest identycznych; te bity wyznaczają ujście /escape, ESC/ instrukcji do koprocatora. CU ignoruje wszystkie instrukcje nie mające tych bitów, instrukcje te są przeznaczone dla CPU. Jeśli CU wykryje w instrukcji kod ujścia ESC, wówczas wykonuje tę instrukcję samodzielnie lub przekazuje ją do NEU, w zależności od typu instrukcji.



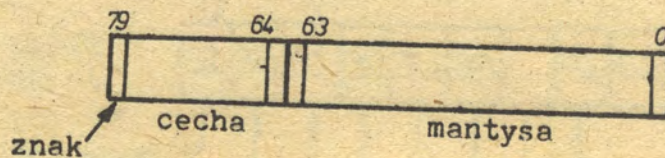
Rys. 3. Schemat blokowy 8087

CPU odróżnia pomiędzy ESC instrukcjami te, które odwołują się do pamięci i te które nie odwołują się do niej. Jeśli instrukcja odwołuje się do operanda w pamięci CPU oblicza adres operanda i wykonuje "sztuczne czytanie" /"dummy read"/ słowa pod tym adresem.

Jest to normalny cykl czytania, przy czym CPU ignoruje otrzymane dane. Jeśli ESC instrukcja nie odwołuje się do pamięci, CPU wykonuje kolejną instrukcję.

Instrukcje 8087 mogą żądać ładowania operanda z pamięci lub żądać pamiętania operanda w pamięci lub nie odwoływać się do pamięci. W pierwszych dwóch przypadkach CU wykorzystuje cykl "sztucznego czytania" inicjalizowanego przez CPU. CU zapamiętuje adres operanda i jeśli jest to operacja ładowania adresuje dodatkowe słowa a jeśli jest to instrukcja pamiętania CU zapamiętuje adres, ignoruje dane słowa występującego w "sztucznym" czytaniu, żąda dostępu do szyn i pisze operanda do zapamiętanego adresu.

NEU wykonuje wszystkie instrukcje na rejestrach stosu. Każdy z ośmiu rejestrów ma 80 bitów szerokości i każdy jest podzielony na pola pokazane na rys. 4.



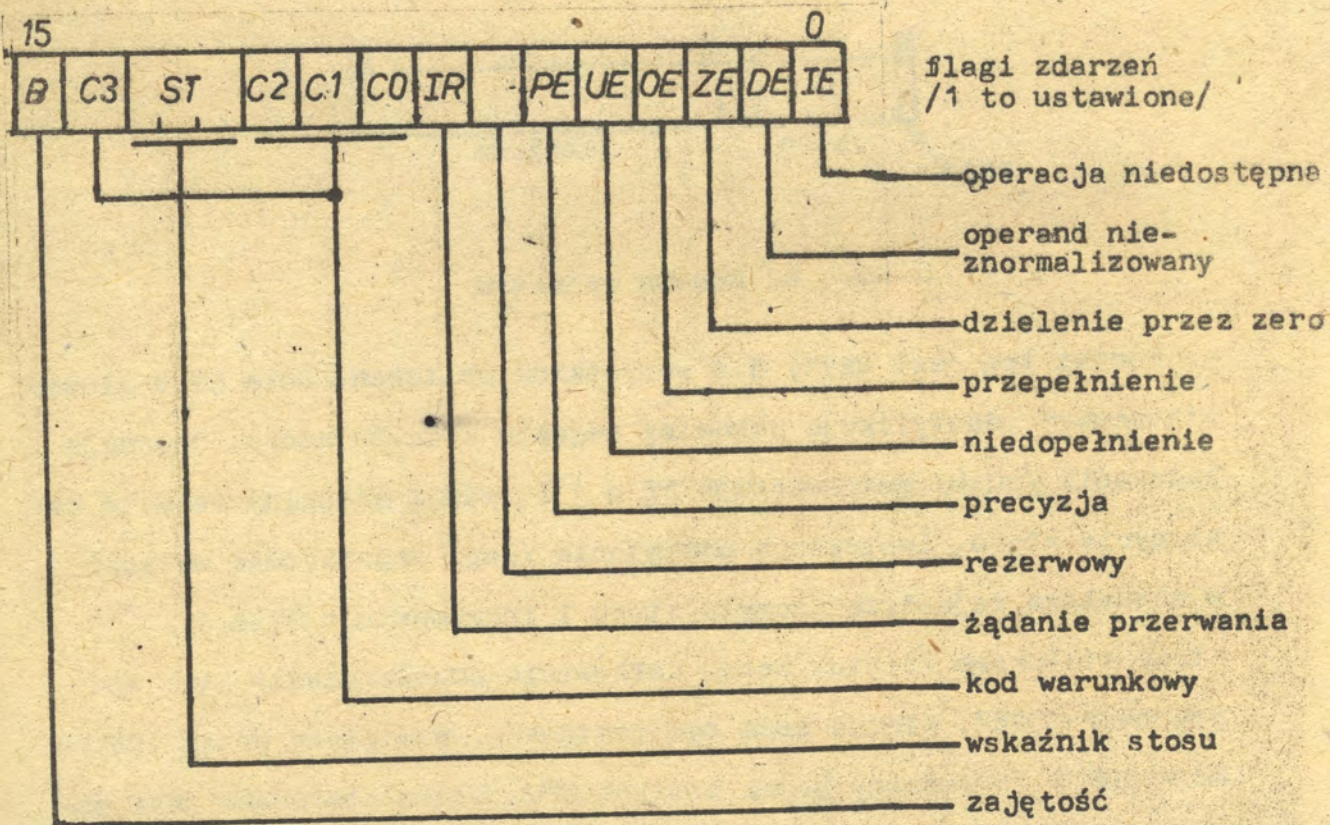
Rys. 4. Budowa rejestru

Format ten jest użyty dla wszystkich obliczeń. Pole ST w słowie statusowym identyfikuje aktualny rejestr szczytu stosu. Operacja ładowania /push/ dekrementuje ST o 1 i ładuje aktualny rejestr na szczycie stosu. Instrukcja pamiętania /pop/ zapamiętuje wartość w aktualnym rejestrze szczytu stosu i inkrementuje ST o 1. Słowo statusowe /Status Word/ całkowicie odzwierciedla stan ko-procesora 8087. Status może być testowany, pamiętany przez instrukcje NDP i sprawdzany przez program CPU. Słowo statusowe jest podzielone na następujące pola /rys. 5/: "B" wskazuje zajętość tzn. gdy NDP wykonuje instrukcje B=1 i B=0 jeśli NDP jest bezczynne.

Niektóre instrukcje 8087 /np. komparacja/ ustawiają w rezultacie kod warunkowy /bity 14 i 10-8 w słowie statusowym/. Kod warunkowy jest użyty przede wszystkim dla warunkowych rozgałęzień. Bity 13-1 w słowie statusowym wskazują, który rejestr jest aktualnie na szczycie stosu /ST/. Tak więc

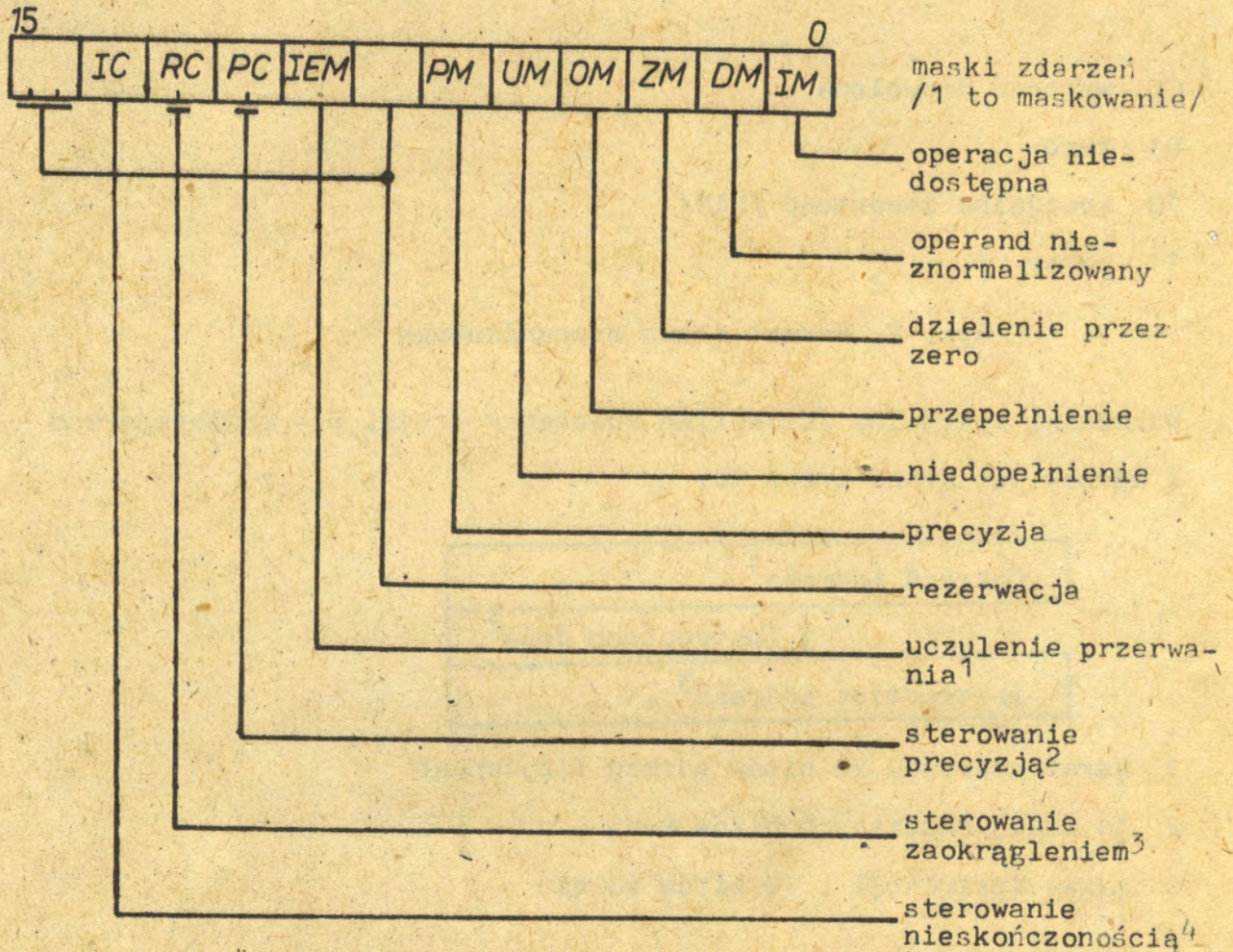
- 000 to rejestr 0 jest na szczycie stosu,
- 001 to rejestr 1 jest na szczycie stosu,
- ...
- 111 to rejestr 7 jest na szczycie stosu.

Jeśli ST=000 i wykonujemy operację "push" dekrementując ST otrzymujemy ST=111, podobnie "pop" stosu z ST=111 daje ST=000. Bit 7 jest to pole żądania przerwania. NDP ustawia to pole świadczące o oczekiwaniu przerwania do CPU. Bity 5-0 są to flagi ustawiane dla wskazania wyjątkowych stanów obliczeń NEU.



Rys. 5. Format słowa statusowego

Dla różnych aplikacji NDP może być programowane przez ładowanie słowa kontrolnego /Control Word/ sterującego pracą koprocatora. Rys. 6 pokazuje format tego słowa.



2 precyzja sterowana: 00-24 bity, 01-rezerwacja, 10-53 bity, 11-64 bity,

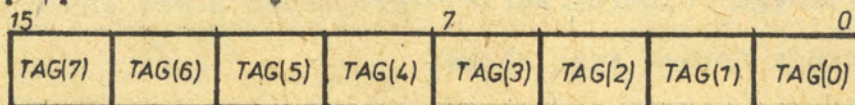
3 zaokrąglenie sterowane: 00-dokładne, 01-w kierunku - , 10-w kierunku + , 11-obcięcie

4 nieskończoność sterowania: 0-zespolona, 1-rzeczywista

1 przerwanie: 0-uczulone, 1-maskowane.

Rys. 6. Format słowa sterującego /Control Word/

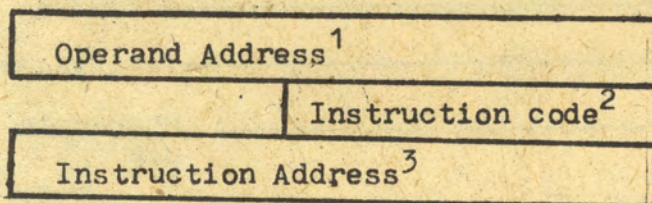
Słowo znacznikowe /Tag Word/ pokazuje zawartość każdego rejestru wg rys. 7.



- 00 wartość dozwolona
- 01 zero
- 10 specjalna zawartość /NAN/
- 11 pusty

Rys. 7. Format słowa znacznikowego

Wskaźniki specjalne /Exception Pointers/ - rys. 8 - zawierają kod i adresy pamiętane przez CU.

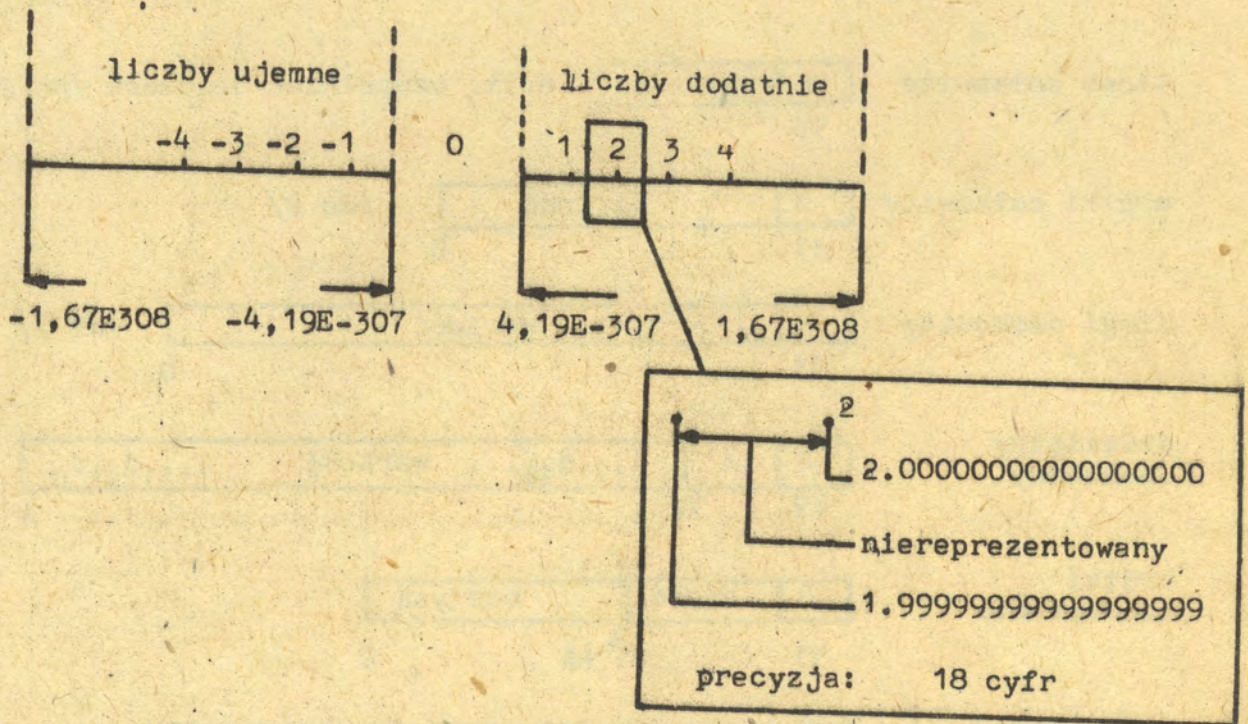


- 1 adres operand, 20 bitów adresu fizycznego
- 2 11 mniej znaczących bitów kodu
- 3 adres instrukcji, 20 bitów adresu

Rys. 8. Format wskaźników specjalnych

4. Podstawy obliczeń numerycznych w 8087

Rys. 9 pokazuje system numeryczny realizowany w 8087. Na uwagę zasługuje przedział liczbowy dostępny na koprocessorze $1,67 \cdot 10^{308}$ do $1,67 \cdot 10^{308}$ w porównaniu z zakresem IBM 370 $\pm 0,72 \cdot 10^{76}$. Wewnętrzne obliczenia są prowadzone w rozszerzonym przedziale $\pm 3,4 \cdot E-4932$ do $\pm 1,2 \cdot E-4932$ przy precyzji 19 cyfr dziesiętnych. Przy pobieraniu danych z pamięci lub przesłaniu wyników do pamięci następuje automatycznie zmiana formatu liczb.

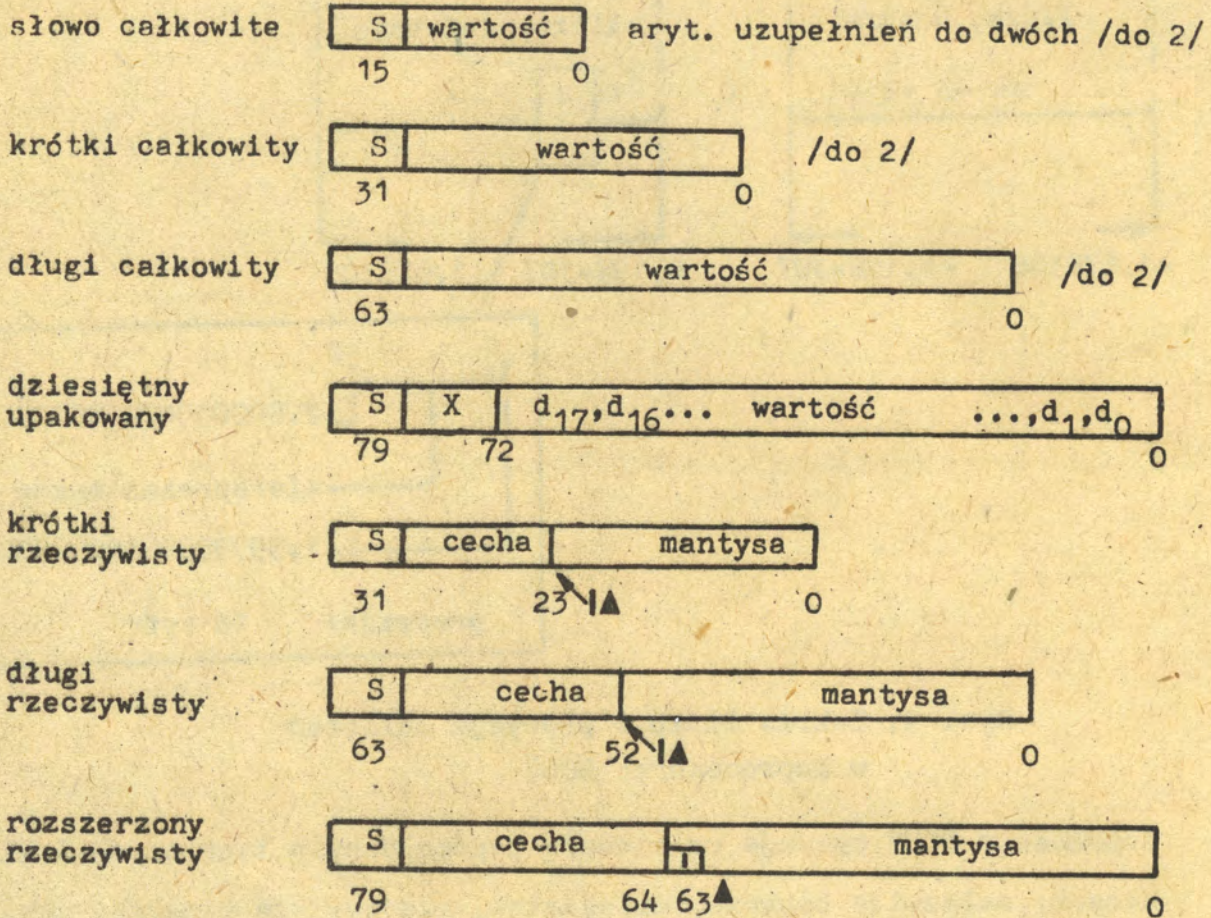


Rys. 9. Zakres liczb i precyzja obliczeń w koprocesorze 8087

Koprocesor 8087 operuje na siedmiu typach danych tworzących 3 klasy danych: całkowite binarne, dziesiętne, rzeczywiste binarne. Rys. 10 pokazuje format tych danych, zaś rys. 11 tabelę rozpisania liczby 178,125 na postać rzeczywistą reprezentowaną w koprocesorze.

zapis	wartość
zwyczajny	178,125
dziesiętny	1,78125E2
binarny	1,0110010001E111
binarny /po przesunięciu cechy/	1,0110010001E10000110
format 8087	Znak cecha mantysa
/krótki rzeczywisty/	0 10000110 01100100000000000000

Rys. 10. Notacja liczby rzeczywistej



Uwaga: S znak /0=dodatni, 1=ujemny/
 d_n cyfra dziesiętna /2 na bajt/
X bity nieznaczące
▲ pozycja przecinka binarnego
| bit znaczący

przesunięcie cechy: krótki o 127, długi 1023, rozszerzony 16383

Rys. 11. Formaty danych

5. Instrukcja synchronizacji

Instrukcja CPU 8086/88 WAIT służy do synchronizacji współpracy CPU z NDP. Podczas wykonywania instrukcji 8087 uaktywnia linię BUSY. Ten sygnał jest połączony z wejściem TEST⁰ CPU. NDP ignoruje instrukcje WAIT, a CPU wykonuje ją. CPU interpretuje instrukcje WAIT

jak "czekaj gdy TEST⁰ jest aktywny". CPU testuje wejście TEST⁰ przez 5. zegarów, jeśli wejście to jest nieaktywne, następuje wykonanie następnej instrukcji; jeśli TEST⁰ jest aktywny CPU testuje to wejście znowu. Instrukcje następujące po WAIT są dekodowane jednocześnie przez oba procesory. NDP nie może pobrać następnego rozkazu przeznaczonego dla NEU przed zakończeniem wykonywania poprzedniego. Jeśli pojawią się kolejno instrukcje:

FMUL; mnożenie

FDIU; dzielenie

to assembler produkuje cztery instrukcje maszynowe

WAIT

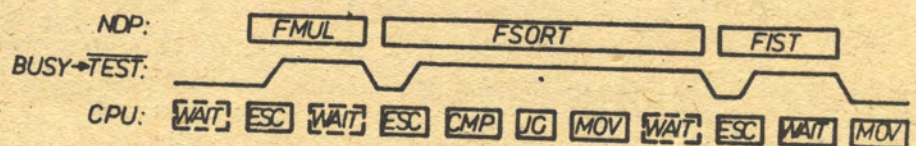
FMUL

WAIT

FDIU

Mnożenie musi być wykonane całkowicie aż CPU i CU 8087 będzie dekodować dzielenie. Rys. 11 pokazuje wykonanie programu przez 1APX86/20.

```
FMUL
FSQRT
CMP ALPHA,100
JG CON
MOV ALPHA,100
CON: FIST BETA
FWAIT
MOV AX,BETA
```



Rys.12. Wykonanie przykładowego programu.

6. Instrukcje

Instrukcje 8087 wykonują operacje na zawartość pamięci /lub rejestrów stosu/ i wierzchołka stosu.

Wyróżniamy następujące grupy instrukcji:

przesłania: load, store, exchange

arytmetyczne: add, subtract, multiply, divide, square root,
scale, etc.

komparacja: test, examine, compare

funkcje : tang, arctang, 2^X-1 , $Y \cdot \log_2/X+1/$, $Y \cdot \log_2/X/$

stałe : 0,1, ,log2, la2, ln10, $\log_2 10$, $\log_2 e$

sterujące : initialize, operacje na słowie sterującym, statusowym,
uczulenie przerwań etc.

VI Szkoła Mikroprocesorowa CSI/PPI
Łódź, grudzień 1984

MIKROPROCESOR WEJSCIA/WYJŚCIA
INTEL 8089

mgr inż. Piotr Krzyżagórski
Instytut Automatyki
Politechnika Poznańska
ul. Piotrowo 3A
60-965 Poznań, tel. 782-370

1. WSTĘP

Pomijając bardzo wczesny okres rozwoju komputerów w ich architekturze zawsze wyróżniano specjalizowany moduł we/wy, przeznaczony do obsługi szybkiej transmisji pomiędzy urządzeniami zewnętrznymi a pamięcią operacyjną. Moduł ten, działając w sposób autonomiczny w stosunku do procesora centralnego, zwalnia go od bieżącej obsługi transmisji umożliwiając pracę współbieżną, co prowadzi do wzrostu efektywności działania systemu. Ponadto, dzięki specjalizacji, moduł ten może obsługiwać transmisje znacznie szybsze niż procesor centralny.

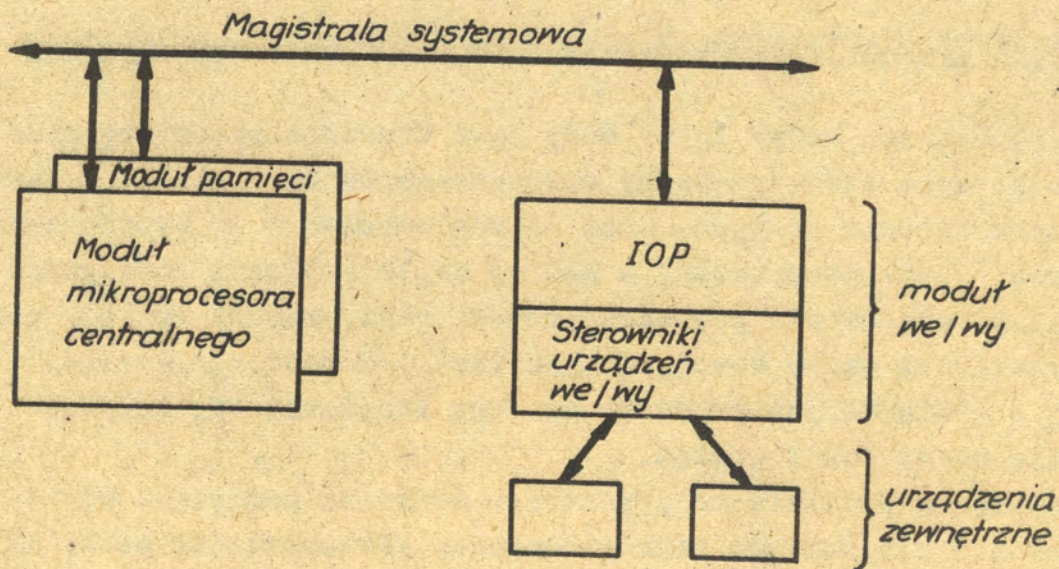
Również w systemach mikrokomputerowych celowe jest wyróżnienie takiego modułu. Z tego względu w skład poszczególnych rodzin układów mikroprocesorowych wchodzi odpowiednie układy umożliwiające jego konstrukcję. Wśród tych układów podstawowe znaczenie mają układy spełniające funkcje kanałów bezpośredniego dostępu (ang. Direct Memory Access - DMA). Przykładowo w rodzinie mikroprocesorowej Intel 8080/8085 funkcje takie spełniają układy 8237 i 8257. Przebieg sterowania transmisją w systemie mikrokomputerowym wyposażonym w taki układ jest następujący. Mikroprocesor centralny programuje

kanal DMA i ewentualnie inne układy programowalne wchodzące w skład modułu we/wy i dołączone do kanału DMA, jak na przykład sterowniki urządzeń zewnętrznych, określając adres bufora w pamięci oraz niezbędne parametry transmisji. Następnie, po wydaniu przez mikroprocesor centralny komendy rozpoczęcia transmisji sterowanie transmisją przejmuje kanał DMA. Pośredniczy on w przesyłaniu danych z(do) urządzenia zewnętrznego poprzez jego sterownik do(z) pamięci operacyjnej mikrokomputera bez angażowania mikroprocesora centralnego, który, teoretycznie, może wykonywać w tym czasie zadania współbieżne. Zakończenie transmisji lub stwierdzenie przez kanał DMA błędów transmisji sygnalizowane jest mikroprocesorowi centralnemu przez przerwanie. Działanie kanału DMA w trakcie transmisji polega na generowaniu kolejnych adresów pamięci i sterowaniu przepływem danych. Należy podkreślić, że w czasie przesyłania każdego bajtu do/z pamięci operacyjnej kanał DMA przejmuje pełną i wyłączną kontrolę nad magistralą mikrokomputera, co oczywiście powoduje zawieszenie działania mikroprocesora centralnego.

Jak wiadomo, kolejny etap rozwoju systemów mikrokomputerowych jest związany z systemami wielomikroprocesorowymi. Również w systemach wielomikroprocesorowych celowe jest wyróżnienie specjalizowanego modułu, przeznaczonego do obsługi transmisji. Jednakże bezpośrednio zastosowanie przedstawionego powyżej schematu sterowania przebiegiem transmisji, realizowane przez układy kanałów DMA, jest niecelowe, gdyż zbyt mocno ogranicza możliwości równoległej pracy mikroprocesorów w systemie wielomikroprocesorowym oraz poważnie angażuje mikroprocesory centralne w obsługę sterowników urządzeń zewnętrznych. Z tego względu, do rodziny układów mikroprocesorowych 16-bitowych, z wbudowanymi mechanizmami pracy wielomikroprocesorowej, wprowadzono układy tak zwanych mikroprocesorów wejścia/wyjścia (ang. Input/Output Processor - IOP), które stanowią podstawowe układy do budowy modułów we/wy. Przykładowo w rodzinie mikroprocesorowej Intel 8086 rolę taką pełni układ 8089.

Podstawową cechą mikroprocesora we/wy jest to, że łączy on w sobie funkcje mikroprocesorów centralnych i kanałów DMA, dzięki czemu może dokonywać szybkich i "inteligentnych" transmisji, w szczególności w systemach wielomikroprocesorowych. Ogólną architekturę takiego systemu oraz wewnętrzną strukturę modułu we/wy przedstawiono na rys.1.

Rozszerzenie funkcji mikroprocesora we/wy w stosunku do kanału DMA o zdolność do przetwarzania danych wynika z dwóch podstawo-



Rys.1. System wielomikroprocesorowy z autonomicznym modułem we/wy.

wych faktów. Po pierwsze z wymagania, aby IOP był zdolny do obsługi szerokiej klasy sterowników urządzeń zewnętrznych, która polega głównie na ich programowaniu, odczycie ich stanów, modyfikacji wybranych rejestrów wewnętrznych oraz wykrywaniu i obsłudze sygnalizowanych błędów. Po drugie, aby mikroprocesor we/wy był zdolny do realizacji wstępnego przetwarzania danych w czasie transmisji przed ich przekazaniem do przetwarzania przez mikroprocesor centralny. Taki sposób organizacji przetwarzania transmitowanych bajtów pozwala na istotny wzrost efektywności całego systemu.

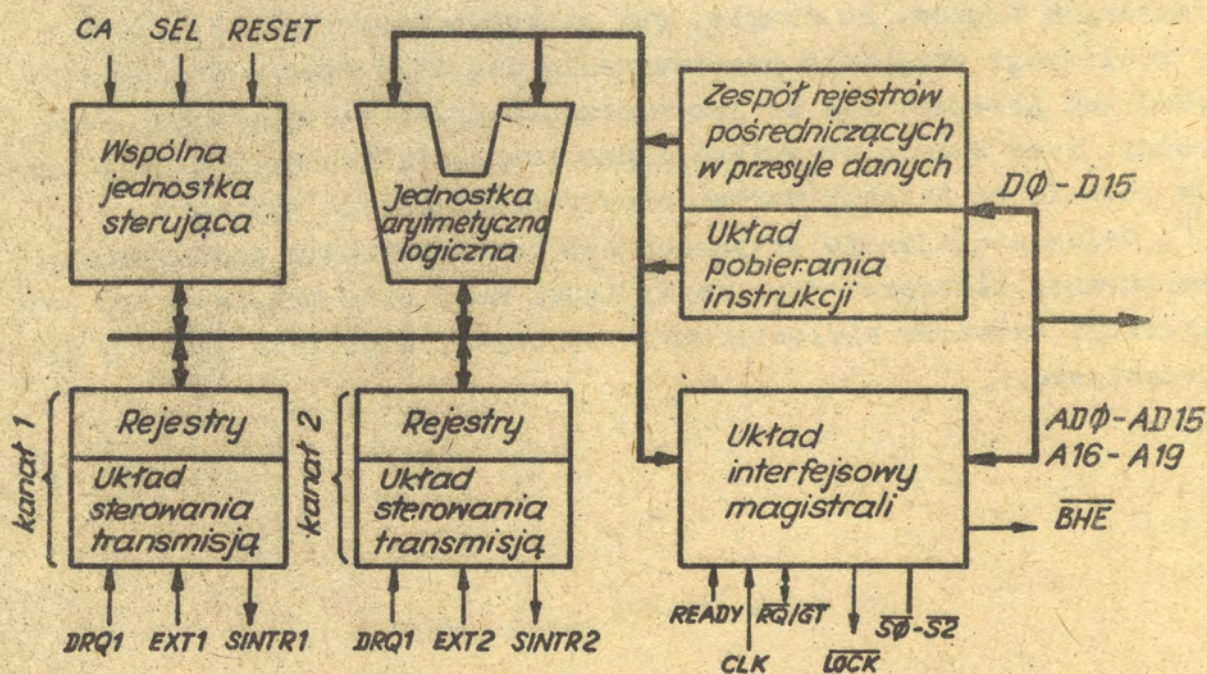
Celem niniejszego artykułu jest przedstawienie architektury wewnętrznej mikroprocesora we/wy Intel 8089 oraz podstawowych konfiguracji systemów wielomikroprocesorowych, w których jest on wykorzystywany.

2. BUDOWA I DZIAŁANIE MIKROPROCESORA WEJSCIA/WYJSCIA INTEL 8089

Mikroprocesor we/wy Intel 8089 jest samodzielnym mikroprocesorem 16-bitowym łączącym cechy mikroprocesora centralnego i kanału bezpośredniego dostępu. Może on zaadresować 1 MB przestrzeń pamięciową, czyli przestrzeń, w której zapis i odczyt dokonywane są sygnałami sterowania pamięcią (memory read, memory write), oraz 64 KB przestrzeń we/wy dostępną sygnałami I/O read, I/O write. W każdej z wymienionych przestrzeni mogą znajdować się zarówno urządzenia we/wy jak i pamięć.

Oczywiście podziału na przestrzeń adresową pamięci i we/wy można dokonać dla każdego mikroprocesora. Mikroprocesor we/wy Intel 8089 posiada jednak pewną wyjątkową cechę. Może on mianowicie wykonywać program umieszczony w przestrzeni we/wy generując na podstawie zadeklarowanego wstępnie znacznika właściwe sygnały sterujące odczytem w czasie wykonywania cyklu pobierania instrukcji. Cecha ta będzie miała istotne znaczenie w niektórych konfiguracjach systemów wielomikroprocesorowych.

Przejdziemy obecnie do omówienia budowy wewnętrznej mikroprocesora we/wy. Na rys.2 przedstawiono najważniejsze elementy struktury wewnętrznej tego mikroprocesora.



Rys.2. Architektura wewnętrzna mikroprocesora wejścia/wyjścia Intel 8089.

są nimi:

- dwa niezależne kanały transmisyjne,
- wspólna jednostka sterująca (ang. Common Control Unit - CCU),
- jednostka arytmetyczno-logiczna (ang. Arithmetic and Logic Unit - ALU),
- zespół rejestrów pośredniczących w przesyłaniu danych (ang. Assembly-Disassembly Registers),
- układ pobierania instrukcji (ang. Instruction Fetch Unit),
- układ interfejsowy magistrali (ang. Bus Interface Unit).

Kanały transmisyjne mikroprocesora we/wy połączone są równolegle i pracują w ogólności naprzemiennie. Praca kanałów polega na wykonywaniu programu lub transmisji. Każdy z nich zawiera grupę własnych rejestrów oraz układ sterowania transmisją. W czasie pracy kanał może korzystać z wewnętrznych elementów mikroprocesora we/wy, które nie są zdublowane i za ich pomocą komunikować się z innymi modułami systemu wielomikrokomputerowego.

Wspólna jednostka sterująca realizuje wewnętrzny arbitraż i rozdział sterowania.

Jednostka arytmetyczno-logiczna wykonuje operacje arytmetyczne i logiczne na liczbach maksymalnie dwudziestobitowych dzięki czemu ułatwione jest generowanie adresu z przestrzeni 1 MB.

Rejestry pośredniczące w przesyłaniu danych łączą wewnętrzną magistralę mikroprocesora we/wy z zewnętrzną magistralą danych. Ponadto w czasie transmisji mogą zamieniać przesyłane bajty w słowa lub odwrotnie, co pozwala na przyspieszenie obsługi układów o różnych szerokościach magistrali danych - 8 lub 16 bitów. Zostanie to dokładniej wyjaśnione w dalszej części artykułu, przy omawianiu przebiegu transmisji.

Układ pobierania instrukcji posiada jednobajtową kolejkę przechowującą sprowadzony nadmiarowo, to jest w czasie transmisji słowa, bajt następnej instrukcji.

Układ interfejsowy magistrali kontroluje dostęp mikroprocesora we/wy do magistrali systemu.

Warto zwrócić uwagę na dwa ograniczenia dotyczące równoległej pracy kanałów mikroprocesora we/wy. Pierwsze ograniczenie polega na konieczności, w ogólnym przypadku, naprzemienną pracę kanałów transmisyjnych spowodowanej wspólnym wykorzystywaniem przez nie układów interfejsowych magistrali adresowej i danych. Drugie ograniczenie wiąże się z wykorzystywaniem tych samych wewnętrznych rejestrów każdego z kanałów w czasie transmisji oraz

w czasie wykonywania programu.

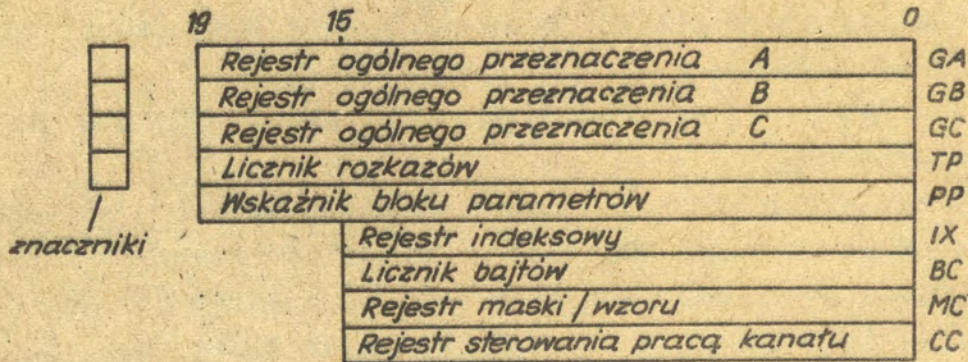
Omówimy obecnie szczegółowo funkcje i budowę kanałów mikroprocesora we/wy Intel 8089.

Jak wynika z rys.2 w strukturze kanału wyróżniamy:

- układ sterujący transmisją oraz
- grupę rejestrów.

Układ sterujący transmisją nadzoruje jej przebieg na podstawie zewnętrznych sygnałów: sygnału DRQ - synchronizacji transmisji i sygnału EXT - zakończenia transmisji. Ponadto układ ten może sygnalizować zakończenie transmisji przez wysłanie sygnału przerwania.

Na rys.3 przedstawiono wewnętrzne rejestry kanału transmisyjnego.



Rys.3. Wewnętrzne rejestry kanału transmisyjnego.

Grupa rejestrów obejmuje pięć rejestrów dwudziestobitowych oznaczonych GA, GB, GC, TP, PP oraz cztery rejestry szesnastobitowe: IX, BC, MC, CC. Rejestry GA, GB, GC, TP poprzedzone są jedno-bitowymi znacznikami, których rola zostanie wyjaśniona podczas omawiania zasad działania kanału.

Przejdziemy teraz do wyjaśnienia przeznaczenia rejestrów w czasie wykonywania transmisji przez kanał. Kanał transmisyjny realizuje transmisję w dwóch cyklach, co oznacza, że odczytana informacja jest najpierw przesyłana do wewnętrznych rejestrów układu, a następnie wysyłana do odbiornika.

W rejestrach GA i GB przechowywane są adresy nadajnika i odbiornika, którymi może być pamięć operacyjna i urządzenia zewnętrzne.

W rejestrze GC znajduje się adres tak zwanej tablicy przyporządkowania, na podstawie której możliwe jest tłumaczenie przesyłanych danych. Tłumaczenie to polega na przyporządkowaniu przesyłanemu bajtowi nowej wartości odczytywanej z komórki pamięci o adresie będącym sumą zawartości rejestru GC i pierwotnej wartości bajtu.

Rejestr BC pełni rolę licznika przesyłanych bajtów.

W rejestrze MC przechowywana jest ośmiobitowa maska oraz ośmiobitowa wartość wzorcowa, które są wykorzystywane do ewentualnego testowania transmitowanych bajtów. Testowanie to pozwala uzależnić przebieg transmisji od wartości przesyłanych bajtów.

Przedstawiona, dwufazowa metoda realizacji transmisji umożliwia:

- łatwą obsługę urządzeń zewnętrznych i pamięci niezależnie od czasów dostępu,
- przeprowadzenie operacji na transmitowanych danych np. porównania, tłumaczenia, itp.
- realizację czterech wariantów transmisji: pamięć-pamięć, pamięć-urządzenie we/wy, urządzenie we/wy-pamięć oraz urządzenie we/wy-urządzenie we/wy.

W szesnastobitowych systemach mikrokomputerowych spotykamy często obok układów szesnastobitowych, na przykład pamięci operacyjnej, również układy ośmiobitowe, zwłaszcza sterowniki urządzeń we/wy. W związku z tym pojawia się problem organizacji transmisji między układami o różnej szerokości magistrali danych. Mikroprocesor we/wy Intel 8089 realizuje tego typu transmisje w ten sposób, że dwóm operacjom dostępu do układu ośmiobitowego odpowiada jeden dostęp do układu szesnastobitowego. Operacja zamiany bajtów w słowo i odwrotnie wykonywana jest przez zespół rejestrów pośredniczących w przesyłaniu danych. Przebieg transmisji jest zależny od wartości słowa sterującego przechowywanego w rejestrze sterowania pracą kanału - CC (ang. Channel Control). Poszczególne pola tego słowa określają:

- wariant transmisji:
 - pamięć - pamięć,
 - pamięć - urządzenie we/wy,
 - urządzenie we/wy - pamięć,
 - urządzenie we/wy - urządzenie we/wy;
- charakterystykę
 - transmisja z równoczesnym tłumaczeniem lub bez,

- synchronizacja transmisji przez nadajnik lub odbiornik,
- przydział rejestrów adresowych nadajnika i odbiornika,
- ewentualna blokada magistrali w czasie transmisji lub wykonywania programu;
- warunki zakończenia transmisji:
 - zakończenie na podstawie rezultatu porównania według maski,
 - zakończenie po przetransmitowaniu zadanej liczby bajtów,
 - zakończenie po otrzymaniu zewnętrznego sygnału,
 - zakończenie po wykonaniu pojedynczej transmisji.

Przed rozpoczęciem transmisji zawartość rejestru sterowania jest ustalona przez program wykonywany w kanale.

Omówimy obecnie działanie kanału w trakcie wykonywania programu. W tym przypadku funkcje poszczególnych rejestrów ulegają zmianie. Rejestry GA, GB, GC i BC stają się rejestrami ogólnego przeznaczenia, natomiast rejestr TP (ang. Task Pointer) - licznikiem rozkazów wykonywanego programu.

Rejestr wskaźnika bloku parametrów PP (ang. Parameter Block Pointer) jest wykorzystywany do współpracy mikroprocesora we/wy z mikroprocesorem centralnym. Zasady tej współpracy zostaną omówione szczegółowo w rozdziale 3. W tym miejscu zasygnalizujemy tylko, że polega ona między innymi na wymianie informacji poprzez wydzielony obszar wspólnej pamięci operacyjnej zwanej blokiem parametrów. Adres początku tego bloku przechowywany jest w rejestrze wskaźnika bloku parametrów.

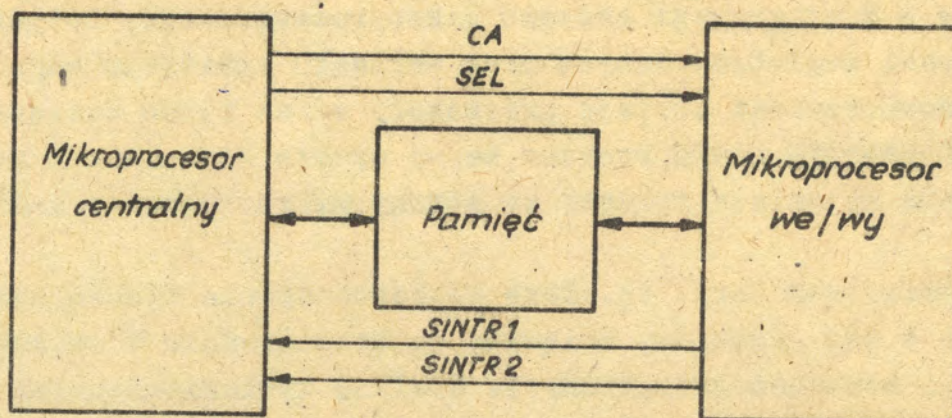
Zawartość rejestru indeksowego IX może być sumowana z zawartością rejestru bazowego GA, GB, GC lub PP w czasie tworzenia adresu argumentu wykonywanego rozkazu.

Rejestry GA, GB, GC, TP poprzedzone są jednobitowymi znacznikami. Jeżeli rejestry te są wykorzystywane do adresowania to powyższy znacznik jednoznacznie określa przestrzeń adresowaną danym rejestrem - przestrzeń adresowa pamięci lub we/wy.

Zestaw pięćdziesięciu typów rozkazów mikroprocesora we/wy Intel 8089 obejmuje rozkazy transmisji danych, rozkazy arytmetyczne i logiczne z możliwością operowania na pojedynczych bitach, rozkazy skoków i odwołań do podprogramów oraz rozkazy sterujące. Dostępne są następujące tryby adresowania: bazowe, bazowe z przesunięciem, indeksowe oraz indeksowe z autoinkrementacją.

3. WSPÓŁPRACA MIKROPROCESORA CENTRALNEGO Z MIKROPROCESOREM WE/WY

Jak wspomniano w rozdziale 2, mikroprocesor we/wy jest samodzielnym mikroprocesorem jednakże w systemie wielomikrokomputerowym pełni rolę układu funkcjonalnie podporządkowanego mikroprocesorowi centralnemu. Współpraca mikroprocesora centralnego z mikroprocesorem we/wy polega na przesyłaniu sygnałów sterujących oraz wymianie komunikatów i danych poprzez wspólnie dostępne obszary pamięci operacyjnej - rys.4.



Rys.4. Zasady komunikacji między mikroprocesorem centralnym a mikroprocesorem we/wy.

Przedstawiony na rysunku sygnał CA (ang. Channel Attention) jest sygnałem uaktywnienia mikroprocesora we/wy natomiast sygnał SEL (ang. selection) umożliwia wybór jednego z dwóch kanałów transmisyjnych.

Sygnały SINTR1 i SINTR2 (ang. System Interrupt) to sygnały przerwania, które służą do poinformowania mikroprocesora centralnego o wykonaniu żądanej operacji.

Zasadniczym celem współpracy jest, jak wiadomo, maksymalne odciążenie mikroprocesora centralnego, a ponadto uproszczenie, z jego punktu widzenia, obsługi urządzeń zewnętrznych.

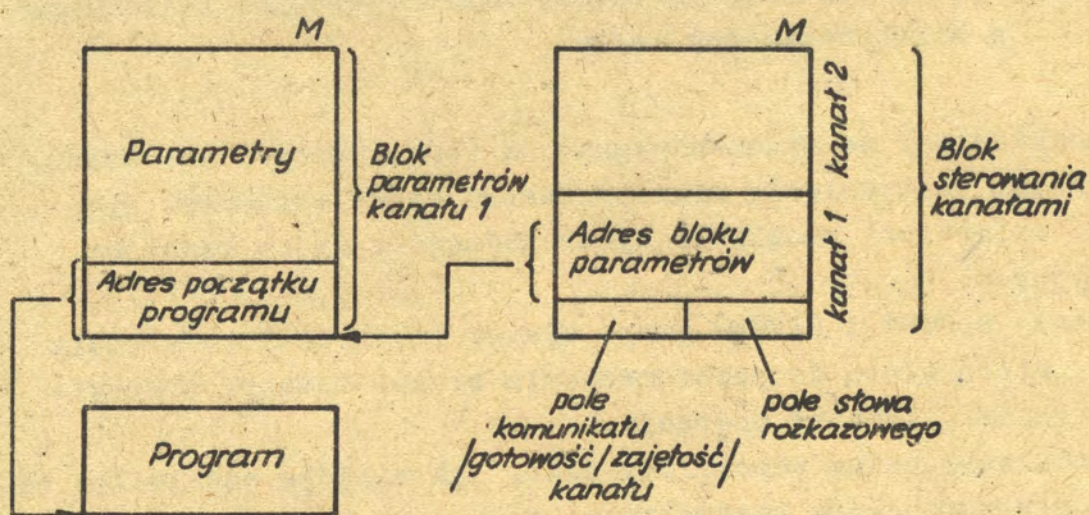
Możemy wyróżnić trzy fazy przebiegu współpracy:

- fazę inicjowania,
- fazę przeprowadzania transmisji,
- fazę ingerowania w przebieg transmisji.

Faza inicjowania to pierwszy etap sterowania pracą mikroprocesora we/wy. Konieczność wystąpienia tej fazy wynika z faktu istnienia wielu możliwych konfiguracji systemów wielomikroprocesorowych wykorzystujących mikroprocesory we/wy. Konfiguracje te zostaną dokładnie omówione w rozdziale 4, teraz wspomnimy tylko, że systemy mogą różnić się szerokościami magistral danych, sposobami połączenia mikroprocesora centralnego z mikroprocesorem we/wy itp.

Faza inicjowania rozpoczyna się w momencie wysłania przez mikroprocesor centralny do mikroprocesora we/wy sygnałów CA i SEL bezpośrednio po sygnale zerującym RESET. Mikroprocesor we/wy odczytuje wówczas z pamięci przygotowane przez mikroprocesor centralny dane o konfiguracji systemu mikroprocesorowego. Dotyczą one szerokości magistral danych oraz wariantu arbitrażu magistrali lokalnej. Równocześnie zostaje przekazany adres bloku sterowania kanałami. Następnie mikroprocesor we/wy wysyła komunikat o gotowości i czeka na dalsze rozkazy ze strony mikroprocesora centralnego.

Dwie pozostałe fazy, tj. faza przeprowadzania transmisji i ingerowania w jej przebieg, związane są bezpośrednio z obsługą transmisji i urządzeń zewnętrznych. Analizę wymienionych faz poprzedzimy przedstawieniem bloków pamięciowych wykorzystywanych przez mikroprocesory.



Rys.5. Bloki wymiany danych i komunikatów

Bloki oznaczone symbolem M muszą znajdować się w przestrzeni adresowej pamięci.

Na rysunku 5 przedstawiono trzy bloki pamięci:

- blok sterowania kanałami,
- blok parametrów kanału,
- blok programu.

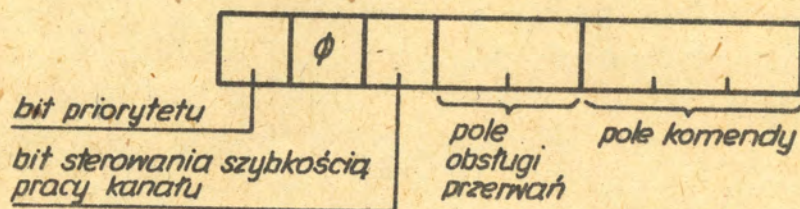
Pierwszy z nich składa się z dwóch, logicznie identycznych modułów przeznaczonych dla dwóch kanałów mikroprocesora we/wy. Przechowywane są w nich: słowo rozkazowe, komunikat informujący o zajętości kanału oraz adres bloku parametrów.

Drugi moduł stanowi obszar wymiany danych między mikroprocesorem centralnym, a mikroprocesorem we/wy.

W trzecim module umieszczony jest program obsługi transmisji przeznaczony dla kanału mikroprocesora we/wy. Moduł ten może znajdować się w przestrzeni adresowej pamięci lub we/wy.

Fazę przeprowadzania transmisji rozpoczyna mikroprocesor centralny od przygotowania bloku parametrów transmisji oraz ewentualnie programu transmisji. Następnie mikroprocesor centralny sprawdza czy kanał mikroprocesora we/wy nie jest zajęty. Po stwierdzeniu niezajętości kanału uaktualnia adres bloku parametrów oraz definiuje słowo rozkazowe w bloku sterowania kanałem i uaktywnia mikroprocesor we/wy sygnałem CA. Po otrzymaniu sygnału uaktywnienia mikroprocesor we/wy wysyła komunikat o zajętości i przechodzi do wykonania polecenia zawartego w słowie rozkazowym. Po zakończeniu transmisji następuje przesłanie komunikatu informującego o gotowości i ewentualne wysłanie sygnału przerwania.

Struktura słowa rozkazowego przedstawiona jest na rys.6.



Rys.6. Struktura słowa rozkazowego

W słowie tym wyróżniamy:

- pole komendy,
- pole obsługi przerw,
- bit sterowania szybkością pracy kanału,
- bit priorytetu.

W polu komendy zawarta jest informacja o lokalizacji programu obsługi transmisji w jednej z dwóch dostępnych przestrzeni adresowych (pamięciowej lub we/wy).

W polu obsługi przerw mikroprocesor centralny przekazuje żądanie usktywnienia lub zablokowanie przerw.

Bit sterowania szybkością pracy kanału jest wykorzystywany do ewentualnego spowolnienia wykonywanego przez kanał program (z wyjątkiem samej transmisji). Powoduje to zmniejszenie obciążenia magistrali przez mikroprocesor co może być istotne w pewnych zastosowaniach systemów wielomikroprocesorowych.

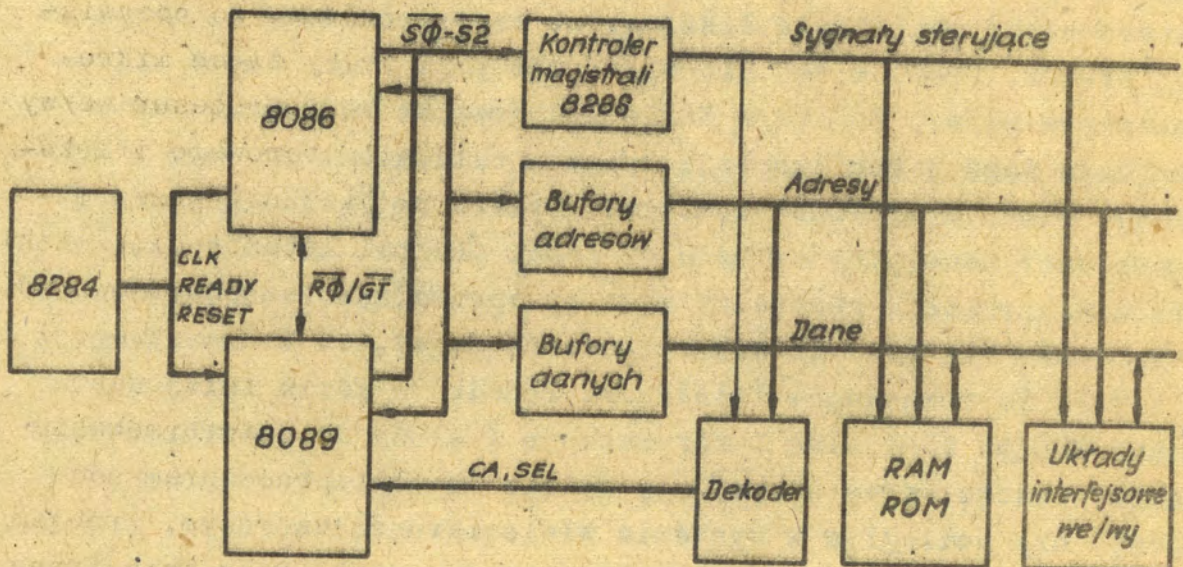
Bit priorytetu używany jest przez jednostkę sterującą w czasie nadzorowania współbieżnej pracy kanałów.

Na zakończenie omówimy fazę ingerencji w przebieg transmisji. Obejmuje ona zatrzymywanie, wznawianie lub anulowanie programu wykonywanego w kanale mikroprocesora we/wy. Przebieg tej fazy jest analogiczny do fazy przeprowadzania transmisji. Występuje tylko różnica w zawartości pola komendy słowa rozkazowego. Istnienie tej fazy pozwala mikroprocesorowi centralnemu na elastyczne, stosowne do wymagań całego systemu, oddziaływanie na mikroprocesor we/wy. Ma to istotne znaczenie zwłaszcza w środowisku wieloprogramowym.

4. ARCHITEKTURY SYSTEMÓW WIELOMIKROPROCESOROWYCH Z MIKROPROCESOREM WE/WY

Jak wiadomo, wyróżniamy systemy wielomikroprocesorowe z dzieloną magistralą lokalną oraz dzieloną magistralą systemową [1]. Mikroprocesor we/wy może pracować w systemach o obu powyższych architekturach.

Omówimy najpierw systemy z dzieloną magistralą lokalną. Na rys.7 przedstawiono system, w którym magistrala lokalna jest dzielona między mikroprocesor centralny i mikroprocesor we/wy.



Rys.7. System z mikroprocesorem centralnym i mikroprocesorem we/wy, o architekturze z dzieloną magistralą lokalną.

Oba mikroprocesory są zsynchronizowane tym samym układem zegarowym oraz wykorzystują wspólnie układy kontrolera magistrali, oraz bufory adresów i danych. Oba mają też dostęp do całej pamięci operacyjnej i wszystkich układów interfejsowych urządzeń zewnętrznych dołączonych do magistrali systemu. Arbitraż magistrali lokalnej dokonywany jest przy wykorzystaniu linii $\overline{RQ}/\overline{GT}$ mikroprocesora centralnego i mikroprocesora we/wy 1. Oczywiście w

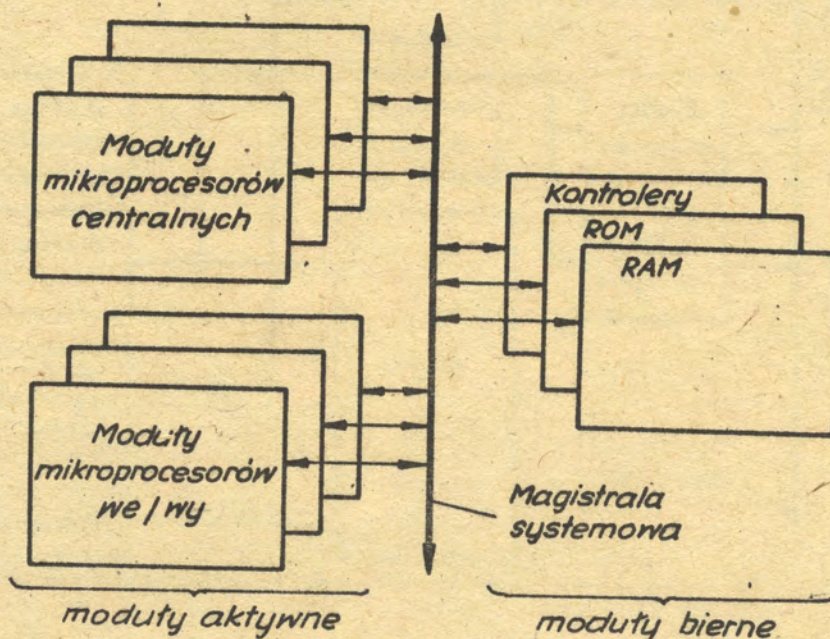
systemie o analizowanej architekturze mikroprocesor we/wy jest mikroprocesorem podległym w stosunku do mikroprocesora centralnego.

W systemach z dzieloną magistralą lokalną, ze względu na istnienie dwóch linii $\overline{RQ}/\overline{GT}0$ i $\overline{RQ}/\overline{GT}1$, w które wyposażony jest mikroprocesor centralny, można dołączyć dwa mikroprocesory we/wy do jednego mikroprocesora centralnego lub mikroprocesor we/wy i koprocesor (np. Intel 8087). Zasady pracy i arbitrażu w takim systemie pozostają bez zmian [1].

W systemach z dzieloną magistralą lokalną można również połączyć dwa mikroprocesory we/wy. Na wstępie zauważmy, że podobne połączenie dwóch mikroprocesorów centralnych nie jest możliwe ponieważ po odebraniu sygnału zerującego RESET oba równocześnie wysterowałyby magistralę lokalną, co doprowadziłoby do oczywistych błędów. Problem ten nie występuje przy pracy dwóch mikroprocesorów we/wy, bowiem w fazie inicjowania mikroprocesor we/wy odczytuje dane o konfiguracji systemu mikrokomputerowego i lokalizacji bloku sterowania kanałami dopiero po wysłaniu przez mikroprocesor centralny sygnału CA (ang. Channel Attention). Dzięki temu mikroprocesor centralny może przeprowadzać fazę inicjowania niezależnie dla każdego mikroprocesora we/wy, co w konsekwencji umożliwia im równoległe działanie. Ponadto w fazie inicjowania sygnał SEL ma specjalną interpretację i służy do poinformowania mikroprocesora we/wy o tym, czy będzie on mikroprocesorem nadrzędnym czy podległym w systemie wielomikroprocesorowym. Problem arbitrażu magistrali lokalnej rozwiązywany jest przez wewnętrzne układy logiczne $\overline{RQ}/\overline{GT}$ przy czym możliwa jest realizacja dwóch wariantów arbitrażu. Wariant tak zwany zerowy jest kompatybilny z układami $\overline{RQ}/\overline{GT}$ mikroprocesorów centralnych; wariant tak zwany pierwszy jest dostępny tylko w mikroprocesorach we/wy 8089. Podstawowa różnica między tymi wariantami polega na częstotliwości z jaką następuje przełączenie mikroprocesorów na magistrali lokalnej. W wariantcie pierwszym następuje ono częściej niż w zerowym dzięki czemu mikroprocesory we/wy korzystają z magistrali w przybliżeniu jednakowo często. Przejście jednego z nich do wykonywania transmisji w kanale pozwala mu szybko uzyskać dostęp do magistrali i korzystać z niej czasie wymaganym do prawidłowego przeprowadzenia transmisji.

Przejdziemy obecnie do omówienia systemów z dzieloną magistralą systemową, w skład których wchodzi moduły z mikroprocesorami

rami we/wy. Na rys.8 przedstawiono ogólną architekturę takiego systemu.



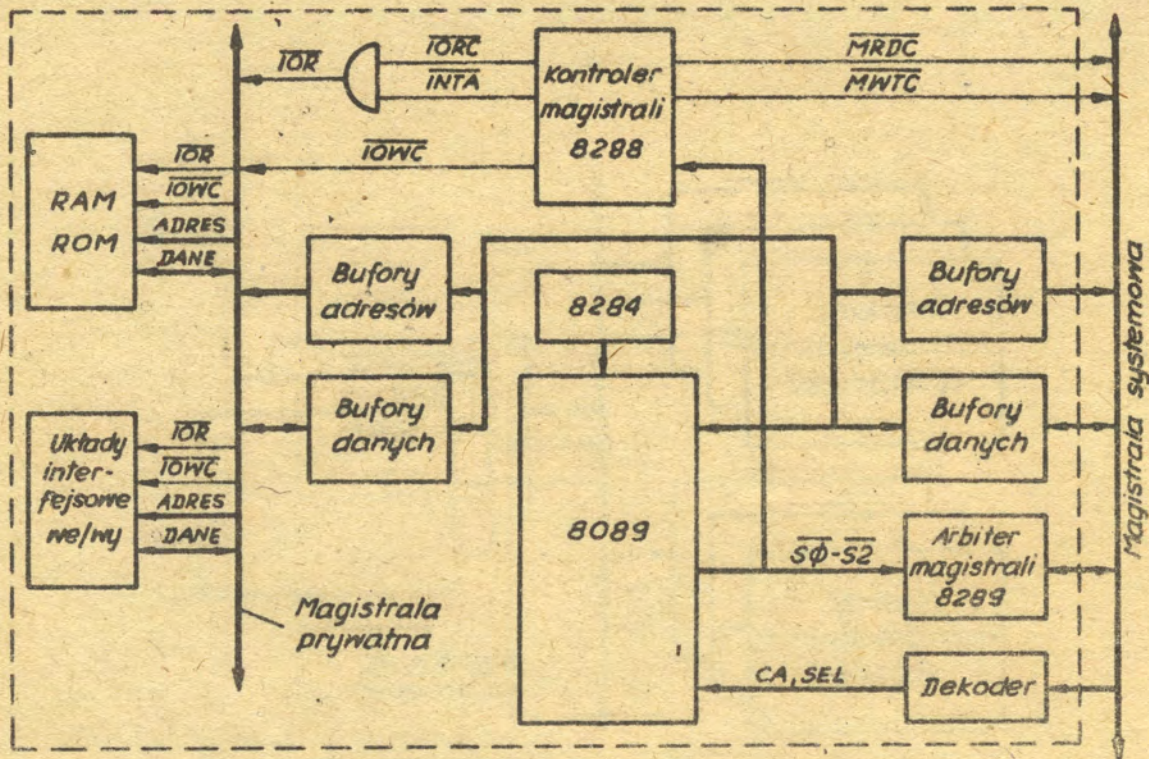
Rys.8. System z modułami mikroprocesorów we/wy, o architekturze z dzieloną magistralą systemową.

Jak widać, w systemie o takiej architekturze występują dwa rodzaje modułów aktywnych: moduły mikroprocesorów centralnych oraz moduły mikroprocesorów we/wy.

Na rys.9 przedstawiono wewnętrzną architekturę modułu mikroprocesora we/wy. Architektura tego modułu jest w dużej mierze analogiczna do architektury modułu mikroprocesora centralnego [1]. Podobnie jak w module mikroprocesora centralnego grupa układów sterowania magistralą systemową składa się z:

- arbitera magistrali (układ 8289),
- kontrolera magistrali (układ 8288),
- buforów magistrali adresów i danych.

Ponadto w skład tej grupy wchodzi dekodery sygnałów: CA i SEL, które umożliwiają uaktywnienie mikroprocesora we/wy przez mikroprocesor centralny.



Rys.9. Architektura wewnętrzna modułu mikroprocesora we/wy.

Grupa układów sterowania magistralą prywatną jest natomiast uproszczona, w stosunku do analogicznej grupy w module mikroprocesora centralnego, i zawiera jedynie układy buforowe magistral: adresowej i danych, nie zawiera natomiast kontrolera magistrali. Ze względu bowiem na możliwość wykonywania rozkazów przez mikroprocesor we/wy pobieranych z jego przestrzeni adresowej we/wy można dokonać rozdziału przestrzeni adresowych: systemowej i prywatnej podporządkowując tej pierwszej przestrzeń adresową pamięci, a tej drugiej przestrzeń we/wy. Wówczas linie komend dotyczące sterowania pamięcią wyprowadzone są z kontrolera na magistralę systemową natomiast linie sterowania operacjami we/wy wykorzystywane są w obrębie modułu.

W prezentowanym module zastosowano typowe układy kontrolera i arbitera magistrali (8288, 8289). Powyższe układy są przystosowane do pracy w systemach z mikroprocesorami we/wy i za pośrednictwem wejść programujących można ustalić właściwy tryb ich pracy. W tym trybie arbiter współpracujący z mikroprocesorem we/wy

analizuje jego stan i po wykryciu faktu adresowania pamięci interpretuje dane żądanie jako żądanie dostępu do magistrali systemowej, natomiast po wykryciu faktu adresowania we/wy interpretuje dane żądanie jako żądanie dostępu do magistrali prywatnej.

Kontroler magistrali 8288 generuje sygnały we/wy bez oczekiwania na zezwolenie ze strony arbitera. Natomiast wysyłanie sygnałów dotyczących operacji na pamięci wymaga tego zezwolenia.

Na zakończenie zwrócimy uwagę na sposób pobierania instrukcji z przestrzeni adresowej we/wy. Jak wiadomo, mikroprocesory centralne pobierają instrukcje zawsze z pamięci operacyjnej. Przyjmując zatem przedstawiony wyżej wariant pracy kontrolera i arbitera magistrali każdy cykl odczytu instrukcji powodowałby żądanie dostępu do magistrali systemowej. Aby tego uniknąć mikroprocesor we/wy sygnalizuje pobieranie instrukcji z różnych przestrzeni adresowych podając różny stan na liniach $\overline{S0} - \overline{S2}$.

Stan $\overline{000}$, oznaczający pobieranie instrukcji z przestrzeni adresowej we/wy, powoduje uaktywnienie wyjścia \overline{INTA} (ang. Interrupt Acknowledge) układu kontrolera 8288. Oczywiście, pojawienie się tego sygnału nie ma nic wspólnego z obsługą przerwań. Wykorzystano tylko wewnętrzny dekodery stanu z układu 8288. Również arbiter magistrali interpretuje powyższy stan jako operację dostępu do przestrzeni adresowej we/wy i w związku z tym nie ubiega się o dostęp do magistrali systemowej. Sygnał odczytu z przestrzeni adresowej we/wy jest zatem iloczynem logicznym sygnałów \overline{INTA} i \overline{IORC} (I/O read).

5. UWAGI KOŃCOWE

Połączenie cech mikroprocesora centralnego i kanałów DMA w mikroprocesorze we/wy Intel 8089 pozwala na przeprowadzenie szybkich transmisji oraz realizowanie skomplikowanych algorytmów obsługi urządzeń zewnętrznych w systemach mikroprocesorowych bez obciążania tymi funkcjami mikroprocesorów centralnych. Dzięki mikroprocesorowi we/wy mikroprocesor centralny widzi urządzenia zewnętrzne jako kanały DMA przeznaczone do transmisji blokowej.

Ze względu na swe właściwości mikroprocesory we/wy znajdują główne zastosowanie w układach obsługi takich urządzeń, jak pamięci dyskowe i monitory graficzne. Podczas obsługi dysków magnetycznych zajmują się oprócz przeprowadzaniem transmisji także

wstępnym zarządzaniem zbiorami. Obsługując monitory graficzne mikroprocesory we/wy zarządzają strukturami danych ponadto obsługują kursori klawiaturę oraz dokonują podstawowych operacji na ekranie.

BIBLIOGRAFIA

- 1 W.Cellary: Architektura systemów wielomikroprocesorowych, Mat. VI Szkoły Mikroprocesorowej, Łódź, grudzień 1984.

VI Szkoła Mikroprocesorowa CSI/PTI
Łódź, grudzień 1984

STANDARD MAGISTRALI MULTIBUS-II

doc.dr inż. Roman Trechciński
Instytut Problemów Jądrowych
05-400 Otwock-Świerk
/799-852/

1. ZAGADNIENIA WYBORU PERSPEKTYWICZNEGO SYSTEMU MODULARNEGO

Do głównych tendencji rozwojowych cyfrowych systemów modularnych w ostatnich latach można zaliczyć wieloprocessorowość i tworzenie struktur wielomagistralowych. W ostatnio anonsowanych systemach nawet arbitraż jest w pełni zdecentralizowany, a występujące w systemach magistrale są zróżnicowane i dostosowane do realizowanych zadań.

System Multibus II jest reprezentantem tego typu nowoczesnych rozwiązań, aczkolwiek nie jest jedynym wyraźnie zarysowanym standardem światowym, jakim był w latach siedemdziesiątych system Camac. Do systemów, które w okresie ostatnich lat rozpatrywano jako systemy kandydujące do roli standardu światowego można zaliczyć: GPIB/IEEE488 /NIM/, S 100, E 3S, Camac /6500/, Multibus I, AMS Bus, Multibus II, Eurobus, VME Bus, Future Bus /P896.1/, Fast Bus, G 64 i G 128, Nu Bus. Ponadto można oczekiwać, że w najbliższym czasie zostanie zaanonsowany przez firmę DEC nowy system pod nazwą "Backplane I". Parametry tego systemu nie są jeszcze znane.

Do systemów wieloprocessorowych dotychczas najbardziej popularnych na rynku światowym należały Multibus I i VME Bus,

zaś wśród systemów jednoprocessorowych Q-Bus, Camac i NIM /analogowy/. Obecne wersje systemu Camac i NIM można już traktować jako cyfrowe systemy wieloprocessorowe. Niektóre z wymienionych uprzednio systemów nie odegrały większej roli na rynku światowym pomimo, iż były rozwiązane prawidłowo z technicznego punktu widzenia.

Do nowych systemów, które mogą w przyszłości odegrać poważniejszą rolę zalicza się obecnie trzy: Multibus II, VME Bus /Versa Module Eurocard Bus/, AMS Bus. Przyszłość systemu "Back plane I" nie jest jeszcze pewna. Wydaje się, że wszystkie powyższe systemy znajdą się na rynku światowym. Jednakże, w jakich proporcjach istniejący rynek zostanie między nie podzielony, jest zagadnieniem trudnym do ustalenia. System VME Bus jest popierany przez firmę Motorola /oraz szereg innych firm/, zaś system Multibus II oraz AMS Bus przez firmę Intel /i inne/. Wprawdzie oba systemy /Multibus II i AMS Bus/ są systemami nowymi /1983-84 r/, jednakże AMS Bus jest systemem w większym stopniu powiązany z rozwiązaniami systemu Multibus I. AMS Bus jest więc obecnie opracowaniem nowym, ale z drugiej strony już w pewnym stopniu znanym na rynku światowym.

Porównanie systemów jest w dużej mierze kwestią indywidualnej interpretacji. Również kwestia doboru cech uwzględnianych w ocenie jest dyskusyjna. W niniejszym porównaniu przyjęto jako istotne następujące cechy:

- a/ potencjał produkcyjny systemów i elementów
 - b/ koszty opracowań i produkcji
 - c/ oprogramowanie
 - d/ stopień normalizacji światowej
 - e/ właściwości techniczne
 - f/ prognoza dotycząca okresu ekonomicznego starzenia się systemu.
- Jako zasadę tej oceny przyjęto, że jeśli nie występuje zdecydowana różnica jakościowa, to daną cechę uznaje się za równorzędną. Ocena formułowana jest tylko w formie lepiej/gorzej.

Potencjał produkcyjny systemów i elementów uznaje się za równorzędny dla wszystkich 3-ch systemów.

Głównym oparciem dla systemów Multibus II i AMS Bus jest firma Intel wraz z kilkunastoma innymi /konsorcjum/ i kilkudziesięcio-

ma /50-100/ popierającymi. Kilkadziesiąt placówek badawczych zajmuje się tymi systemami. Podobnie przedstawia się sytuacja odnośnie systemu VME Bus z wiodącą firmą Motorola.

Koszty opracowań i produkcji uznaje się także za równorzędne dla wszystkich 3-ch systemów. Główne uzasadnienie wynika z unifikacji rozwiązań konstrukcyjnych.

Aspekty oprogramowania podstawowego systemów są równoważne. Wszystkie systemy są wieloprocesorowe i wymagają odpowiedniego systemu operacyjnego/w chwili obecnej nie ma jeszcze pełnej informacji na ten temat/.

Stopień normalizacji światowej uznaje się za równorzędny dla wszystkich 3-ch systemów, natomiast właściwości techniczne systemu Multibus II są korzystniejsze w stosunku do VME Bus i AMS Bus /p. 5 niniejszego opracowania/.

Prognoza dotycząca ekonomicznego starzenia się systemu może powstać tylko na podstawie pewnego niedokładnego szacunku. Praktyka ostatnich lat dowodzi, że tylko systemy, które mają poparcie znaczących producentów mogą liczyć na sukcesy rynkowe. Dla systemów modularnych, poza 2-ma grupami firm zrzeszonych wokół firmy Motorola i Intel nie ma liczących się potęg. Problemem podstawowym, który interesuje obecnie opinię światową jest pytanie, czy można oczekiwać w najbliższych latach pojawienia się propozycji nowego systemu o jeszcze doskonalszych właściwościach. Większość ekspertów reprezentuje w tym przypadku opinię negatywną. Jako uzasadnienie podawane są następujące aspekty:

- koszty całości operacji wprowadzenia nowego systemu są znaczne i już je poniesiono, a pojawienie się nowej propozycji nie pozwoliłoby na wykorzystanie tego zaangażowania,
- odpowiedzią firmy Motorola na zaanonsowanie przez firmę Intel systemu Multibus II było przedstawienie propozycji magistrali VMX, stanowiącej rozszerzenie VME; gdyby magistrala zamierzała wprowadzić inny system, to nie inwestowałaby w VMX i nie wprowadzałaby w ten sposób w błąd swoich klientów,
- w ostatnich latach w dziedzinie modularnych systemów cyfrowych nastąpił wyraźny postęp, szereg właściwości systemów poprawiono nieomal do granic możliwości, można również mówić o

osiągnięciu znacznego stopnia optymalizacji i unifikacji; prawdopodobieństwo uzyskania nowego, znacznego skoku jakościowego jest więc obecnie niezwykle małe, a tylko system wyraźnie lepszy mógłby się okazać konkurencyjny /w tych rozważaniach pominięto system "Backplane I", ze względu na brak danych/.

W tych warunkach można przyjąć, że Multibus II jako system wyraźnie lepszy technicznie powinien dłużej istnieć na rynku światowym. Jeżeli nawet za 5 lat pojawi się nowy konkurent, to i tak jeszcze co najmniej przez dalsze 5 lat pozycja ustabilizowanego standardu światowego jest nienaruszalna. Wynika stąd, że minimalny okres starzenia się dla systemu Multibus II może być przyjęty na 10 lat. W stosunku do Multibus II okres ekonomicznego starzenia się VME Bus i AMS Bus powinien być mniejszy.

2. UNIFIKACJA ROZWIĄZAŃ KONSTRUKCYJNYCH

Procesy tworzenia wielu różnych systemów w ostatnich latach doprowadziły do pewnej selekcji poszczególnych rozwiązań, co w ostatecznym efekcie stworzyło w sposób pierwotnie zamierzony dosyć daleko zaawansowaną unifikację. Do tych aspektów należą: konstrukcja mechaniki, typ złącz magistrali, zasilanie.

Konstrukcja mechaniki opiera się o system Eurokart z tym, że w prawie wszystkich nowych systemach stosowana jest podwójna Eurokarta /wysokość 233,35mm/ o głębokości 220 lub 160 mm /bardziej perspektywiczny wydaje się wymiar 220mm/. Szerokość modułu wynosi prawie zawsze 20,32 mm /tj. 4x5,08mm/. Pozwala to na umieszczenie 20 lub 21 modułów w tzw. 19 calowej kasecie.

Jako typ złącza stosowane jest wyłącznie złącze C 96 /z 96 kontaktami lub rzadziej z 64 kontaktami/. Magistrale systemu z reguły zajmują jeden rząd złącz. Drugi rząd pozostawiany jest do wykorzystania przez projektantów systemów, np. do połączeń zewnętrznych z obiektem automatyzowanym lub do rozmieszczenia magistrali dodatkowej /LBX w systemie Multibus II, VMX jako uzupełnienie VME Bus/.

Jako napięcia zasilające przyjmuje się obecnie prawie wyłącznie następujące:

- + 5V -- główne źródło zasilania
- + 12V- uzupełniające źródła zasilania
- + 5V - zasilanie bateryjne /ochrona pamięci/.

Tendencja do unifikacji występuje również w stosunku do innych własności nowych systemów, nie jest jednak jeszcze tak wyrażana, aby można było mówić o osiągnięciu unifikacji.

3. TENDENCJE ROZWOJOWE

Podstawową tendencją rozwojową systemów modularnych reprezentowaną w ostatnich latach było dążenie do uzyskania dużej uniwersalności, a więc do stworzenia dla danego systemu możliwie najszerszego zakresu zastosowań, co przejawia się w przyjętych założeniach i wyborze właściwości systemów.

Do najważniejszych właściwości nowoczesnych systemów modularnych można zaliczyć:

- a/ wieloprocesorowość - pierwotne rozwiązania jednoprocessorowe zastępowane przez wieloprocesorowe z arbitrażem scentralizowanym, a ostatnio przez wieloprocesorowe ze zdecentralizowanym arbitrażem,
- b/ wielomagistralowość - systemy jednomagistralowe zastępowano przez wielomagistralowe, a ostatnio przez wielomagistralowe z różnymi magistralami dostosowanymi do określonych zadań /operacji/,
- c/ szybkość transmisji - dążono do uzyskania maksymalnej szybkości transmisji tak, aby system mógł być stosowany zarówno w automatyce przemysłowej jak i w automatyzacji badań naukowych; ostatnio osiągnięto dla głównej magistrali równoległej szybkość 40 M bajtów/s przy zachowaniu logiki TTL, tj. niewiele mniej od najszybszego aktualnie systemu Fast Bus /70 M bajtów/s z logiką ECL/,
- d/ odporność na zakłócenia - pierwotne rozwiązania /np. w systemie Camac/ zastępowano przez systemy z kontrolą parzystości /w zasadzie z jednym bitem parzystości na jeden transmitowany bajt/; ponadto stosowana przed kilkoma laty transmisja asynchroniczna

- /handshake/ została zastąpiona przez tzw. "synchronizowany handshake" co znacznie zmniejsza prawdopodobieństwo wystąpienia błędu,
- e/ długość słowa - różnorodne uprzednie rozwiązania /8, 16, 24 lub 8 i 16 bitów/ zostały zastąpione przez występujące w jednym systemie słowa: 8-, 16- i 32-bitowe lub nawet 8-, 16-, 24- i 32-bitowe /Multibus II/,
- f/ pole adresowania - obecnie rozróżnia się na ogół kilka obszarów adresowania, obszar dotyczący adresowania pamięci obejmuje 32 bity,
- g/ likwidacja procedury przerw - kilka nowych systemów /1983-84r./ nie posiada już odrębnego rozwiązania przerw, a funkcja przerw stała się zbędna /zastąpiona przez zgłoszenia w celu uzyskania dostępu do magistrali/ przyczyną tego stanu jest wprowadzenie wielomagistralowości, wieloprocesorowości i zwiększenie szybkości transmisji.

4. PORÓWNANIE KONFIGURACJI MULTIBUS II, VME BUS, AMS BUS

Na rys.1. przedstawiono konfigurację systemów Multibus II, VME Bus oraz AMS Bus.

Multibus II jest systemem 32-u bitowym /opcje: 24, 16, 8 bitów/, zaś VME Bus i AMS Bus są systemami 16-bitowymi. Wprawdzie specyfikacja VME Bus przewiduje wersję 32-u bitową, ale zajmuje ona rząd B złącza II, co jest ogromnym utrudnieniem. Stąd też wynika fakt, że większość obecnych zastosowań VME Bus opiera się na wersji 16-bitowej i chyba nie można liczyć się z możliwością uzyskania dużej popularności VME w wersji 32-bitowej.

Konfiguracja Multibus II przewiduje istnienie pięciu różnych magistral. Podstawową, równoległą magistralą ogólnego zastosowania jest PSB /parallel system bus/. Jej odpowiednikami są: VME w systemie VME Bus i AMS w systemie AMS Bus.

Szeregowa magistrala SSB /serial system bus/ zajmująca 2 kontakty I złącza służy do niezależnej, wolnej komunikacji pomiędzy modułami danej kasy, jak też i komunikacji międzykasetowej na niewielkie odległości /10 m/. Jej odpowiednikiem jest

VMS w systemie VME Bus. W systemie AMS Bus nie przewiduje się magistrali szeregowej, jednakże złącze I posiada 6 wolnych kontaktów, z których dwa mogą być z powodzeniem wykorzystane na magistralę szeregową. Potencjalna możliwość wprowadzenia magistrali szeregowej istnieje więc także w systemie AMS Bus.

Magistrala lokalna LBX /local bus extension/ służy wyłącznie do niezależnej komunikacji pomiędzy CPU /modułem typu master/ i pamięciami. Jest to szybka /48 M bajtów/s/, równoległa magistrala, zajmująca część kontaktów złącza II, ale obejmująca nie więcej niż 6 stanowisk/ często mniej niż 4/. Jej odpowiednikiem jest VMK w systemie VME Bus. W systemie AMS Bus nie przewiduje się tego typu magistrali, jednakże złącza II są pozostawione do dyspozycji projektanta i wprowadzenie magistrali jest potencjalnie możliwe.

Magistrala SBX /system bus extension/ służy do wewnętrznej komunikacji pomiędzy SBC /single board computer - moduł typu master/, a modułami SBX Multimodule. Celem wprowadzenia tej magistrali jest umożliwienie korzystania z popularnych modułów "Multimodules" dla rozszerzenia funkcji SBC. Z punktu widzenia porównania struktury /konfiguracji/ systemu magistrala ta nie odgrywa więc żadnej roli. Dodanie do SBC płytki "Multimodule" tworzy moduł podwójnej szerokości, z tym, że część "Multimodule" nie ma dostępu do złącz I i II.

Magistrala Multichannel I/O DMA Bus jest wprowadzona przez złącze umieszczone na przednim panelu modułów i służy do komunikacji międzysystemowej /międzykasetowej/ oraz między modułami danej kasety. Umożliwia ona dołączenie 16 urządzeń przy odległości 15 m. Magistrala taka może być stosowana w zasadzie w każdym systemie /specyfikacje tego nie ograniczają/. Wynika stąd, że dla porównania konfiguracji można jej nie brać pod uwagę.

Ostatecznie można powiedzieć, że struktura Multibus II jest identyczna ze strukturą VME Bus, zaś w AMS Bus istnieje możliwość uzyskania identycznej struktury jak w dwu poprzednich systemach. Wszystkie trzy systemy są więc w zasadzie równorzędne z punktu widzenia konfiguracji, zaś podstawę do wyłonienia różnic stanowią inne właściwości.

5. PORÓWNANIE PODSTAWOWYCH WŁAŚCIWOSCI TECHNICZNYCH SYSTEMU MULTIBUS II Z VME BUS I AMS BUS

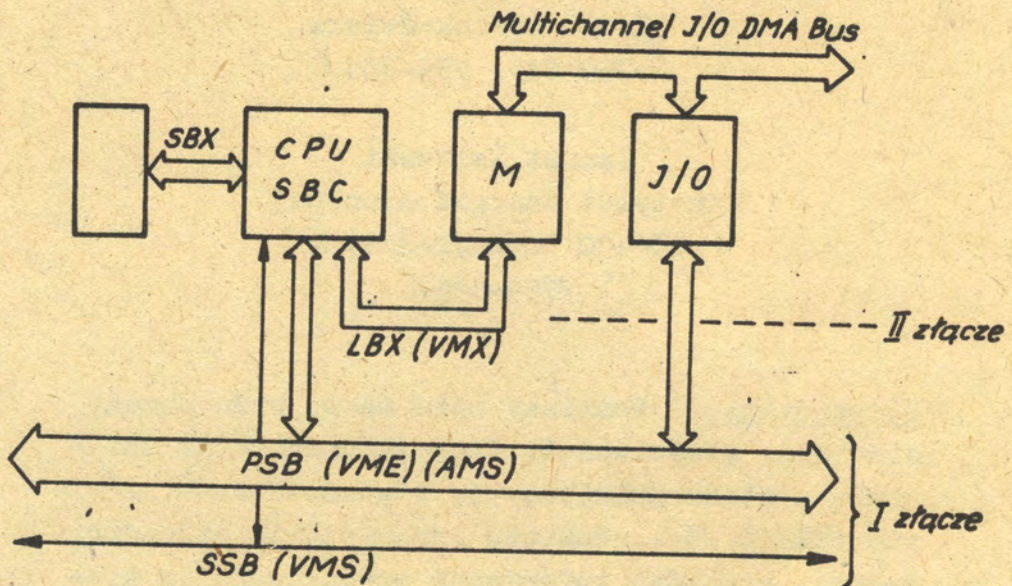
W niniejszym porównaniu uwzględniono tylko najważniejsze różnice /wg oceny autora/. Uwzględniono w nim 16-bitową wersję VME Bus oraz pominięto fakt, że w systemie AMS Bus w zasadzie nie ma magistrali lokalnej i szeregowej. Przy tych założeniach lista głównych różnic przedstawia się następująco:

	Multibus II	VME Bus	AMS Bus
a/ Długość słowa	32, 24, 16, 8	16, 8	16, 8
b/ Głębokość modułu	220 mm	160 mm	160 mm
c/ Pole adresowe	4 G bajty	16 M bajtów	16 M bajtów
d/ Szybkość /max/	40 M bajtów/s	20 Mbajtów/s	20 Mbajtów/s
e/ Typ operacji	synchr. handshake	handshake	handshake
f/ Odrębna procedura przerwania	nie ma	jest	jest
g/ Baza elementowa	Intel	Motorola	Intel
h/ Arbitraż	zdecentralizowany	centralny	centralny
i/ Magistrala szeregową	2 Mb/s, 10m	3,5 Mb/s, 50 cm	-

Szereg innych właściwości wymienionych systemów rozwiązano w różny sposób, jednakże jakość tych rozwiązań może być uznana za równorzędną. Do takich rozwiązań zalicza się np. brak w systemie Multibus II niewygodnych linii typu "daisy chain" dla wyróżnienia numeru stanowiska; organizacja logiczna systemów jest również odmienna, ale równorzędna jakościowo. Problemem, który również pominięto w tym porównaniu, a który mogłyby mieć istotne znaczenie, jest stopień powiązania systemu z określonym typem mikroprocesora. Jednakże ten wskaźnik jest trudny do wykorzystania porównawczego, gdyż nawet względnie silne powiązanie nie stanowi w zasadzie cechy ujemnej, o ile dany typ mikroprocesora jest

łatwo dostępny.

W rezultacie porównania właściwości systemów można ustalić ogólny wniosek, że po pominięciu punktu g/ wszystkie właściwości są rozwiązane korzystniej w systemie Multibus II. Natomiast systemy VME Bus i AMS Bus są systemami równorzędnymi względem siebie.



Rys. 1. Konfiguracja systemów Multibus II, VME Bus, AMS Bus

VI Szkoła Mikroprocesorowa CSI/PTI
Łódź, grudzień 1984

ORGANIZACJA LOGICZNA SYSTEMU MULTIBUS-II

Grzegorz Dzięglewski, Krzysztof Rzymkowski
Instytut Problemów Jądrowych
05-400 Otwock-Świerk
/799-615, 799-831/

Janusz Zalewski
Instytut Energii Atomowej
05-400 Otwock-Świerk
/799-516/

Streszczenie. Poniższy opis ma na celu przedstawienie właściwości, funkcji oraz zasad działania systemu Multibus-II, i stanowi rozwinięcie publikacji /1/. Wskutek ograniczenia objętości tekstu, niektóre informacje zredukowano do niezbędnego minimum, dlatego do zrozumienia całości konieczne jest wcześniejsze zapoznanie się z treścią wymienionego artykułu.

1. OGÓLNE WŁAŚCIWOŚCI SYSTEMU

W wieloprocessorowym systemie Multibus-II przewidziano pięć specjalizowanych magistrali umożliwiających bezpośrednią, niezależną i jednoczesną komunikację pomiędzy jego elementami. Wybór liczby i konfiguracji wykorzystywanych magistrali zależy od bezpośredniego użytkownika, dzięki czemu możliwe jest obniżenie kosztów systemu oraz optymalne rozwiązanie architektury.

Poszczególne magistrale są dostosowane do wykonywania określonych funkcji, np. magistrala równoległa PSB zapewnia optymal-

ną komunikację pomiędzy procesorami, magistrala lokalna LBX jest dostosowana do bardzo szybkiej komunikacji procesorów z pamięcią, a magistrala szeregową służy do wolnej komunikacji między procesorami umieszczonymi w jednej lub kilku kasetach. W systemie Multibus-II mogą również występować dwie magistrale zdefiniowane w systemie Multibus-I:

- magistrala uzupełniająca SBX przeznaczona przede wszystkim do rozszerzenia możliwości komputera lub bloku systemu przez bezpośrednie dołączenie go do urządzeń zewnętrznych za pomocą specjalizowanych układów umieszczonych na oddzielnej płycie,
- magistrala bezpośredniego dostępu DMA dostosowana do wymiany informacji z szesnastoma różnymi urządzeniami o pojemności pamięci wewnętrznej 16 M bajtów.

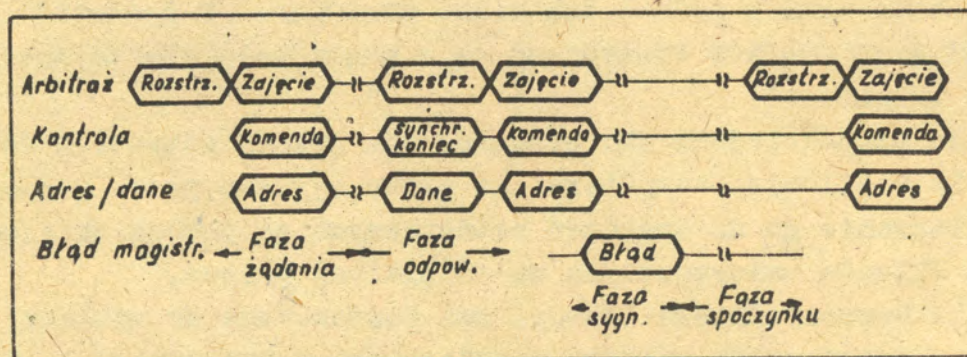
2. PROTOKÓŁ MAGISTRALI RÓWNOLEGŁEJ

W protokole magistrali równoległej zdefiniowano trzy rodzaje cykli: arbitrażu, przesyłania i błędu magistrali.

Cykl arbitrażu składa się z dwóch faz: fazy rozstrzygnięcia dostępu i fazy zajmowania magistrali. Faza rozstrzygnięcia jest określona przez odcinek czasu, w którym wszystkie ubiegające się o dostęp bloki, zgodnie z algorytmem arbitrażu, rozstrzygają między sobą, który z nich otrzyma przydział magistrali w chwili, gdy obecny właściciel ją zwolni. Blok, który wygra arbitraż oraz uzyska dostęp do magistrali, rozpoczyna fazę zajmowania, stając się właścicielem magistrali. Następnie rozpoczyna się cykl przesyłania, podczas którego układy arbitrażu przechodzą do następnej fazy rozstrzygnięcia /wybór następnego właściciela magistrali/, która trwa aż do zakończenia bieżącego cyklu przesyłania.

Cykl przesyłania rozpoczyna się od fazy wywołania, podczas której właściciel magistrali umieszcza na niej informacje dotyczące adresu i rozkazu. Informacje te adresują blok /lub bloki/ odpowiedzi, określają rodzaj operacji oraz przestrzeni adresowej. Faza wywołania trwa przez jeden cykl zegara magistrali. Podczas fazy odpowiedzi, następującej natychmiast po fazie wywołania, blok odpowiedzi wykonuje rozkaz z fazy wywołania. W fazie odpowiedzi, zarówno blok wywoławczy jak i blok odpowiedzi wykorzystują sygnały potwierdzeń, synchronizujące sekwencję przesyłania da-

nych. Ostatni cykl przesyłania jest sygnalizowany przez blok wywoławczy. Podczas ostatniego przesłania blok wywoławczy zwalnia magistralę, umożliwiając następnemu właścicielowi natychmiastowe jej przejęcie.



Rys. 1. Kolejność cykli magistrali równoległej.

Jeżeli blok wykryje błąd podczas cyklu przesyłania, natychmiast rozpoczyna cykl błęd magistrali. Cykl ten rozpoczyna się od fazy sygnalizacji, podczas której blok wykrywający błąd aktywizuje odpowiednie linie błęd magistrali. Powoduje to zakończenie cyklu arbitrażu lub przesyłania we wszystkich blokach w kasecie. Następnie rozpoczyna się faza spoczynkowa, podczas której żaden z bloków nie aktywizuje żadnej z linii magistrali równoległej przez ustalony z góry odcinek czasu. Pozwala to na ustalenie na liniach magistrali poziomów spoczynkowych napięć przed wznowieniem normalnej pracy.

Poszczególne cykle magistrali zilustrowano na rys. 1.

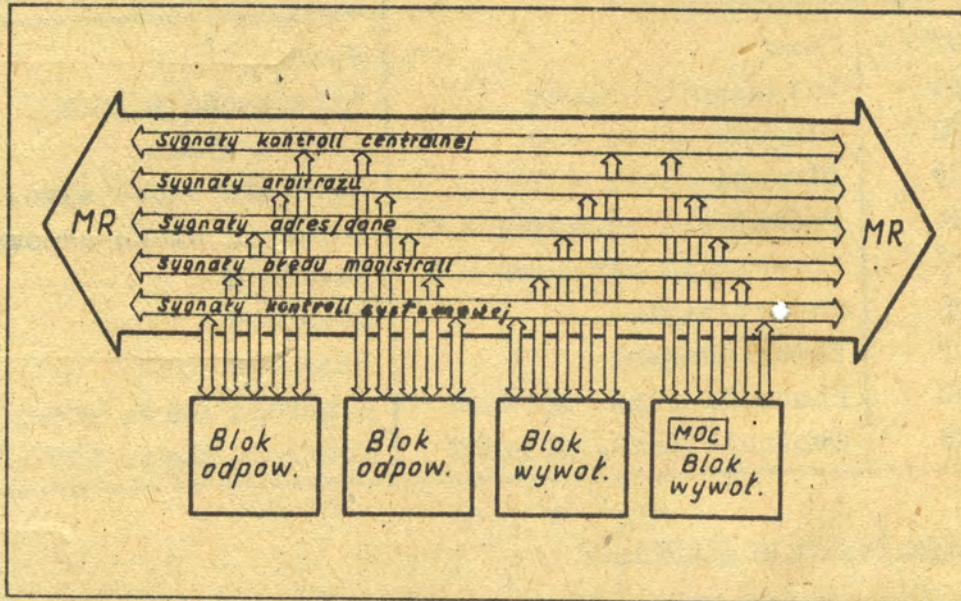
3. STRUKTURA SYGNAŁÓW MAGISTRALI RÓWNOLEGŁEJ

Na magistrali równoległej można wyróżnić pięć grup sygnałów /rys. 2/, za pomocą których realizuje się protokół wymiany informacji/poniżej, gwiazdka przy nazwie sygnału oznacza, że jedynka logiczna odpowiada stanowi niskiemu linii/.

Sygnały arbitrażu.

Sygnały arbitrażu - $BREQ^*$, $ARB5^*$.. $ARB0^*$ - określają, który blok uzyska przydział magistrali. Blok żądający dostępu do magistrali aktywizuje linię $BREQ^*$ oraz wysyła na linie $ARB4^*$.. $ARB0^*$

swój numer poziomu priorytetu. Linia ARB5^z służy do wyróżnienia jednego z dwóch algorytmów arbitrażu - normalnego lub priorytetowego.



Rys. 2. Struktura sygnałów magistrali równoległej.

Sygnały adresowe i danych.

Linie AD31^z..AD0^z przewodzą multipleksowane sygnały zmieniające znaczenie w zależności od fazy cyklu przesyłania. Podczas fazy wywołania linie te zawierają adres, do którego lub z którego nastąpi przesłanie danych. Podczas fazy odpowiedzi przesyła się po nich dane. Linie aktywizowane są przez blok wywoławczy podczas operacji zapisu, a przez blok odpowiedzi podczas operacji odczytu. Linie PAR3^z..PAR0^z służą do przesyłania sygnałów kontroli parzystości dla poszczególnych czterech bajtów linii AD.

Sygnały kontroli systemowej.

Sygnały SC9^z..SC0^z służą do określenia rozkazu lub stanu, zależnie od fazy cyklu przesyłania /tab. 1/. Podczas fazy wywołania blok wywoławczy aktywizuje wszystkie linie SC9^z..SC0^z. Zawierają one kod rozkazu dla bloku odpowiedzi. Podczas fazy odpowiedzi, blok wywoławczy aktywizuje linie SC8^z oraz SC3^z..SC0^z, synchronizując cykl przesyłania oraz generując informacje kontrolne. Blok odpowiedzi używa pozostałych linii SC do synchronizacji cyklu prze-

syłania oraz do określenia stanu.

Tab. 1. Funkcje sygnałów kontroli systemowej.

Sygnal	Faza wywołania	Faza odpowiedzi
SC0	Faza	Faza
SC1	Wyłączość - LOCK	Wyłączość - LOCK
SC2	Długość słowa - 0	Koniec cyklu
SC3	Długość słowa - 1	Gotowość bloku wywoław.
SC4	Przestrzeń adresowa - 0	Gotowość bloku odpowiedzi
SC5	Przestrzeń adresowa - 1	Błąd - 0
SC6	Odczyt/zapis	Błąd - 1
SC7	Zarezerwowane	Błąd - 2
SC8	Kontrola parz. SC7-SC4	Kontrola parz. SC7-SC4
SC9	Kontrola parz. SC3-SC0	Kontrola parz. SC3-SC0

Sygnaly błędu magistrali.

Blok, który wykrył błąd parzystości na liniach AD lub SC, lub - niemożliwy do poprawienia błąd pamięci, aktywizuje linię BUSERR^{##}. W sytuacji, gdy sygnały synchronizujące cykl przesyłania /SC3 i SC4 w fazie odpowiedzi/ nie przychodzą mimo upływu ustalonego czasu, moduł obsługi centralnej /1/ aktywizuje linię TIMOUT^{##}. Sygnaly BUSERR^{##} i TIMOUT^{##} są odbierane przez wszystkie bloki w kasecie.

Sygnaly sterowania centralnego.

Do tej grupy zaliczane są następujące sygnały, generowane głównie przez moduł obsługi centralnej:

- RST^{##}, sygnał inicjowania systemu,
- RSTNC^{##}, wysyłany przez każdy blok /obwód z otwartym kolektorem/, którego wewnętrzne inicjowanie trwa dłużej niż przewiduje sygnał RST^{##}; dzięki sumowaniu zachodzącemu na linii RSTNC^{##}, sygnał ten trwa do chwili, gdy najpowolniejszy z bloków zostanie zainicjowany,
- DCLOW^{##}, ostrzega wszystkie bloki przed zanikiem zasilania,
- PROT^{##}, generowany z niewielkim opóźnieniem w stosunku do DCLOW; informuje wszystkie urządzenia, że powinny ignorować zmiany stanu magistrali równoległej, gdyż są one skutkiem spadku napięcia,

- $BLCK^{\#}$, zegar systemowy, którego opadające zbocze stanowi sygnał odniesienia dla całego systemu,
- $CCIK^{\#}$, dodatkowy zegar o częstotliwości dwa razy większej niż $BLCK^{\#}$,
- $LACHn^{\#}$, służy blokowi do zarejestrowania kodu stanowiska n, który jest przesyłany liniami $ARB4^{\#}..ARB0^{\#}$ podczas inicjowania systemu; na każdym stanowisku linia $LACHn^{\#}$ jest połączona z linią AD o tym samym numerze /W czasie, gdy sygnał RST jest aktywny, moduł obsługi centralnej wysyła kolejno kody stanowisk na linie $ARB4^{\#}..ARB0^{\#}$, aktywując jednocześnie odpowiednie linie AD. Inicjowany blok pobiera ten kod w chwili, gdy sygnał $LACHn^{\#}$ przechodzi na poziom niski/

4. SZCZEGÓŁY ARCHITEKTURY

W systemie zdefiniowano cztery niezależne przestrzenie adresowe, odgrywające różną rolę:

- przestrzeń pamięci /32 bity adresowe/, służąca do przechowywania danych i programów /w postaci słów 8-, 16-, 24- i 32-bitowych/
- przestrzeń wewnętrzną /16 bitów/, przeznaczona do identyfikowania bloków, konfigurowania systemu i do diagnostyki stanu,
- przestrzeń wejścia-wyjścia /16 bitów/, służąca do komunikacji z urządzeniami zewnętrznymi, jak sterowniki komunikacyjne i pamięci masowe, przy użyciu słów 8-, 16-, 24- i 32-bitowych,
- przestrzeń komunikatów /8 bitów/, przeznaczona do komunikacji międzymodułowej i międzyprocesorowej.

Dostęp do wszystkich przestrzeni jest określony tylko dla magistrali równoległej /linie SC4-SC5/. Magistrala lokalna ma dostęp do pierwszych dwóch przestrzeni, a magistrala szeregową - tylko do przestrzeni komunikatów.

Przeźródźń wewnętrzną umożliwia 16-bitowe adresowanie 8-bitowych rejestrów. Na numer stanowiska bloku przeznaczone jest 5 bitów, a na numer rejestru wewnątrz bloku - 9 bitów adresu; dwa najmniej znaczące bity adresu są zawsze równe zeru. Początkowe spośród 512 rejestrów bloku mają funkcje ustalone przez producenta, np.:

- 0-1 - numer producenta /obowiązkowy/,
- 2-3 - typ bloku,
- 4-5 - wersja,

6-7 - adres początkowy pamięci,

8 - pojemność pamięci,

9-511 - inne funkcje.

Stanowiska magistrali równoległej są numerowane od 0 do 19, a magistrali lokalnej - od 24 do 29. Numery stanowisk są przypisywane blokom po włączeniu zasilania.

Przestrzeń pamięci musi być podzielona przez projektanta na dwa rozłączne obszary, przyporządkowane odpowiednio - magistrali równoległej i magistrali lokalnej /używającej do adresowania jedynie 26 najmniej znaczących bitów/.

W przestrzeni komunikatów zachodzi wymiana informacji w zestawach wieloprocessorowych, polegająca na przesyłaniu komunikatów w formie pakietów. Pakiet jest grupą 4-32 bajtów, stanowiących komunikat lub jego część. Cztery pierwsze bajty mają ustalone znaczenie:

0 - adres przeznaczenia,

1 - adres nadawczy,

2 - typ pakietu,

3 - charakterystyka pakietu.

Pozostałe bajty pakietu zawierają dane /w segmentach po 4 bajty/. Dopuszcza się pakiety sześciu typów. Pięć z nich odpowiada tzw. komunikatom niezapowiedzianym /mającym charakter przerwań/, a szósty - zapowiedzianym /zbliżonym do programowego przesyłania danych/:

1/ zwykły komunikat niezapowiedziany /00_H/, stanowiący typowe przerwanie /przesłanie wyłącznie adresu lub adresu z danymi/,

2/ rozgłaszany komunikat niezapowiedziany /01_H/, tj. adresowany do wszystkich bloków /odpowiada mu użycie adresu przeznaczenia FF_H/,

3/ komunikat zerowania /10_H/, służący do zerowania wybranego bloku na magistrali,

4/ komunikat żądania bufora /24_H/ i komunikat przyznania bufora /35_H/, służące do uzgodnienia transmisji komunikatów zapowiedzianych.

Wyróżnia się trzy rodzaje operacji rozkazowych:

- pojedyncze, polegające na jednokrotnym przesłaniu danych pod jeden adres, możliwe do wykonania we wszystkich przestrzeniach adresowych,

- sekwencyjne, polegające na blokowym przesyłaniu danych, możliwe do wykonania w przestrzeni wejścia-wyjścia /bez inkrementacji adresu/, przestrzeni pamięci /z inkrementacją adresu/ i przestrzeni komunikatów. /tylko na magistrali równoległej/,
 - rozgłaszanie, polegające na przesyłaniu danych do wszystkich bloków, możliwe do wykonania tylko w przestrzeni komunikatów.
- We wszystkich trzech pierwszych przestrzeniach możliwy jest zapis i odczyt danych, natomiast w przestrzeni komunikatów możliwe są tylko operacje zapisu.

Podstawowe informacje dotyczące architektury systemu zgrupowano w tabeli 2.

Tab. 2. Podstawowe właściwości architektury Multibus-II.

Przestrzeń	Wewnętrzna	Pamięci	We-wy	Komunikatów
Adres	16 bitów	32 bity /26 bitów magistrali lokalnej/	16 bitów	8 bitów
Dane	8 bitów	8/16/24/32 bity	8/16/24/32 bity	32 bajty
Rodzaje operacji	Pojedyncze	Pojedyncze, sekwencyjne z inkrement.	Pojedyncze, sekwencyjne bez inkrem.	Pojedyncze, sekwencyjne /tylko dla magistrali równoległej/ rozgłaszanie
Zapis/odczyt danych	Tak	Tak	Tak	Tylko zapis
Magistrala	Równoległa, lokalna	Równoległa, lokalna	Równoległa	Równoległa, szeregowa

5. SPECJALIZOWANE UKŁADY VLSI DO REALIZACJI SYSTEMU

Do realizacji systemu Multibus-II firma Intel przewiduje zastosowanie dwóch specjalizowanych układów VLSI wykonanych w technologii CMOS, zasilanych napięciem +5 V, w obudowie kwadratowej 84-końcówkowej, zajmujących powierzchnię 0,96x0,96 cala.

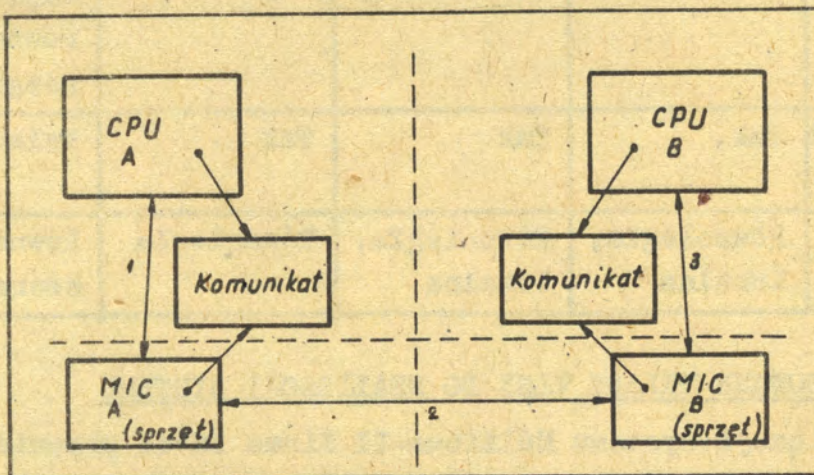
Sterownik magistrali /ang. Bus Arbiter/Controller, BAC/ wykonuje następujące funkcje:

- realizuje protokół arbitrażu,
- przeprowadza sterowanie przesyłaniem danych i adresów po magistrali,
- generuje sygnały kontrolne i przeprowadza kontrolę parzystości,
- umożliwia transmisję blokową.

Sterownik przerw /ang. Message Interrupt Controller, MIC/ umożliwia blokom systemu:

- obsługę przerw,
- odbiór i wysyłanie przerw metodą transmisji sekwencyjnej,
- realizację połączeń pośredniczących między komputerem wewnętrznym bloku a magistralą równoległą,
- specjalną obsługę przerw przy użyciu pamięci FIFO oraz nieprzewidzianą w transmisji sekwencyjnej.

Przykład użycia sterownika MIC przedstawiono na rys. 3. Po otrzymaniu od procesora A parametrów przesyłanego komunikatu, sterownik MIC samodzielnie realizuje proces przesyłania kopii komunikatu. Protokół wymiany komunikatów, przedstawiony w opisie rysunku, umożliwia sterownikom MIC korzystanie z właściwości magistrali równoległej /jak np. 32-bitowe słowo czy transmisja sekwencyjna/ niezależnie od rodzaju procesora. Nawet procesory ośmiobitowe lub procesory wejścia-wyjścia mogą, przy użyciu układów MIC, przesyłać komunikaty w pełni wykorzystujące właściwości magistrali.



Rys. 3. Przesyłanie komunikat w systemie Multibus-II /1. Procesor A żąda przesłania komunikatu, 2. Uzgadnianie przesyłania - niezależne od obu procesorów, 3. Procesor B przydziela pamięć na komunikat oraz otrzymuje potwierdzenie kompletności nadesłanego komunikatu

6. BIBLIOGRAFIA

- /1/ Trechciński R.: Multibus-II - wieloprocesorowy system modułarny. Informatyka, nr 6, str. 21-23, 1984
- /2/ Intel Corp.: Multibus II Bus Architecture Specification. Santa Clara /CA/, September 1983
- /3/ Intel Corp.: Multibus II Design Seminar. Frankfurt/Muenchen/Juelich, May 1984
- /4/ Beaston J.: Multiprocessor Bus Is Ready to Meet 32-bit Applications of Future. Electronics, Vol. 57, No. 6, pp. 126-131, 22 March 1984
- /5/ Beaston J.: Multibus II - An Architecture for 32-bit Systems. Electronic Engineering, Vol. 56, No. 687, pp. 47-51, March 1984
- /6/ Packer S.J.; S.D. Kassel, R.S. Tetrick: Multiple Buses Satisfy Speed, I/O Needs of Future. Electronic Design, pp. 103-108, 26 January 1984
- /7/ Tucker S.: Multibus Version 2. Systems International, pp. 51-53, January 1984
- /8/ Wilson D.: Multibus II - A Top Down Approach. Digital Design, pp. 71-77, December 1983

VI Szkoła Mikroprocesorowa CSI/PTI
Łódź, grudzień 1984

WPROWADZENIE DO EMULACJI
MIKROPROCESORÓW 16-BITOWYCH
NA PRZYKŁADZIE PROCESORA 8086

mgr inż. Krzysztof Buchała
Zakład Systemów Automatyki Kompleksowej PAN
ul. Bałtycka 5
44-100 Gliwice
tel. 31-08-11/wewn. 179

1. WSTĘP.

Istniejące definicje emulacji można by sprowadzić do najogólniejszego określenia emulacji jako symulacji normalnej pracy mikroprocesora w układzie uruchamianym poprzez system zawierający mikroprocesor otoczony układami umożliwiającymi dodatkowo śledzenie jego pracy i wpływanie na jej przebieg. Układy realizujące te zadania /tzw. emulatory układowe/ dostarczają narzędzi zarówno urządzeńowych /dostęp do zasobów systemu prototypowego, emulacje pamięci i podsystemu we/wy/ jak i programowych /obserwacja i zmiana stanu mikroprocesora emulowanego, modyfikacje komórek pamięci prototypu i pamięci emulowanej, itp./ służących integracji urządzeniowo-programowej systemu uruchamianego [1]. Tej klasie urządzeń stawiane są następujące podstawowe wymagania:

- jak najdokładniejsza emulacja samego procesora /odpowiednia charakterystyka elektryczna i dynamiczna emulatora/,

- nie ograniczanie zasobów systemu prototypowego - SP,
- emulacja procesora w czasie rzeczywistym,
- śladowanie procesorów SP i zatrzymywanie ich po wykryciu warunku przynależności procesu do określonej klasy procesów,
- dostęp do zasobów wewnętrznych procesora i wymuszanie nowego stanu procesora emulowanego.

Stopień spełnienia tych zadań wyznacza podstawową linię podziału wszystkich istniejących emulatorów. Wśród powyższych wymagań możnaby wydzielić zadania realizujące emulację procesora /emulowany w sensie "równy" lub "nie gorszy"/ oraz zadania związane z problematyką dzielenia zasobów i funkcjami usługowymi.

W przypadku emulatorów mikroprocesorów 8-bitowych te pierwsze zwykle nie stwarzały większych problemów, a różnicowanie postępowało w miarę wzrostu kompleksowości całych systemów mikroprocesorowych. Emulatory mikroprocesorów 16-bitowych osiągają kolejny szczebel złożoności nie tylko ze względu na znacznie większą kompleksowość i potencjał funkcjonalny systemów uruchamianych, ale i samych mikroprocesorów tej generacji.

2. EWOLUCJA EMULATORÓW MIKROPROCESORÓW 8-BITOWYCH.

Analiza linii rozwojowej emulatorów pozwala zaobserwować wyraźne tendencje w dziedzinie konstrukcji tego typu urządzeń. Tendencje te istnieją niezależnie od rodzaju i typu mikroprocesora emulowanego, a więc i od wspomnianych uprzednio stymulatorów rozwoju emulatorów. Za główne trendy wyznaczające kierunek ewolucji tych urządzeń możnaby uznać:

- maksymalizację "przeźroczystości" /w sensie elektrycznym, czasowym, funkcjonalnym i dostępu do zasobów/ emulatora od strony systemu użytkowego,
- zwiększanie "elastyczności" emulatora, rozumianej głównie jako możliwości wprowadzania zmian i adaptacji emulatora do konkretnej aplikacji systemu uruchamianego,

- zwiększanie uniwersalności emulatora, prowadzące do wyodrębnienia oddzielnych modułów urządzeńowych i programowych wymiennalnych przy zmianie procesora emulowanego,
- minimalizowanie wymaganej od użytkownika wiedzy na temat emulowanego procesora,
- rozszerzanie możliwości funkcjonalnych emulatora.

Droga do zaspokojenia wymienionych dążeń prowadzi poprzez następujące generacje emulatorów:

Emulatory jednoprocessorowe /tzw. dedykowane/, w których procesor systemu nadrzędnego emulatora /zwanego dalej systemem wspomagającym - SW/ posiada bezpośredni dostęp do SP i sam wykonuje wszystkie zadania emulacji. Musi on w związku z tym być tego samego typu co procesor emulowany. Taka architektura ogranicza jednak znacznie zgodność charakterystyki mikroprocesora emulowanego z rzeczywistym oraz rezerwuje określone zasoby /np. przerwania/ czy też obszary przestrzeni adresowej procesora emulowanego przeznaczając je dla pamięci i portów wejścia/wyjścia realizujących zadania emulacji. Typowym przedstawicielem tej klasy emulatorów jest system "MM80" firmy Ramtek Corp. [4]. Oczywiście jest ograniczony zakres zastosowań tej klasy emulatorów powodowany wymuszeniem z góry dostosowania prototypu do warunków emulacji.

Emulatory dwuprocessorowe /tzw. master-slave/ zawierające procesor systemowy SW /niezmienny/ oraz drugi - odrębny procesor emulowany, będący procesorem tego samego typu co procesor systemu uruchamianego. W tej klasie można wydzielić dwie podgrupy:

- a/ emulatory ze wspólną magistralą dla obydwu procesorów,
- b/ emulatory z oddzielną magistralą procesora systemu bazowego i procesora emulowanego.

Przedstawicielem pierwszej grupy systemów może być system "UMDS" zaprojektowany w Instytucie Fizyki Węgierskiej Akademii Nauk [5].

Systemy te są z pewnością krokiem do przodu w stosunku do emulatorów poprzedniej klasy, głównie dzięki "odciążeniu" procesora emulowanego od zadań systemowych oraz ujednoczeniu oprogramowania systemowego. Charakteryzują się one jednak trzema istotnymi ograniczeniami:

- tylko jeden z procesorów może być w danym momencie aktywny na magistrali,
- układy sterujące emulacją /i dostępem do pamięci/ wprowadzają istotne opóźnienia przy niektórych operacjach, jak np. przy dostępie do pamięci emulowanej,
- programy systemowe zajmują pewne zasoby pamięci i peryferii czyniąc je niedostępnymi dla procesora emulowanego.

Koncepcja emulatorów z oddzielnymi magistralami pozwala na usunięcie tych ograniczeń, umożliwiając równocześnie łatwe konfigurowanie emulatora z oddzielnych bloków funkcjonalnych z wyraźnym wydzieleniem modułów zależnych od typu mikroprocesora emulowanego. Przykładem tego typu emulatorów są emulatory ICE przyłączane do systemu Intellec MDS firmy Intel [1]. Efektywność tych emulatorów jest nieraz dwukrotnie większa dzięki współbieżności emulacji i procesów systemu nadrzędnego. Przy zmianie procesora SP wymienia się tu cały moduł emulatora /bazujący na magistrali tego procesora/. Wadą tego rozwiązania jest między innymi niemożność emulacji pamięci w czasie rzeczywistym.

Kolejny poziom ewolucyjny reprezentuje inny przedstawiciel emulatorów 8-bitowych tej klasy - system 64000 LDS firmy Hewlett Packard [3]. Standardowa magistrala emulatora powoduje, że wymienna jest tu tylko sonda emulowanego procesora oraz pół-universalny moduł sterowania emulacją.

Przeniesienie programu emulatora do pamięci RAM /tzw. pamięci tła - ang. background memory/ wydatnie zwiększa w tym przypadku atrybut elastyczności systemu.

Zasada działania większości emulatorów opiera się na wykonywaniu przez procesor emulowany instrukcji "usługowych" emulatora, nie znajdujących się w programie prototypu, a pobieranych z oddzielnego obszaru pamięci. W dotychczas omówionych emulatorach najczęściej stosuje się pracę mikroprocesora emulowanego w jednym z dwóch trybów [1]:

- trybie EMULATOR, w którym procesor staje się procesorem systemu prototypowego SP, wykonującym programy prototypu,
- trybie MONITOR, w którym staje się procesorem emulatora wykonując procedury dostępu do stanu i zasobów SP oraz procedury sterowania stanem SP.

Wspomniane wyżej systemy mają stosunkowo rozbudowane układy przełączania tych trybów i złożone sterowanie przebiegiem emulacji. Należące również do tej klasy emulatory działające na zasadzie forsowania instrukcji wirtualnej [1] pozwalają na znaczne uproszczenie konstrukcji przy zachowaniu tego samego poziomu przejrzystości, elastyczności i uniwersalności emulatora. Przedstawicielem tej grupy jest system "RTDS-8" opracowany w Polskiej Akademii Nauk [6].

Przedstawione dotąd klasy emulatorów były odpowiednie i wystarczające dla uruchamiania systemów opartych o mikroprocesory 8-bitowe. Jako kontynuacja tej linii rozwojowej pojawiły się w końcu wielomagistralowe emulatory wieloprocessorowe, których konfiguracja, jak się później okazało, równie dobrze pozwala na realizację zadań emulacji systemów opartych na mikroprocesorach 16-bitowych. Typowym reprezentantem tego typu emulatorów jest Slave Emulator Control Unit /SECU/ firmy Gen Rad [8]. Jest to system bazujący na oddzielnych autonomicznych modułach emulatorów, z których każdy posiada uniwersalną magistralę skupiającą poszczególne bloki funkcjonalne emulatora /uniwersalne i wymienne/. System SECU pozwala na równoczesną i niezależną pracę do ośmiu emulatorów 8- lub 16-bitowych, a więc emulację systemu wieloprocessorowego.

3. EMULATORY NOWEJ GENERACJI.

Jak pokazano wyżej, emulatory mikroprocesorów 16-bitowych stanowią kontynuację pewnej linii rozwojowej urządzeń tego typu. Jest to nie tylko efektem dążeń do konstruowania coraz doskonalszych układów uruchamiania systemów mikroprocesorowych ale wymogiem narzuconym przez nową generację mikroprocesorów /są to układy o zupełnie odmiennych budowach i zasadach działania, produkowane według nowych technologii, stosowane w znacznie bardziej złożonych i wyszukanych aplikacjach mikroprocesorowych/. Przykładem wspomnianej ciągłości może być nowa generacja emulatorów systemu HP 64000 LDS, powstała na bazie poprzedniej po wprowadzeniu do niej odpowiednich zmian i rozszerzeń [7].

Realizacja zadania emulacji mikroprocesorów 16-bitowych wiąże się z rozwiązaniem problemów, które dotychczas nie występowały w ogóle albo pojawiały się niezbyt jaskrawie.

Pierwszą oczywistą zmianą w konstrukcji emulatorów procesorów 16-bitowych było uwzględnienie rozszerzenia magistrali danych i adresów. W przypadku magistrali danych pojawia się tu problem jej multipleksowania dla operacji bajtowych i słowowych. Zwiększenie linii adresowych emulatorów musi zapewniać pełny obszar adresacji takich procesorów jak Motorola 68000 /16 M bajtów, adresacja bez segmentacji/ czy Z 8001 /8 M bajtów, adresacja z segmentacją/.

Weryfikacji musiano również poddać sposób zatrzymywania procesora i przechodzenia do trybu MONITOR.

Mikroprocesory 8-bitowe charakteryzowały się sekwencyjnym wykonywaniem instrukcji po cyklu maszynowym, a stan magistrali odzwierciedlał wewnętrzne procesy mikroprocesora. Pozwalało to na stosunkowo łatwe wykrywanie cyklu pobierania kodu instrukcji /cykle OP CODE FETCH/ i przełączanie trybu pracy procesora drogą forsowania instrukcji skoku do programu monitora /ICE, UMDS/ lub forsowania adresu tego programu /ang. address jamming/ /HP 64000 LDS/.

W przypadku procesorów 16-bitowych taki mechanizm jest nieodpowiedni. Cykle /tzw. cykle magistrali - ang. bus cycles/ pobierania kolejnych bajtów/słów instrukcji są przeplatane cyklami wykonawczymi instrukcji pobranych wcześniej. Wynika to z często stosowanej architektury wewnętrznej mikroprocesorów, opartej na dwóch niezależnych interakcyjnych modułach funkcjonalnych /8086/: module wykonawczym /ang. execution unit/ i module interfejsu z magistralą /ang. bus interface unit/. Taka koncepcja ściśle wiąże się z istnieniem wewnętrznej kolejki rozkazów stanowiącej bufor pomiędzy obydwojma modułami. Wprowadzenie kolejki pozwala na tzw. wyprzedzone pobieranie rozkazów /ang. prefetching/ istotnie zwiększające prędkość działania procesora /8086, 8088, Motorola 68000/. Rozwiązanie takie stanowi jednak utrudnienie z punktu widzenia konstrukcji emulatorów, związane głównie z zatrzymywaniem procesora w określonym momencie oraz śladowaniem procesów SP. wykrycie warunku zatrzymania lub śladowania na magistrali zewnętrznej procesora wymaga śledzenia kolejki wewnętrznej w celu określenia, czy warunek ten dotarł do modułu wykonawczego /został spełniony/, czy też sterowanie procesem zostało przeniesione w inne miejsce programu z ominięciem tego warunku /gdy w kolejce znajdowała się instrukcja zmieniająca licznik programu lub wystąpiło zdarzenie zewnętrzne prowadzące do wyzerowania kolejki/. Nasuwa się tu od razu spostrzeżenie, że samo zatrzymanie procesora musi się odbywać przy pustej kolejce. Dlatego też dla tych celów wykorzystuje się najwyższe priorytetowo lub niemaskowalne przerywania oraz linię zerowania procesora emulowanego /RESET/. Wiąże się to jednak z monopolizacją przez emulator pewnych zasobów /obszarów zastrzeżonych w pamięci/ SP, co w świetle tego co powiedziano uprzednio, stanowi jego niewątpliwą wadę. Ograniczenie to próbuje się usunąć różnymi sposobami. Jedną z metod jest wyodrębnienie i zawężenie przedziałów czasu, w których poziomy te są monopolizowane. Wykorzystuje się tu też często indywidualne własności poszczególnych mikroprocesorów. Na przykład w emulatorze mikroprocesora MOTOROLA 68000 systemu HP 64000 LDS [9] korzysta się z trybu pracy procesora zwanego "non - automatic - vectoring", pozwalającego na używanie w wymienionym celu

najwyższego priorytetowo poziomu przerwań /INT7/ z zewnętrzną generacją numeru wektora przerwania, deklaruwalnego przez użytkownika w fazie inicjacji emulatora systemu LDS.

Kolejka wewnętrzna istotnie komplikuje również problem śladowania w czasie rzeczywistym, szczególnie śladowania selektywnego. Układy śladujące są tu już z reguły znacznie bardziej złożone, gdyż muszą uwzględniać nie tylko możliwość "nie dotarcia" warunków śladowania do wykonania, ale i fakt wzajemnie asynchronicznej pracy obydwu modułów wewnętrznych procesora. Moduł interfejsu z magistralą /zapełniają^{cy} kolejkę/ pracuje synchronicznie z cyklami magistrali, moduł wykonawczy zaś /pobierający instrukcje z kolejki/ jest synchronizowany z taktami zegara podstawy czasu procesora. W przypadku procesora 8086 oznacza to, że dla danego stanu linii $\overline{S2}$, $\overline{S1}$, $\overline{S0}$ może nastąpić trudna do przewidzenia ilość zmian stanów na liniach QSO, QS1.

Warunki śladowania i zatrzymania dzielą się na tzw. warunki pobrania kodu instrukcji /ang. execution breakpoints/ i warunki wykonawcze /ang. non-execution breakpoints/. Wykrycie na magistrali warunku pierwszego typu wymaga aktywacji układu śladowania kolejki wewnętrznej, warunki wykonawcze natomiast mogą bezpośrednio generować przerwania do procesora /przerwania są akceptowane pod koniec aktualnie wykonywanej instrukcji, są więc synchronizowane z pracą modułu wykonawczego/.

Istotną zmianą w stosunku do mikroprocesorów 8-bitowych jest nowa klasyfikacja typów cykli magistrali. Przykładowo w procesorze 8086 cykle dostępu do pamięci programu noszą nazwę cykli CODE FETCH i są między sobą nierozróżnialne. Związane z tym trudności w wyodrębnieniu cyklu pobrania pierwszego bajtu /lub słowa/ instrukcji są powodem, że metoda forsowania instrukcji wirtualnej, chociaż teoretycznie możliwa, nie została zastosowana w emulatorze przedstawionym w dalszej części niniejszego opracowania.

Emulacja mikroprocesorów 16-bitowych wymaga również zmiany koncepcji alokacji programu emulatora. Przełączanie z pamięci pierwszoplanowej /ang. foreground memory/, z której procesor emulowany pobierał instrukcje w fazie EMULATOR, na pamięć tła

zawierającą program emulatora jest tutaj o wiele bardziej skomplikowane /wyprzedzone pobieranie instrukcji, dynamiczna zmiana trybów pracy niektórych typów mikroprocesorów/. Postanowiono przenieść program emulacji do pamięci pierwszoplanowej umieszczając go w rozłącznym obszarze adresowym z obszarem programu użytkownika. Najczęściej program emulatora znajduje się w odpowiednio skonfigurowanej pamięci emulowanej, której mechanizm selekcji jest znacznie prostszy niż pamięci tła. Mając na uwadze o wiele większą pojemność adresową mikroprocesorów 16-bitowych praktyka ta nie ogranicza prawie wcale zasobów SP, natomiast wykazuje kilka użytecznych własności. Podstawową zaletą jest modularność programu emulatora dostarczanego w wersji źródłowej na odrębnym nośniku /programy pamięci tła były zwykle bardziej monolityczne/. Użytkownik przeprowadza cały proces kompilacji, dołączenia do oprogramowania SP i alokacji tego programu mogąc równocześnie go modyfikować lub np. zawęzić stosownie do własnych potrzeb. Ponieważ pamięć emulowana oprócz programów użytkownika zawiera również program emulatora, musi ona w związku z tym być większa niż dotychczas. Typowa struktura tak ulokowanego programu emulatora posiada następujące moduły: obszar wektorów przerwań i tzw. zdarzeń wyjątkowych, moduł wymiany informacji i komunikacji z systemem nadrzędnym i blok właściwego programu emulacji.

Emulatory nowej generacji wymagają również weryfikacji metody synchronizacji dostępu do pamięci emulowanej przez obydwie procesory systemu uruchamiania /realizacja samej emulacji pamięci nie uległa istotniejszym zmianom/. Szybkość działania procesorów 16-bitowych powoduje, że zajmują one dużą procentowo część cyklu magistrali na dostęp do pamięci. Klasyczne wykorzystanie pamięci dwubramowej czy też metoda przerywania pracy i ponownego restartu procesora emulowanego nie zdają tu często egzaminu. Do komunikacji procesora systemu nadrzędnego z emulatorem wykorzystuje się częściej technikę kradzieży cykli magistrali. Przykładowo we wspomnianym tu już systemie HP-LDS mikroprocesor MOTOROLA 68000 jest wstrzymywany co 500 ms na ok.

500 μ sek /komunikacja z odpytywaniem zamiast stosowanej dotąd często techniki martwej pętli/. Oczywiście użytkownik może zadeklarować pracę w czasie rzeczywistym, wyłączając ten mechanizm aż do wykrycia warunku zatrzymania.

Symulacja we/wy wymaga teraz szczególnie konsekwentnego przestrzegania zasad przyjętej implementacji podsystemu we/wy /urządzenia zadeklarowane jako bajtowe muszą być dostępne na określonej części magistrali, zaś urządzenia słowowe - zawsze na całej/.

Zasadniczym nowym problemem jest zadanie emulacji systemów wieloprocesowych, typowych dla systemów 16-bitowych. Emulatory nowej generacji powinny uwzględniać obecność koprocatora w systemie /np. 8087 dla procesora 8086/. Prowadzi to do następujących rozwiązań konstrukcyjnych:

- 1/ Dopuszcza się emulację pamięci bez separacji magistrali danych SP, umożliwiając w ten sposób monitorowanie operacji na tej pamięci przez ewentualny koprocator. Wymaga to jednak wyprowadzenia dodatkowych sygnałów do SP, zabezpieczających przed zakłócaniem magistrali w tym czasie przez inne moduły pamięci.
- 2/ Umożliwia się dostęp koprocatora /np. w cyklach DMA/ do pamięci emulowanej. W tym celu doprowadza się stroby tej pamięci od wyprowadzeń nieaktywnego procesora w SP oraz dodaje się dodatkowe wyprowadzenia od pamięci emulowanej do wykorzystania w dowolnym miejscu w prototypie /np. sygnał gotowości pamięci emulowanej/.
- 3/ Elastyczność struktury programowej emulatorów pozwala na dołączanie ich do różnego rodzaju systemów operacyjnych.
- 4/ Rozwijają się emulatory wieloprocesorowe/wielomagistralowe /patrz rozdział drugi/.

Warto tu wspomnieć, że nowa generacja systemów 64000 LDS pozwala na podłączenie i równoległą pracę czterech emulatorów.

Wśród nowej problematyki związanej z emulacją mikroprocesorów 16-bitowych szczególnie nacisk kładzie się również na zwiększenie uniwersalności emulatorów. Powinny one przewidywać zarówno bardzo zróżnicowaną strukturę systemów SP jak i uwzględniać możliwość włączania w architekturę specjalizowanych bloków funkcjonalnych /np. Memory Management Units/. Jedną z dróg jest tutaj zwiększanie parametryzacji struktury urządzeniowej i programowej emulatorów. Już takie naszkicowanie niektórych aspektów emulacji mikroprocesorów 16-bitowych może nasunąć pytania dotyczące sposobu i stopnia trudności użytkowania nowych emulatorów. Coraz większy zasób możliwości tych urządzeń przy coraz większej złożoności i różnorodności systemów uruchamianych prowadzi do konieczności stosowania języków symbolicznych /wraz z odpowiednimi kompilatorami/ na poziomie komunikacji operatora z emulatorem. Prace w tym kierunku są obecnie bardzo zaawansowane /coraz więcej emulatorów wprowadza na bieżąco nowe elementy oprogramowania tego poziomu, tworząc np. makra komend, pliki komend itp./.

4. UWARUNKOWANIA EMULACJI MIKROPROCESORA INTEL 8086.

Dokonana wyżej próba naszkicowania głównych aspektów emulacji klasy mikroprocesorów 16-bitowych daje ogólny obraz nowych problemów i sposobów ich rozwiązywania. Oczywistym jest, że każdy typ mikroprocesora oznacza specyficzne, charakterystyczne dla niego cechy, które czasem okazują się stanowić ułatwienie w realizacji określonych zadań emulacji, czasem zaś stają się dodatkowym utrudnieniem dla osiągnięcia idealnej emulacji substytutatywnej. Bazując na informacjach zawartych w [10] omówiono tu krótko wybrane własności procesora 8086, które okazały się szczególnie interesujące z punktu widzenia emulacji. Uzasadnieniem wyboru tego właśnie procesora są:

- po pierwsze: perspektywy dużego rozpowszechnienia aplikacji opartych na odpowiedniku tego procesora /produkowanym w ZSRR/

w naszym kraju w najbliższej przyszłości,

- po drugie: posiadanie przez procesor 8086 większości typowych cech tej klasy procesorów,
- po trzecie: zebranie doświadczeń z pierwszych prób emulacji procesora 8086 przeprowadzonych w Zakładzie Systemów Automatyki Kompleksowej PAN w Gliwicach.

Z punktu widzenia emulacji procesora 8086 istotne są następujące elementy jego architektury wewnętrznej:

- 1/ Istnienie flagi TF, aktywującej tryb pracy po kroku. Zlecenie pracy krokowej w emulatorze wykorzystuje tę własność procesora przy ograniczeniu, że flaga ta w tym czasie jest ignorowana /program prototypu nie może jej zmieniać/.
- 2/ Sześciobajtowa kolejka wewnętrzna definiuje głębokość i sposób śladowania dla układów śledzących operacje dostępu do niej.
- 3/ Rejestry segmentowe /adresacja z segmentacją/ narzucają dwa sposoby formowania adresu fizycznego przez użytkownika /z zadeklarowaniem wartości segmentowej i wartości nakładki /ang. offset/ oraz z wykorzystaniem aktualnej zawartości określonego rejestru segmentowego/.
- 4/ Istnienie modułu wykonawczego /EU/ i modułu interfejsu z magistralą /BIU/ wymaga sprecyzowania sposobów zewnętrznego selektywnego oddziaływania na każdego z nich.

Analiza sygnałów procesora w trybach MINIMUM i MAXIMUM prowadzi do zdecydowania do wyboru konfiguracji maksymalnej dla mikroprocesora sondy emulatora. Dodatkowe zastosowanie tutaj kontrolera magistrali 8288 pozwala na uzyskanie wszystkich niezbędnych strobów pamięci i systemu we-wy, przy równoczesnym dostępie do informacji o rodzaju cyklu magistrali /sygnały $\overline{S2}$, $\overline{S1}$, $\overline{S0}$ / i statusie kolejki wewnętrznej /sygnały QS1, QS0/. Konfiguracja MAXIMUM procesora SW wynika również z przyjętego założenia, że system ten realizuje podstawowe zadania procesu uruchamiania dla dowolnego SP opartego na danym procesorze /a więc dowolnej

konfiguracji procesora SP/, nie może więc w żaden sposób ograniczać struktury i stanów SP. Oczywiście jest, że mikroprocesor emulatora pracujący w trybie minimalnym nie spełniał by tego wymagania.

Linie $\overline{S2}$, $\overline{S1}$, $\overline{S0}$ pozwalają jednoznacznie określić początek i koniec cyklu magistrali.

Sygnały $QS1$, $QS0$ umożliwiają śladowanie procesorów EU niezależnie od działalności BIU /również podczas stanów "wstrzymania" wymiany informacji na magistrali zewnętrznej, jak np. WAIT, HOLDACK, IDLE/.

Multipleksowanie adresów i danych na magistrali wymaga "zatrzas-kiwania" tych pierwszych w określonych momentach cyklu dla potrzeb np. komparacji warunku śladowania czy zatrzymania.

Związane z segmentacją pamięci sygnały statusu $S3$ i $S4$ mogą być wykorzystane w emulatorze jako dodatkowe linie adresowe zwiększając w ten sposób jego możliwości funkcjonalne.

Dezaktywacja sygnału READY /nie później niż 8 ns przed rozpoczęciem fazy T3/ pozwala na zatrzymywanie modułu BIU w bieżącym cyklu w stanie WAIT, a w konsekwencji wymuszenie pracy procesora z pustą kolejką wewnętrzną, co może być bardzo użyteczne dla realizacji niektórych funkcji emulatora /np. forsowania instrukcji czy śladowania selektywnego/.

Organizacja pamięci w systemie 8086 definiuje sposób dostępu do danych. W związku z tym monitorowanie sygnałów $A0$ i BHE pozwala jednoznacznie rozróżniać przesyły słowowe i bajtowe, a często przez to przewidywać rodzaj następnego cyklu magistrali.

Zasady akceptacji i obsługi niemaskowalnego przerwania NMI implikują wybór tego sygnału dla zatrzymywania i przechodzenia procesora emulowanego z trybu EMULATOR do trybu MONITOR.

Rozróżnialność operacji dostępu do urządzeń peryferyjnych /sygnał M/\overline{IO} / pozwala na implementację symulacji /lub emulacji/ podsystemu wejścia-wyjścia.

Zasady synchronizacji przejmowania kontroli nad magistralą /w cyklach DMA/ różnią się dla trybu MINIMUM i MAXIMUM. Ponieważ procesor sondy SW pracuje stale w konfiguracji MAXIMUM konieczna jest transformacja protokołu dostępu do magistrali HOLD/HLDA /trybu minimalnego/ na protokół RQ/GT/RL /trybu maksymalnego/. "Przeźroczystość" emulatora wymaga również ścisłego przestrzegania charakterystyki elektrycznej i czasowej wszystkich sygnałów procesora w cyklach DMA. Istotnym jest tu fakt, że zależnie od trybu pracy procesora ta sama linia może posiadać nieaktywny poziom sygnału albo przyjmować stan wysokiej impedancji /np. wyprowadzenie procesora QS1 / \overline{INTA} //.

Od strony programowej zwraca uwagę duża użyteczność instrukcji łańcuchowych procesora 8086 do realizacji wybranych funkcji emulatora /np. wyświetlanie zawartości całego obszaru pamięci, wypełnianie pamięci wzorcem, itp./.

5. EMULATOR PROCESORA 8086

Zaprezentowana tu problematyka została wykorzystana podczas projektowania w Zakładzie Systemów Automatyki Kompleksowej PAN w Gliwicach emulatora mikroprocesora 8086. Konstruktorzy starali się wybrać najbardziej optymalną dla aktualnych warunków koncepcję, dlatego też nie wszystkie wymienione wyżej aspekty konstrukcji tego typu urządzeń zostały tu wykorzystane. Schemat blokowy sondy i pakietu adaptera przedstawiają rysunki 1 i 2.

Emulator ten przewidziany jest jako podsystem systemu RTDS-8. Program emulatora umieszczony jest w dwubramowej pamięci tła /tzw. pamięci testowej/ o organizacji 4K x 16 bitów /zrealizowanej na elementach szybkiej pamięci statycznej 2147/. Od strony systemu 8086 pamięć ta jest dostępna w obszarze adresowym deklarowanym przez użytkownika /z dokładnością do 4 K/. Procesor systemu bazowego ma dostęp do pamięci testowej w fazie T4 /i ewentualnych następnych fazach biernych - TI/ każdego cyklu magistrali. Przełączanie procesora emulowanego na tryb

pracy MONITOR można przeprowadzić wykorzystując sygnały RESET, NMI lub forsowaną jednobajtową instrukcję INT /w tym ostatnim rozwiązaniu wymusza się pracę procesora z pustą kolejką wewnętrzną/. Wyjście z obszaru pamięci emulatora odbywa się poprzez skok /a więc wyzerowanie kolejki wewnętrznej/ do wyróżnionego miejsca pamięci /MONITOR-EXIT/ gdzie znajduje się uprzednio przygotowana instrukcja skoku do pamięci pierwszoplanowej /kontynuacja procesu SP w trybie EMULATOR/. Rozróżnienie funkcji emulatora realizuje się przez modyfikację petli MONITOR-LOOP.

Taka koncepcja emulatora zupełnie nie ogranicza zasobów SP, generując w tle /tylko w trybie MONITOR/ wektory RESET, NMI, SINGLE STEP i INT. Ładowany przy inicjacji procesu uruchamiania program emulatora bazujący w pamięci tła /tzw. program niskiego poziomu/ ma typową strukturę /opisaną w rozdziale trzecim/ umożliwiającą łatwe wprowadzanie zmian.

Komunikuje się on według określonego protokołu z programem wyższego poziomu, realizowanym przez SW. Ten z kolei jest programem komunikacji z użytkownikiem na poziomie tzw. Języka Opisu Procesu Uruchamiania /z odpowiednim kompilatorem i interpreterem/ [11].

Oprócz sondy i pakietu adaptera w skład emulatora wchodzi dodatkowo dwa standardowe pakiety pamięci dynamicznej /2x128kB/ systemu RTDS-8, stanowiące tu blok pamięci emulowanej.

Emulator ten umożliwia realizację symulacji /emulacji podsystemu wejścia-wyjścia.

Opracowywany emulator pozwala emulować mikroprocesor 8086 w dowolnej konfiguracji nie ograniczając jego własności funkcjonalnych /z wyjątkiem flagi TRAP w trybie pracy krokowej emulatora/.

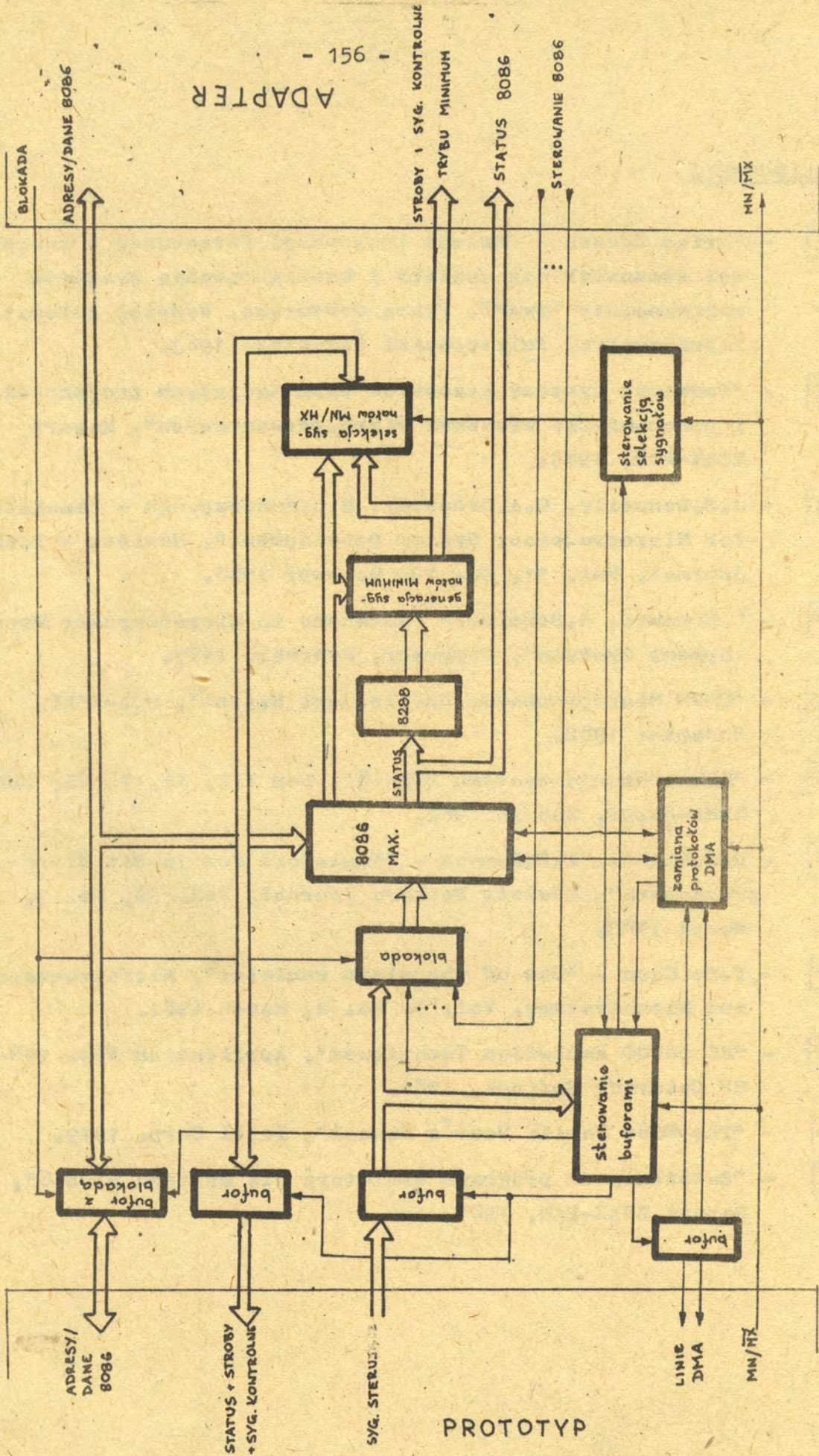
PODZIEKOWANIE

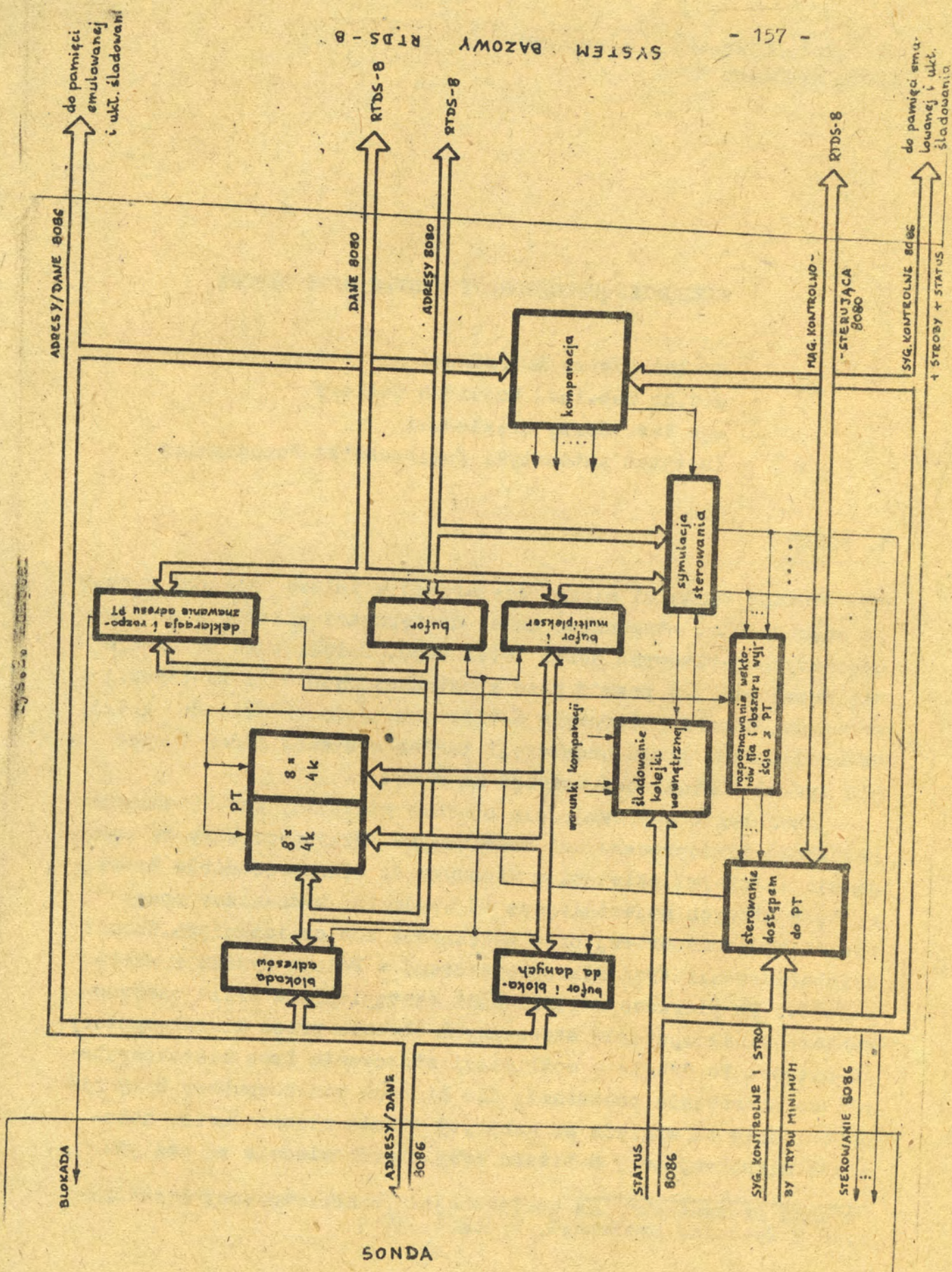
Autor pragnie podziękować kolegom z Zespołu Mikroprogramowania ZSAK-PAN w Gliwicach za cenne uwagi i pomoc w przygotowaniu niniejszego referatu.

BIBLIOGRAFIA

- [1] - Marian Konsek - "Metoda instrukcji forsowanej w koncepcji stanowisk dla analizy i konfigurowania systemów mikrokomputerowych", Praca doktorska, Wydział Automatyki i Informatyki Politechniki Śląskiej, 1983.
- [2] - "Podstawy syntezy stanowisk wspomagających projektowanie i uruchamianie systemów mikroprocesorowych", Raport ZSAK-PAN, 1983.
- [3] - J.B.Donnelly, G.A.Greenley, M.E.Muterspaugh - "Emulators for Microprocessor System Development", Hewlett - Packard Journal, Vol. 31, No. 10, October 1980.
- [4] - L.Krummel, G.Schultz - "Advances in Microcomputer Development Systems", Computer, February 1977.
- [5] - "UMDS Microprocessor Development System", MTA-KFKI, Budapest 1982.
- [6] - "Dokumentacja systemu RTDS-8", Tom III, IV, V, VI, ZUK MERA-ELZAB, Zabrze 1982.
- [7] - D.B.Richey, J.P.Romano - "Emulators for 16-Bit Microprocessors", Hewlett Packard Journal, Vol. 34, No. 3, March 1983.
- [8] - T.Mc Cann - "Use of the slave emulator", Microprocessors and Microsystems, Vol. 5, No. 2, March 1981.
- [9] - "MC 68000 Emulation Techniques", Application Note 298-A, HP Colorado Springs, 1981.
- [10] - "The 8086 Family User's Manual", Intel Corp. 1979.
- [11] - "Założenia do programu emulatora dla systemu RTDS-8", Raport ZSAK-PAN, 1984.

rys.1. Sonda





VI Szkoła Mikroprocesorowa CSI/PTI
Łódź, grudzień 1984

WIELOMIKROPROCESOROWY SYSTEM 8/16 BITOWY

dr inż. Jerzy Brzeziński
doc.dr hab.inż. Wojciech Cellary
mgr inż. Jerzy Kręglewski
Instytut Automatyki Politechniki Poznańskiej

1. WSTĘP

Rozwój techniki mikroprocesorowej w Polsce osiągnął w chwili obecnej stan stosunkowo dużej dostępności układów mikroprocesorowych 8-bitowych serii Intel 8080. Dzięki temu obserwuje się coraz większy rozwój prac konstrukcyjnych oraz zastosowań systemów mikrokomputerowych 8-bitowych. Jednocześnie do chwili obecnej pomimo wielu propozycji takich systemów żaden z nich nie zyskał dominującej pozycji na rynku.

Równolegle obserwuje się obecnie gwałtowny wzrost zainteresowania mikroprocesorami 16-bitowymi. Mikroprocesory te cechują się istotnymi zaletami w stosunku do mikroprocesorów 8-bitowych, z których najważniejsze to wbudowane mechanizmy pracy wielomikroprocesorowej oraz zwiększona moc obliczeniowa. Wzrost zainteresowania tymi mikroprocesorami w Polsce wynika z wielu powodów. Po pierwsze z możliwości zastąpienia w wielu zastosowaniach minikomputerów stosowanych obecnie przez mikrokomputery 16-bitowe. Po drugie z możliwości stosowania tych mikrokomputerów do sterowania procesami, dla których minikomputery były nieodpowiednie ze względu na gabaryty i małą niezawodność, natomiast mikrokomputery 8-bitowe były nieodpowiednie ze względu

Referat prezentowany na konferencji: "Mikrokomputery w Automatyce i Technice Systemów", Wrocław, 1984.

na zbyt małą efektywność działania. Po trzecie ze względu na stale malejącą cenę układów mikroprocesorowych 16-bitowych w krajach zachodnich oraz spodziewane podjęcie produkcji układów serii Intel 8086 w krajach RWPG.

W związku z takim stanem oraz tendencjami rozwojowymi systemów mikrokomputerowych w Polsce zachodzi obawa, że w najbliższych latach odbywać się będzie nieskoordynowany rozwój niezależnie systemów 8-mio i 16-to bitowych. Taki brak koordynacji jest oczywiście zjawiskiem niekorzystnym, stąd pilny postulat opracowania i wdrożenia do produkcji rodziny systemów mikroprocesorowych 8/16 bitowych o wspólnej architekturze. Oczywiście architektura taka nie może ograniczać możliwości mikroprocesorów 16-bitowych. System taki musi charakteryzować się zatem następującymi cechami: /1/ musi być systemem wielomikroprocesorowym, /2/ musi mieć analogiczne moduły z procesorami 8-mio i 16-to bitowymi, /3/ musi mieć możliwość niezależnej konfiguracji w wersji 8-bitowej, 16-bitowej oraz mieszanej 8/16-bitowej. Opracowanie takiego systemu pozwoli na uzyskanie następujących ogólnych celów. Po pierwsze rozwoju badań nad systemami wielomikroprocesorowymi i ich zastosowaniami już na bazie techniki mikroprocesorowej obecnie rozpowszechnionej w kraju /układy mikroprocesorowe, systemy uruchomieniowe itp/. Po drugie łagodne przejście od konstrukcji i zastosowań systemów mikroprocesorowych 8-bitowych do 16-bitowych oraz po trzecie, możliwość wykorzystania istniejącego oprogramowania dla mikrokomputerów 8-bitowych w systemach mieszanych 8/16 bitowych.

Celem tego artykułu jest przedstawienie architektury proponowanego powyżej systemu oraz architektury modułów mikroprocesorów wchodzących w skład tego systemu.

2. ARCHITEKTURA SYSTEMU

W celu skonstruowania systemu spełniającego wymagania przedstawione w rozdziale 1 przyjęto, że podstawowym modułem tego systemu będzie jednopłytowy, jednoprocessorowy mikrokomputer /por. [1, 7]. Analogiczne moduły zostały opracowane dla mikroprocesorów 8-mio i 16-to bitowych.

Architekturę wewnętrzną tych modułów przedstawiamy w rozdziale 3. W tym miejscu wspomnijmy jedynie, że każdy taki moduł

może być samodzielnie stosowany, gdyż posiada własną pamięć operacyjną, podstawowe układy wejścia - wyjścia oraz system przerwań. W zastosowaniach wymagających mikrokomputera o większych możliwościach zestawia się system złożony z modułu mikrokomputera jednopłytkowego oraz odpowiednich modułów biernych takich jak : modułów pamięci operacyjnej, sterowników pamięci zewnętrznych, modułów wejścia - wyjścia specjalizowanych do współpracy z określonymi urządzeniami lub procesami przemysłowymi. Ważną cechą architektury systemu jest możliwość konstrukcji uniwersalnych modułów biernych, tzn. przystosowanych do współpracy zarówno z jednopłytkowymi mikrokomputerami 8-bitowymi jak i 16-bitowymi. Uzyskano to dzięki przyjęciu odpowiedniego standardu magistrali systemowej przewidującego przesyłanie danych w postaci bajtów lub słów 16-bitowych.

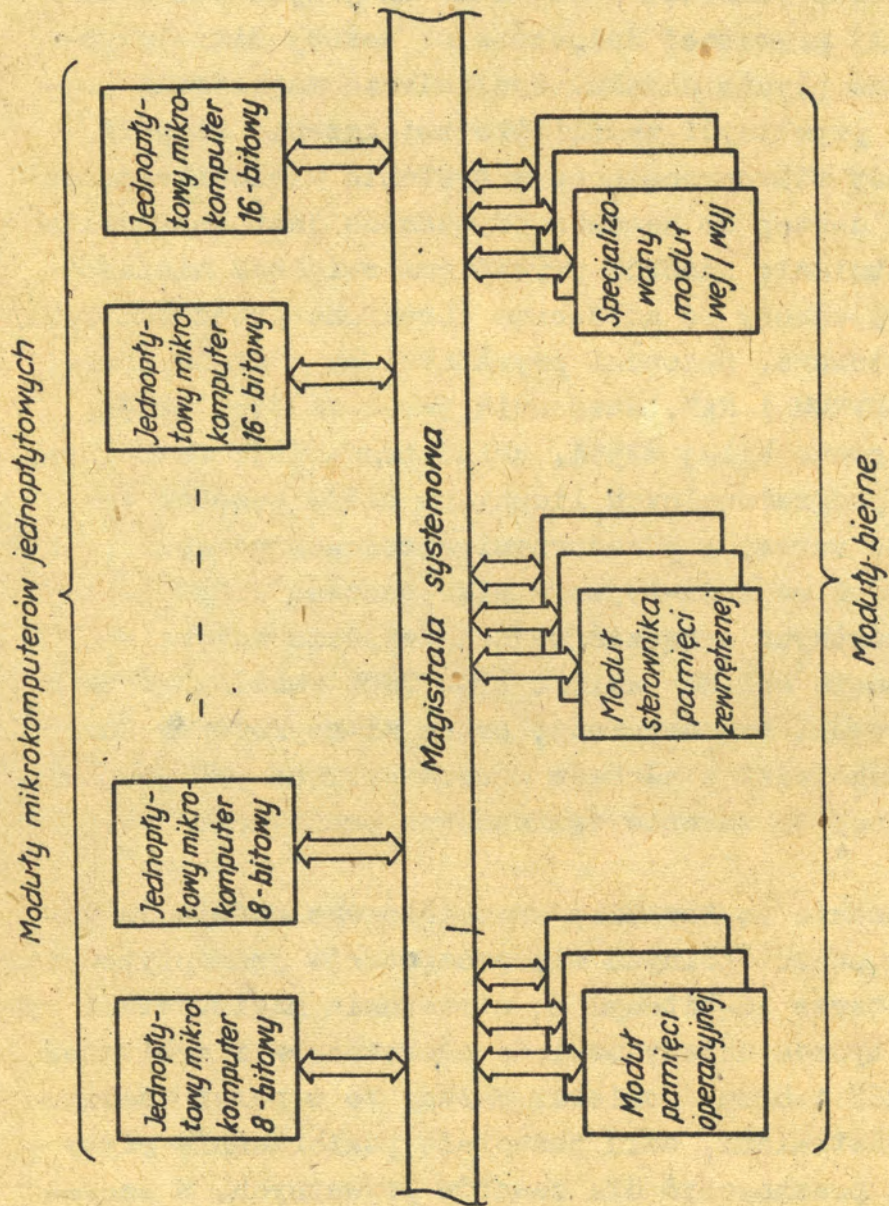
Następną bardziej skomplikowaną strukturą systemu jest jednorodna struktura wielomikroprocesorowa. W skład systemu o takiej strukturze wchodzi wiele mikrokomputerów jednopłytkowych z mikroprocesorem danego typu oraz dowolne moduły bierne. Warto podkreślić, że zasada organizacji pracy systemu wielomikroprocesorowego jest niezależna od użytego typu mikroprocesora. Jednocześnie w systemie wielomikroprocesorowym 16-bitowym wykorzystane są wszystkie potencjalne możliwości mikroprocesora Intel 8086.

Najbardziej ogólną strukturą systemu jest struktura mieszana przedstawiona na rys.1, w skład której wchodzi mikrokomputery 8-bitowe i 16-bitowe oraz moduły bierne dołączone do wspólnej magistrali systemowej.

3. ARCHITEKTURA MODUŁÓW MIKROKOMPUTERÓW JEDNOPLYTKOWYCH

Jak wspomnieliśmy w rozdziale 2 podstawowym modułem systemu jest moduł mikrokomputera jednopłytkowego mogącego pracować samodzielnie, w połączeniu z modułami biernymi oraz z innymi modułami mikrokomputerów jednopłytkowych.

W skład systemu wchodzi dwa moduły : pierwszy z mikroprocesorem 8-bitowym Intel 8080 oraz drugi z mikroprocesorem 16-bitowym Intel 8086. Ogólna architektura modułu mikrokomputera jednopłytkowego niezależnie od stosowanego mikroprocesora przedstawio-



Rys.1. Architektura systemu wielomikroprocesorowego 8 / 16 bitowego.

na jest na rysunku 2.

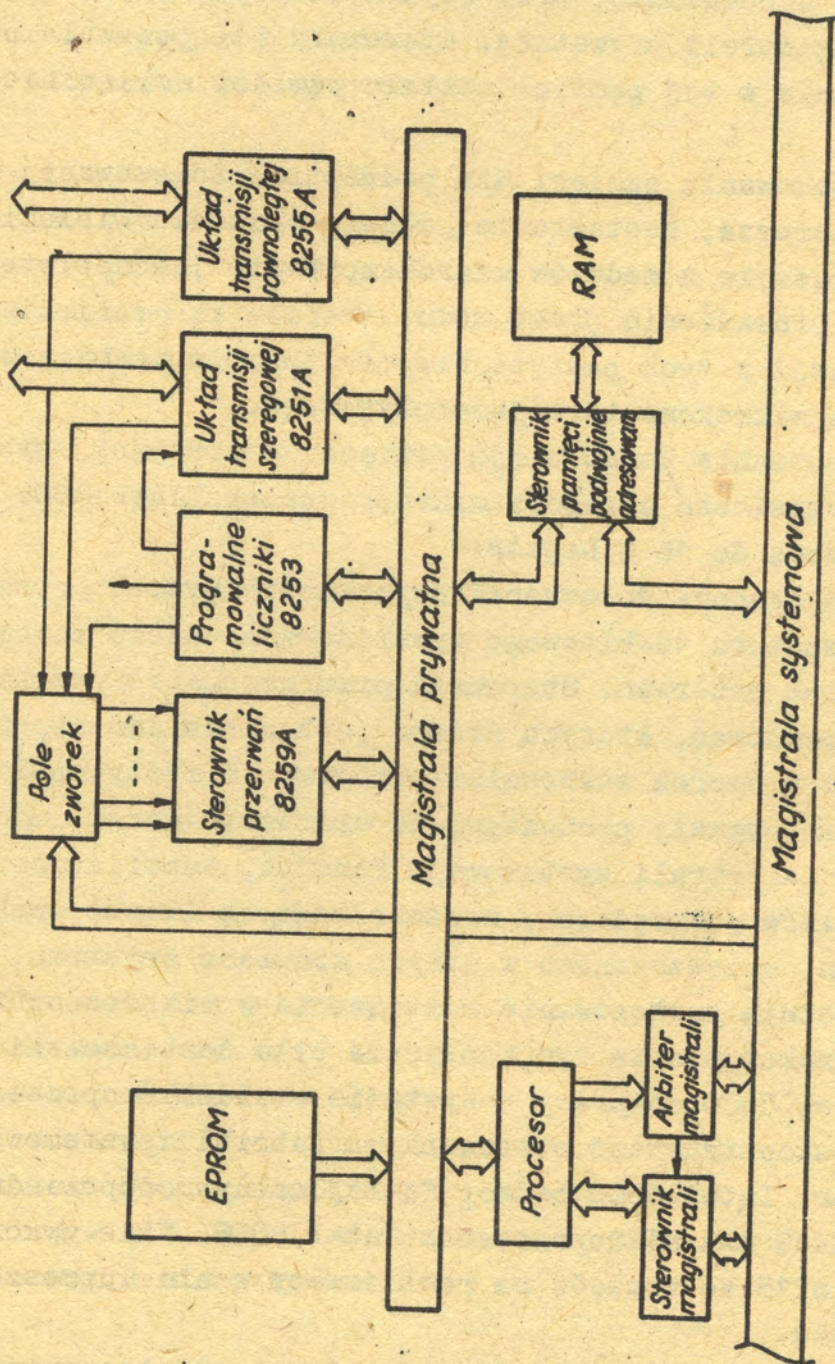
W mikrokomputerze wyróżniono dwie magistrale : magistralę systemową oraz magistralę prywatną.

Magistrala systemowa służy do realizacji połączenia z innymi modułami systemu, zarówno z modułami aktywnymi jak i biernymi. Do magistrali prywatnej dołączone są zasoby mikrokomputera umieszczone na płycie modułu. Wydzielenie magistrali prywatnej z zasobami prywatnymi umożliwiło zwiększenie stopnia równoległości pracy mikroprocesorów w systemie wielomikroprocesorowym, ponieważ dostęp do zasobów prywatnych jest bezkonfliktowy. Ponadto, istnienie zasobów prywatnych zwiększa niezawodność systemu, gdyż zasoby te są dobrze chronione przed skutkami awarii innego procesora. Zasobami prywatnymi są : część pamięci operacyjnej typu EPROM i RAM, sterownik przerwań typu 8259A, układ transmisji równoległej 8255A, układ transmisji szeregowej 8251A oraz układ programowalnych liczników 8253. Zasobom tym przydzielono pewne adresy z przestrzeni adresowej pamięci operacyjnej oraz urządzeń wejścia-wyjścia mikroprocesora. Ostateczny wybór adresów prywatnych urządzeń wejścia-wyjścia odbywa się przez zaprogramowanie układu pamięci typu PROM realizującego funkcje dekodera adresów. Wygenerowanie przez mikroprocesor adresu różnego od któregośkolwiek z adresów zasobów prywatnych powoduje automatycznie dostęp do zasobów systemowych przez magistralę systemową.

Przejdźmy obecnie do bardziej szczegółowego omówienia indywidualnych cech poszczególnych mikrokomputerów jednopłytkowych.

W mikrokomputerze 16-bitowym do sterowania magistralami wykorzystano monolityczne układy scalone sterowników i arbiterów magistral typu 8288 i 8289. Ponieważ układy te w pełni realizują sterowanie magistralami, całą pozostałą powierzchnię płyty modułu można było przeznaczyć dla zasobów prywatnych. W szczególności w skład modułu wchodzi 8 lub 16 k bajtów pamięci typu EPROM w zależności od zastosowania układów pamięci typu 2716 lub 2732 oraz 32 lub 128 k bajtów pamięci typu RAM w zależności od zastosowania układów pamięci typu 2117 lub 2164.

Pamięć typu RAM umieszczona na płycie modułu mikrokomputera jednopłytkowego została zorganizowana w taki sposób aby była jednocześnie pamięcią prywatną z tego modułu i pamięcią syste-



Rys. 2. Architektura modułu mikrokomputera jednoportowego.

mową wszystkich pozostałych mikrokomputerów jednopłytkowych systemu. Sterownik tej pamięci umożliwia jej podwójne adresowanie: od strony magistrali prywatnej oraz od strony magistrali systemowej. Sterownik ten pełni również funkcje arbitera dostępu do pamięci podwójnie adresowanej, przy czym wyższy priorytet ma dostęp od strony magistrali prywatnej. Sterownik ten pozwala ponadto na wydzielenie w tej pamięci obszaru pamięci wyłącznie prywatnej.

Dzięki zastosowaniu pamięci RAM podwójnie adresowanej możliwe jest : po pierwsze, zestawienie pełnego systemu wielomikroprocesorowego wyłącznie z modułów mikrokomputerów jednopłytkowych oraz, po drugie, prowadzenie przez moduł sterownika pamięci zewnętrznych transmisji z tych pamięci bezpośrednio do pamięci umieszczonej w module mikrokomputera jednopłytkowego.

Na zakończenie omawiania konstrukcji pamięci operacyjnej systemu wspomnijmy, że przestrzeń adresowa mikroprocesora Intel 8086 została tu rozszerzoną do 16 M bajtów.

Przejdziemy obecnie do scharakteryzowania systemu przerwań. W module mikrokomputera 16-bitowego zrealizowano wielopoziomowy, priorytetowy system przerwań. Sterownik przerwań może przyjmować 8 wektorowanych przerwań, których źródło jest określane poprzez układ zwerek w polu zwerek sterownika przerwań. Przyczynami przerwań mogą być zatem sygnały pochodzące od urządzeń danego modułu lub pochodzące z magistrali systemowej. Ponadto, umożliwiono dołączanie sterowników podrzędnych, rozszerzających liczbę wektorowanych przerwań, umieszczonych w innych modułach systemu.

Omówimy obecnie zastosowanie rozwiązania w mikrokomputerze 8-bitowym. W mikrokomputerze tym konieczne było dostosowanie mikrokomputera Intel 8080 do pracy w systemie wielomikroprocesorowym. W tym celu skonstruowano sterownik magistrali systemowej dla mikroprocesora Intel 8080 będący funkcjonalnym odpowiednikiem układów 8288 i 8289 dla mikroprocesora Intel 8086. Nie wykorzystano tu układu 8218 ze względu na realizowany w nim uproszczony algorytm arbitrażu.

Ze względu na konieczność umieszczenia na płycie wymienionego powyżej sterownika, konieczne było uproszczenie sterownika pamięci RAM podwójnie adresowanej. Uproszczenie to polega na tym, że dostęp do tej pamięci od strony magistrali systemowej blokuje mikroprocesor.

Ponadto ze względu na różną koncepcję wektorowania przerwań dla mikroprocesorów Intel 8080 i Intel 8086 musiano zrezygnować z możliwości rozszerzania systemu przerwań przez dołączenie sterowników podrzędnych z innych modułów systemu.

Mikrokomputer jednopłytkowy zawiera 4 lub 8 k bajty pamięci EPROM oraz 32 k bajty pamięci RAM /układy pamięci typu 2117/, a przestrzeń adresowa mikroprocesora Intel 8080 została rozszerzona do 1 M bajtu.

BIBLIOGRAFIA

1. Brzeziński J., Cellary W., Kręglewski J.,
Multi-microprocessor system for real-time process control based on Intel 8086 microprocessor family, Proc. of IV Polish-English Seminar On Real-Time Process Control, Jabłonna 1983.

VI Szkoła Mikroprocesorowa CSI/PTI
Łódź, grudzień 1984

PAKIET JEDNOSTKI CENTRALNEJ MM86
SYSTEMU MIR-PROWAY

dr inż. Andrzej Syrczyński
Przemysłowy Instytut Automatyki
i Pomiarów
Al. Jerozolimskie 202
02-222 Warszawa, tel. 23.70.81 w.116

1. SYSTEM

Zdecentralizowany mikroprocesorowy system automatyki kompleksowej MIR-PROWAY jest opracowywany do automatyzacji obiektów przestrzennie rozłożonych w wielu dziedzinach gospodarki. Obiektowe realizacje systemu będą się składały z dużej ilości stacji /do stu/, połączonych siecią lokalną.

Sieć lokalna będzie zrealizowana według standardu IEC PROWAY, którego projekty opracowuje Podkomitet 65C Międzynarodowej Komisji Elektrotechnicznej.

Stacjami będą sterowniki mikroprocesorowe maszyn i urządzeń, koncentratory danych, stanowiska operatorskie i komputery sterowania nadrzędnego. Każda stacja dołączona do sieci lokalnej musi zawierać standardowe urządzenia komunikacyjne /kontroler komunikacyjny i urządzenia liniowe/ oraz wewnętrzną moc obliczeniową warstwy użytkownika. Tą mocą obliczeniową będą, w ogromnej większości stacji, mikroprocesory w pakietach jednostek centralnych.

Ze względu na ograniczenia bazy elementowej w pierwszej kolejności opracowano jednopłytkowy mikrokomputer MM 80 z mikro-

procesorem typu MCY 7880, znajdujący się teraz w badaniach pełnych prototypów i równolegle w fazie pilotowych aplikacji.

Moc obliczeniowa tego pakietu jest niewystarczająca dla szeregu zastosowań, szczególnie do układów sterowania robotów, w których opracowanie jest zaangażowany Instytut. Dlatego podjęto opracowanie i oprogramowanie mikrokomputera 16-bitowego MM 86.

W systemie przewiduje się szerokie stosowanie pracy wieloprocessorowej wewnątrz stacji. Stacja może składać się z jednej lub kilku kaset, przy czym w każdej z kaset z reguły będzie pracować więcej niż jeden pakiet aktywny. Z tego względu wszystkie pakiety aktywne, w tym oczywiście pakiety jednostek centralnych, muszą być dostosowane do pracy w systemie wieloprocessorowym. Poza pakietami jednostek centralnych pakietami aktywnymi na magistrali kasety będą: mikroprocesorowy kontroler komunikacyjny MK 40, komunikujący się z zasobami danych stacji umieszczonymi we wspólnej pamięci danych, pakiet M170 sprzężenia kaset, pakiet M105 adaptera Wspólnej Szyny, sprzęgający stację z minikomputerem serii SM oraz pakiety sprzężeń z pamięciami masowymi dyskowymi.

Omawiany pakiet jednostki centralnej jest przeznaczony do zastosowań w automatyce przemysłowej i robotyce, co rzutowało na jego rozwiązania. Nie przewiduje się natomiast jego stosowania w małej informatyce, urządzeniach i sieciach obsługi oraz w komputerach osobistych.

2. ZAŁOŻENIA KONSTRUKCJI

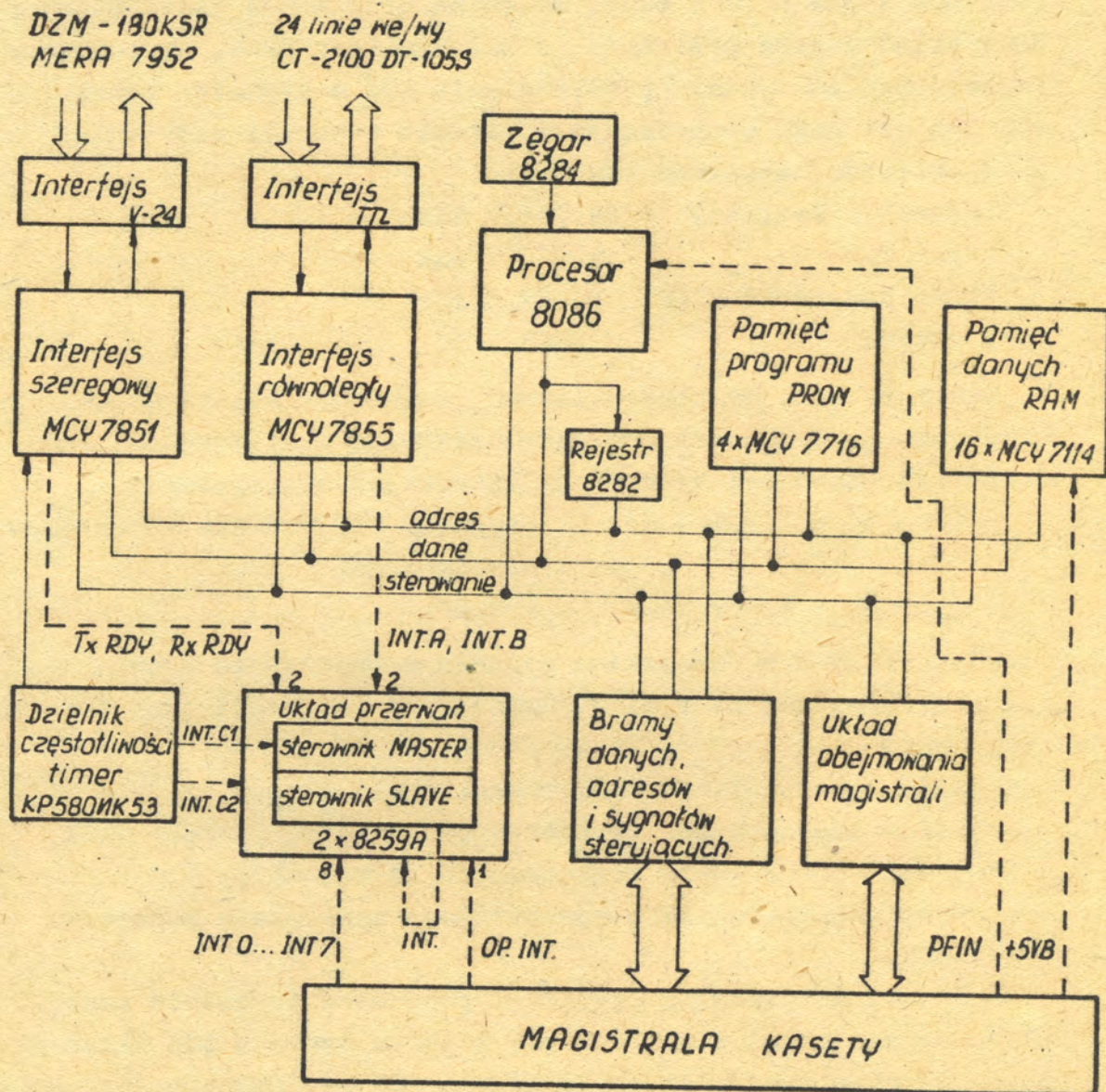
Presja czasu - konieczność przekazania modeli użytkowych do długotrwałych prac nad oprogramowaniem, m.in. robotów, przeprowadzenia badań funkcjonalnych i weryfikacji do prototypu, przy utrzymywaniu się wielkich trudności zaopatrzeniowych i technologicznych, spowodowała przyjęcie szeregu założeń ograniczających konfigurację i rozwiązania pakietu. Uznano, że obecne opracowanie będzie stanowiło I generację jednostki 16-bitowej. Najważniejsze założenia można przedstawić następująco:

1. pakiet MM 86 musi być jednopłytkowym mikrokomputerem o pełnej konfiguracji podstawowej;
2. musi spełniać wymagania funkcjonalne do zastosowań w automatyce, w zakresie obsługi pakietów pasywnych /wejść-wyjść/, współpracy z mikroprocesorowym kontrolerem komunikacyjnym i siecią lokalną PROWAY, zabezpieczenia informacji przy zanikach zasilania i kompatybilności elektromagnetycznej;
3. konstrukcja mechaniczna jest wyznaczona standardem systemu. Wymiary płyty drukowanej - podwójna Eurocarta 220 x 233 mm, złącza magistrali - dwa pośrednie 96-stykowe, złącza do urządzeń peryferyjnych - szufladowe gięte 25-stykowe;
4. magistrala kasety według przyjętej w kraju normy branżowej BN-84/3105-02, opartej o magistralę AMS-BUS;
5. baza elementowa wyłącznie krajowa i KS, przy czym łączna ilość i wartość elementów importowanych powinna być minimalna. Tylko do wykonania serii modeli użytkowych dopuszczono import KK;
6. ze względu na możliwości technologiczne producentów ograniczono się do płyty dwustronnie drukowanej;
7. brak w kraju systemów uruchomieniowych do mikroprocesorów 16-bitowych oraz długi czas konieczny do oprogramowania systemów dyskowych zmusiły do przyjęcia pracy z taśmą papierową, a więc wprowadzenia na pakiet interfejsu równoległego.

3. KONFIGURACJA

Z wymienionych założeń wyniknęła konfiguracja przedstawiona na rys. 1. Jest to oczywiście konfiguracja minimalna, tak w sensie wyboru modu minimalnego dla mikroprocesora 8086, jak i ograniczenia pojemności pamięci i rezygnacji z niektórych mniej koniecznych funkcji i układów. Główne decyzje konstrukcyjne to: wybór modu minimalnego procesora, przyjęcie szeregowej arbitracji magistrali, statycznej pamięci danych i rezygnacja z wykorzystania magistrali jednodostępnej /resident bus/. Przyjęta konfiguracja zapewnia jednak wykonanie wszystkich podstawowych funkcji.

Główną trudnością była realizacja nawet tej konfiguracji na jednej płycie przy braku wielu znanych układów scalonych wyższej skali integracji.



Rys. 1. Konfiguracja

4. PROCESOR

Procesor zrealizowano w module minimalnym, głównie w celu wyeliminowania trudno dostępnego układu kontrolera magistrali

wewnętrznej typu 8288. Sygnały rozkazowe są wydawane bezpośrednio przez mikroprocesor, a po przekodowaniu podawane na magistralę kasety. W rejestrze demultiplikacji adresu i jako nadajniki/odbiorniki sygnałów magistrali kasety wykorzystano układy 8-bitowe typów 8282 i 8287, które ma produkować CEMI. Standardowy układ zegara procesora, z układem typu 8284, uzupełniono iloczynowaniem sygnału potwierdzenia ACK z sygnałem zabezpieczenia pamięci MPRO pochodzącym z pakietu kontroli zasilania.

Częstotliwości zegarowe wynoszą:

rezonator kwarcowy 14.745,6 kHz

zegar CLK 4.915,2 kHz

zegar peryferii PCLK 2.457,6 kHz

Cykl zegara 203 ns

Wyjaśnić trzeba, że częstotliwość rezonatora bliską górnej granicy równej 15 MHz dobrano jako całkowitą wielokrotność standardowych szybkości transmisji interfejsu szeregowego:

$9.600 \cdot 16 = 153.600 \text{ Hz}$ /sygnały RxC, TxC dla interfejsu szeregowego/

$153,6 \cdot 16 = 2.457,6 \text{ kHz}$ /PCLK/

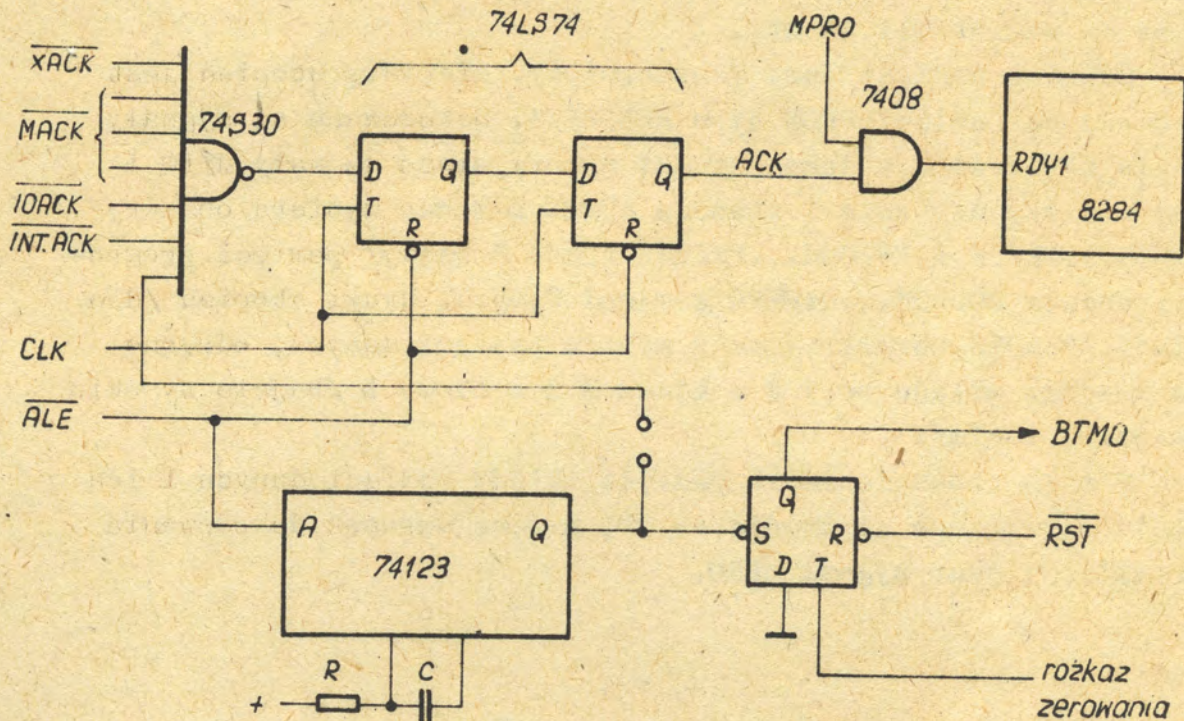
Obwody związane z tworzeniem sygnału wewnętrznego potwierdzenia ACK przedstawiono na rys. 2. Sygnał potwierdzenia jest tworzony jako suma:

- sygnału potwierdzenia XACK z magistrali kasety;
- sygnałów zdekodowania wewnętrznych obszarów pamięci MACK,
- sygnałów zdekodowania wewnętrznych adresów I/O,
- sygnału potwierdzenia z wewnętrznego kontrolera przerwań INT.ACK,

W celu dopasowania szybkości procesora do czasów dostępu adresowanych układów wprowadzono w torze sygnału ACK układ stałego opóźnienia o jeden okres zegara CLK za pomocą przerzutnika 74LS74. Po otrzymaniu hipotetycznych pamięci CEMI MCY 7114 i MCY 7716 ten fragment będzie weryfikowany.

Wprowadzono również kontrolę potwierdzenia adresu. W czasie pracy uniwibrator 74123 jest stale pobudzany impulsami ALE. Brak potwierdzenia adresu, jeżeli trwa dłużej niż 1 ms, powoduje zakończenie wydawania sygnału z uniwibratora, co ustawia przerzutnik sygnału time-outu BTMO. Zależnie od ustawienia mikroprzełączników na pakiecie nastąpi:

- a/ zatrzymanie procesora z sygnalizacją świetlną na pulpicie,
- b/ zgłoszenie przerwania,
- c/ kontynuacja biegu programu z możliwością programowego odczytu zapamiętanego sygnału BTMO. Odczyt zeruje przerzutnik BTMO.



Rys. 2. Obwody potwierdzenia

5. PAMIĘCI

Wspomniane poprzednio ograniczenia wpłynęły na pojemności pamięci. Na pakiecie znajduje się 8 K bajtów pamięci programu /4 układy MCY 7716/ i 8 K bajtów pamięci danych /16 układów MCY 7114/. Pojemności te pozwalają umieścić system operacyjny oraz wybrane najczęściej używane procedury i podprogramy w pamięci programu, zaś w pamięci danych wektory przerwania, stos, bieżące dane systemu operacyjnego i wybrane bloki danych użytko-

wnika. Przyjęto, że w większości aplikacji będą stosowane pakiety rozszerzenia pamięci:

ML 50, o pojemności 32 K bajtów programu i 8 K bajtów danych,

ML 32, o pojemności 32 K bajtów danych

w ilości potrzebnej dla danej aplikacji.

We wszystkich pamięciach /wewnętrznej i w pakietach rozszerzenia/ można adresować zarówno słowa, jak i każdy z bajtów, stosując jeden z trzech typów przekazu przewidzianych do stosowania na magistrali kasety.

Dekoder pamięci jest dwustopniowy. Pierwszy stopień jest wykonany na pamięci PROM typu KP556PT4. Dekodowane są sygnały adresu A19...A12, a jako warunki wykorzystano sygnały M/IO i sumę rozkazu czytania i pisania RDWR. Dekoder wybiera obszary FE000H...FEFFF i FF000H...FFFFFH /po 4 K bajty/ pamięci programu i obszar 00000H...01FFFH pamięci danych. Drugi stopień /dwa układy 74S405/ tworzy sygnały wyboru pamięci danych, odrębne dla każdego układu po 1 K w bloku H i w bloku L /bajcie wysokim i bajcie niskim/.

W celu zabezpieczenia pamięci układy pamięci danych i ich dekodery^{sa} zasilane ze źródła +5 VB, a jako warunek dekodowania wprowadzony jest sygnał MPRO.

6. UKŁADY PRZERWAŃ

Jest to najbardziej rozbudowany fragment omawianego mikrokomputera. Przyjęta magistrala kasety stosuje 64-poziomowy układ przerwań, z dwoma warstwami kontrolerów przerwań typu 8259A. Kontroler wyższy - MASTER znajduje się przy procesorze, a więc na pakiecie MM86, zaś kontrolery niższe - SLAVE mogą być zarówno na pakiecie MM86, jak i na innych pakietach, obsługiwanych przez pakiet jednostki centralnej.

Wobec istnienia ośmiu linii przerwań INT 0...INT 7 na magistrali kasety i kilku przerwań wewnętrznych na pakiecie MM86, zaistniała konieczność umieszczenia w jednostce centralnej dwóch kontrolerów przerwań - jednego MASTER i jednego SLAVE.

Na stałe do najwyższego wejścia IR 0 kontrolera MASTER jest

dołączone wejście podstawowego przerwania zegarowego INT C1; stosowanego przez system operacyjny /rys. 1/.

Pozostałe linie sygnałów przyczyn przerwania wchodzi na wspólny dla obu kontrolerów kros, z którego mogą być, zależnie od konfiguracji zestawu, dołączane do jednego z kontrolerów.

Są to:

- 8 linii INT 0 ... INT 7 magistrali kasety. Każda z tych linii może przenosić indywidualne przerwanie, albo może pochodzić z zewnętrznego kontrolera i wtedy reprezentuje 8 poziomów przerwania. W tym drugim przypadku linia taka musi być dołączona do jednego z wejść kontrolera MASTER,
- linia INT wewnętrznego kontrolera SLAVE,
- przerwania interfejsu szeregowego TxRDY, RxRDY,
- przerwania z bram interfejsu równoległego INT A, INT B,
- drugie przerwanie zegarowe INT C2,
- sprzętowe przerwanie operatora OP.INT.

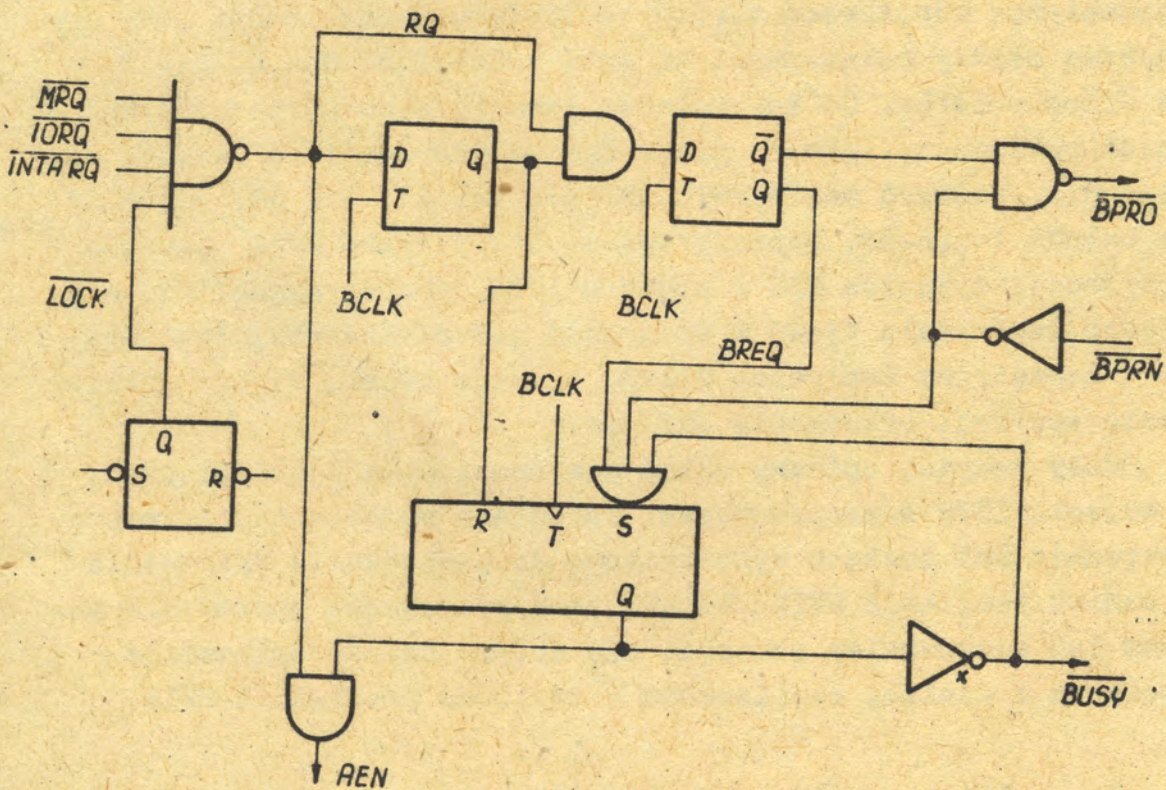
Autonomiczna współpraca między kontrolerami obu stopni jest dokonywana między połączonymi ze sobą końcówkami CAS 2, CAS 1, CAS 0 kontrolerów. Do kontrolerów zewnętrznych umieszczonych na innych pakietach, sygnały grupy CAS są doprowadzone liniami rozszerzenia przerwania magistrali INT A2, INT A1, INT A0. Odpowiednie układy logiczne, zaprojektowane dla pakietu MM86, sterują dystrybucją sygnałów CAS i sygnału INTA, tworzą sygnał żądania objęcia magistrali tylko w przypadku przerwania obsługiwanego przez zewnętrzny kontroler SLAVE, a także tworzą sygnał potwierdzenia operacji przerwania INT.ACK.

Cały powyżej opisany układ wielopoziomowy jest dołączony do wejścia INTR mikroprocesora. Natomiast wejście niemaskowane przerwania NMI zostało wykorzystane do podłączenia przerwania od zaniku zasilania PFIN. Dzięki temu programowe operowanie zakazem lub maskowaniem przerwania nie wpływa na zabezpieczenie procesora i pamięci realizowane w obsłudze przerwania PFIN.

7. UKŁAD OBEJMOWANIA MAGISTRALI

Niedostępność w kraju i w KS układu scalonego typu 8289 - arbitera magistrali spowodowała zaprojektowanie sekwencyjnego układu logicznego wykonującego zadania arbitracji i obejmowania magistrali. Układ spełnia wszystkie wymagania funkcjonalne przyjętej magistrali kasety, dla szeregowego wariantu określania priorytetu.

Na wejściu układu /rys. 3/ tworzona jest suma logiczna przyczyn żądania objęcia magistrali: sygnałów żądania dostępu do zewnętrznej pamięci MRQ, zewnętrznych bram wejść/wyjść IORQ, zewnętrznego kontrolera przerwań INTA RQ i sygnału LOCK wewnętrzznego programowo ustawianego przerzutnika stałego dostępu do magistrali.



Rys. 3 Uproszczony schemat układu obejmowania

Utworzony sygnał żądania RQ podlega synchronizacji zegarem

magistrali BCLK w dwóch stopniach przerzutników i wchodzi jako sygnał BREQ do obwodów arbitracji.

Z chwilą pojawienia się sygnału BREQ układ arbitracji przekazuje do pakietów o niższym priorytecie zakaz obejmowania magistrali i czeka na wolną magistralę /sygnał BUSY = 0/ . /sygnał BPRN = 1/. Gdy zostanie spełniony warunek następuje ponowna synchronizacja zegarem magistrali BCLK i zajęcie magistrali /wydanie BUSY = 1/.

Równocześnie z sygnałem BUSY jest tworzony sygnał AEN zezwolenia przekazu adresów i danych. Z wymaganym przez warunki magistrali opóźnieniem jest tworzony sygnał BCOM zezwolenia wydania sygnałów rozkazowych na magistralę.

Wielokrotna synchronizacja tym samym zboczem sygnału zegarowego w toku sekwencji arbitracji ma na celu wykluczenie kolizji na magistrali. Samo uzależnienie arbitracji od sygnałów łańcucha BPRN, BPRO nie gwarantuje zapobieżenia kolizji, gdy dodawałyby się czasy propagacji po magistrali i opóźnień w bramkach prowadząc do wyścigów.

8. INTERFEJS SZEREGOWY

Interfejs szeregowy umożliwia dołączenie dowolnego urządzenia peryferyjnego stosującego standard V-24, jak terminal DZM 180 KSR czy monitor ekranowy, bądź urządzenia transmisyjnego - modemu lub konwertera.

Interfejs obsługuje sygnały odbierane RxD, CTS, DSR oraz sygnały nadawane TxD, DTR, RTS. Interfejs wykorzystuje krajowy układ scalony MCY 7851 oraz układy symetrycznych nadajników typu UCY 75110. Do odbioru sygnałów zastosowano wzmacniacze tranzystorowe z dzielnikiem rezystorowym na wejściu.

Sygnał zegara nadawczego i odbiorczego jest tworzony w układzie programowanego licznika K 580 IK 53, w jego kanale nr 0. Pozostałe dwa kanały tworzą przerwania zegarowe. Zarówno układ licznikowy, jak i układ MCY 7851 otrzymują na wejścia CLK ciąg zegarowy PCLK/2 o częstotliwości 1227,4 kHz tworzonej

przez dodatkowy dzielnik - dwójkę liczącą typu 7474.

9. INTERFEJS RÓWNOLEGŁY

Interfejs wykorzystuje układ scalony MCY 7855. Bramy A i B mogą pracować jako 8-bitowe bramy wejściowe lub wyjściowe i są wyposażone w podstawki 14-stykowe, do układów scalonych nadajników lub odbiorników sygnału. Można instalować wszystkie typy układów TTL o układzie wyprowadzeń jak 7400, a więc uzyskać wyjścia TTL lub OC o obciążeniu do 48 mA, bądź wejścia TTL. Brama C może być wykorzystana do obsługi przerwaniowej, z zawieszaniem, bram A i B. Jest wyposażona w kros i bramki typu 7404, przy czym kros pozwala dołączać bramki jako wejściowe lub wyjściowe.

W typowym wykorzystaniu do sprzężenia z czytnikiem i dziurkarką taśmy, brama C obsługuje wszystkie sygnały sterujące i kontrolne tych urządzeń. W innych zastosowaniach, np. do sprzężenia z pulpitemi operatora, sygnały bramy C mogą służyć do multiplikacji wejść lub wyjść.

Na zakończenie pozostaje wyrazić przekonanie, że w stosunkowo krótkim czasie będzie można opracować i przekazać do wdrożenia następny pakiet jednostki centralnej 16-bitowej o znacznie rozszerzonej konfiguracji i mocy obliczeniowej.

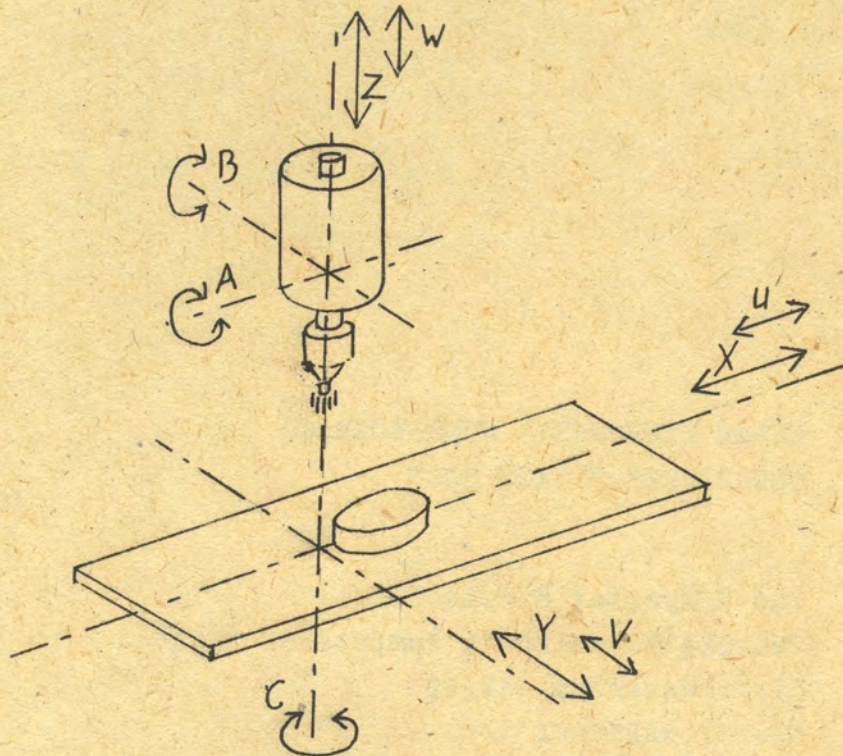
VI Szkoła Mikroprocesorowa CSI/PTI
Łódź, grudzień 1984

UKŁAD STEROWANIA NUMERYCZNEGO
OBRABIAREK "NUXON 500"

inż Krzysztof M. Święcicki
Fabryka Mierników i Komputerów "ERA"
ul. Łopuszańska 117/123
02-232 WARSZAWA
tel 237044 w 511

1. WSTĘP

Produkowany seryjnie od 1977 roku na licencji szwedzkiej układ sterowania numerycznego MERA CNC/NUCON 400 oparty na mikroprocesorze 8080 i technice TTL, nie jest w stanie sprostać rosnącym wysokim wymaganiom przemysłu obrabiarkowego, a ponadto fakt że wymaga on znacznego wkładu dewizowego spowodowało podjęcie intensywnych prac konstrukcyjnych i wdrożeniowych nad nową generacją Układu Sterowania Numerycznego obrabiarek /USN/. Z założenia ma to być system nowoczesny, całkowicie zrealizowany w technice mikroprocesorowej i wysoce elastyczny aby można go było stosować od prostej, dwuosiowej tokarki, poprzez frezarki różnych typów po najbardziej rozbudowane centra obróbcze o maksymalnej liczbie osi sterowanych.



Rys 1. Ruchy robocze obrabiarki

Osie liniowe	X, Y, Z
Osie kątowe (obrotowe)	A, B, C
Osie liniowe drugiego rzędu (wtórne)	U, V, W

2. WYMAGANIA

Nowa generacja USN musi spełniać następujące wymagania / wymieniono najważniejsze z punktu widzenia konstrukcji komputera / :

- ilość osi sterowanych 8
- ilość osi sterowanych jednocześnie 2, 3 do 5
- interpolacja liniowa typu 2D, 3D /na 4 osie/ do 5D /na 8 osi/
- interpolacja kołowa 2D na 4 osie
- interpolacja helikoidalna 3D na 4 osie
- zdolność rozdzielcza sterowania zmian położenia
 - dla osi liniowych 0.001 mm
 - dla osi obrotowych
 - przy pomiarze inдуктосyнем obrotowym 0.001 stopnia
 - przy pomiarze rezolwerem 2 biegunowym 0.001 obrotu
- zakres programowanych przesunięć
 - dla osi liniowych $+ 10\ 000\ \text{mm}$ /do $- 10\ 000\ \text{mm}$ /
 - dla osi obrotowych $+ 10\ 000$ stopni/

- zakres prędkości posuwu roboczego
 - dla osi liniowych 1 mm/min do 10 000 mm/min
 - dla osi obrotowych 1 stopień/min do 10 000 stopni/min
 - dla ruchów synchronicznych 0.001 mm/obrót do 100 mm / obrót
- automatyczna kompensacja promienia i długości narzędzia oraz położenia stołu
- automatyczna kompensacja luzów zwrotnych oraz błędu śruby obrabiarki
- programowanie bazy wymiany narzędzia lub palet
- kontrola czasu pracy narzędzia
- wprowadzenie Programu Obróbki Technologicznej /POT/ do USN:
 - z czytnika taśmy papierowej
 - z pamięci kasetowej
 - poprzez interfejs szeregowy RS 232
 - poprzez kanał DNC
- wyjście informacyjne USN
 - na dziurkarkę taśmy perforowanej
 - do pamięci kasetowej
 - poprzez interfejs szeregowy RS 232
 - poprzez kanał DNC
- diagnostyka pracy systemu
 - układu sterowania
 - napędów obrabiarki

3. ARCHITEKTURA

Spełnienie powyższych wymagań przy jednoczesnym wymogu łatwej adaptacji układu sterowania dla różnych obrabiarek wymusza zaprojektowanie systemu wieloprocessorowego, zmodularyzowanego z daleko idącym standaryzowaniem szyn. Dodatkowym i istotnym ograniczeniem jest konieczność "zmieszczenia" się w bazie elementarnej dostępnej na krajowym rynku w przewidywanym horyzoncie czasowym przy zmini - malizowaniu importu. Wyłączne oparcie się na produkowanym w CBMI mikroprocesorze MCY 7880 nie jest możliwe ze względu na jego zbyt małą moc obliczeniową. Zdecydowano się na układ trzech mikroprocesorów 16 - bitowych typu INTEL 8086 wspartych dwoma mikroprocesorami 8 - bitowymi MCY 7880. Wszystkie moduły są jednopłytkowe w formie podwójnej eurokarty /233,4 x 220mm/ wykonane w techno -

-logii druku dwustronnego z dwoma złączami pośrednimi 3 x 32 kontaktowymi.

Za podstawowy typ interfejsu przyjęto szynę 16 - bitową, wielodostępną w/g zaleceń normy branżowej BN-84/3105 - 02. Jest to interfejs typu MULTIBUS w/g IEEE 796 w realizacji konstrukcyjnej odpowiadającej projektowi AMS - BUS firmy SIMENS. Umożliwia on współpracę między sobą mikroprocesorów 16 - to i 8 - mio bitowych. Współpraca modułów odbywa się na zasadzie wymiany potwierdzeń / hand - shake / między modułem sterującym / master / a sterowanym / slave/. Zastosowano możliwość przekazywania funkcji modułu sterującego między modułami dołączonymi do szyny za pomocą szeregowego mechanizmu priorytetowego.

W systemie istnieją trzy niezależne szyny systemowe łączące się ze sobą za pośrednictwem modułów przejścia - dwudostępnej pamięci RAM o pojemności 4 KB. Ponadto istnieje szyna lokalna łącząca PROCESOR STEROWANIA USI z modułami sterowania osi napędowych i modułami pomiaru położenia tych osi.

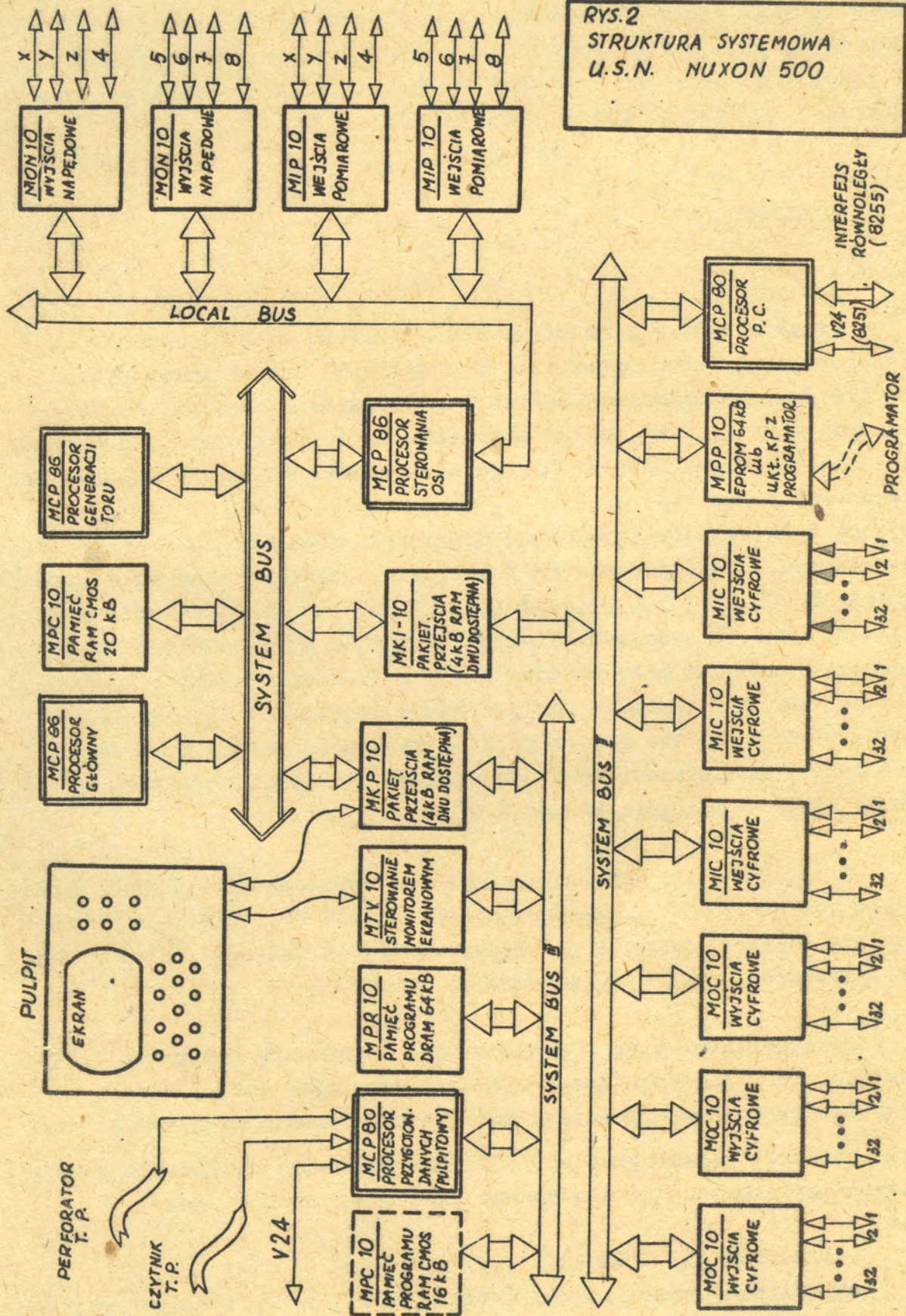
Interfejsy zewnętrzne to 2 x V24 /RS 232/ zrealizowane na układach 8251 i interfejs równoległy - układ 8255.

/ rysunek 2 /

Formalnie można układ zakwalifikować jako system trójmikrokomputerowy - jednym komputerem 16 - sto bitowym, trójprocesorowym sprzężonym z dwoma 8 - mio bitowymi jednoprocessorowymi. Takie zaklasyfikowanie wynika z wyróżnienia trzech szyn systemowych. Jednakże ze względu na ścisłe sprzężenie tych szyn i ich jednolitą budowę 16 - sto bitową budowę wszystkich modułów pamięciowych i wspólny zegar systemowy - taktujący przerwaniem cykliczną pracę całego systemu, można go uznać za pojedynczy komputer 16 - sto bitowy z pięcioma mikroprocesorami w tym dwa 8 - mio bitowe. Z kolei biorąc pod uwagę, że każdy moduł mikroprocesora 16 - bitowego zawiera również na pakiecie pewną ilość pamięci lokalnej i własny układ przerwań - można je uznać za mikrokomputery jednopakietowe /SBC - single board computer/ - otrzymujemy układ pięciokomputerowy.

Funkcjonalnie, niezależnie od głębokiej standaryzacji konstrukcyjnej, każdy moduł mikroprocesorowy jest specjalizowany z 1 wyraźnie głównym a inne są mu podległe. Wpływają na to pewne i-

RYS.2
STRUKTURA SYSTEMOWA
U.S.N. NUXON 500



stotne różnice w uzbrojeniu w elementy /przy zachowaniu identyczności druku/ a przede wszystkim zdecydowanie różne oprogramowanie. Zdaniem autora kryterium funkcji w systemie powinno przeważać i należy uznać, że mamy do czynienia z architekturą jednego komputera 16 - bitowego, pięcioprocessorowego.

4. MODUŁY.

4.1. MCP 86

Wszystkie trzy procesory szesnastobitowe są zrealizowane na identycznej płytce drukowanej. We wszystkich trzech przypadkach płytka zawiera mikroprocesor INTEL 8086 wraz z układami towarzyszącymi 8284 A, 8288, układ arbitrażu szyny 8289 oraz programowalny układ przerwań 8259 A. Mikroprocesor jest sterowany generatorem kwarcowym 5 MHz.

Oprócz tych układów płytka zawiera osiem podstawek dla kostek pamięciowych. Sześć pierwszych /czyli trzy pary/ są przeznaczone pod układy EPROM 2732 lub 2716, co daje 24 KB lub 12 KB a właściwie 12k lub 6k słów szesnastobitowych. Ostatnie dwie podstawki mogą zawierać następną parę układów EPROM lub, w zależności od ustawienia zworek, parę układów pamięci RAM o organizacji bajtowej TMS 4016 lub 6116. W ten sposób można w zależności od potrzeb i/lub możliwości zaopatrzeniowych otrzymać moduł z pamięcią stałą od 12 KB do 32 KB i pamięcią RAM od 4 KB do \emptyset .

4.1.1. PROCESOR GŁÓWNY posiada całą dostępną przestrzeń wypełnioną kostkami 2732 uzyskując na pakiecie 32 KB pamięci EPROM. Nie posiada lokalnej pamięci RAM wykorzystując w tym celu odrębny moduł pamięciowy -MCP - 10 zawierający 20 KB.

Procesor główny jest przeznaczony do nadzoru nad pracą całego systemu, koordynowania zadań wykonywanych przez inne procesory oraz interwencji w sytuacjach awaryjnych. Ponadto przetwarza program obróbki technologicznej /POT/ przygotowując cyklicznie porcje danych wejściowych dla następnego procesora, którym jest:

4.1.2. PROCESOR GENERACJI TORU.

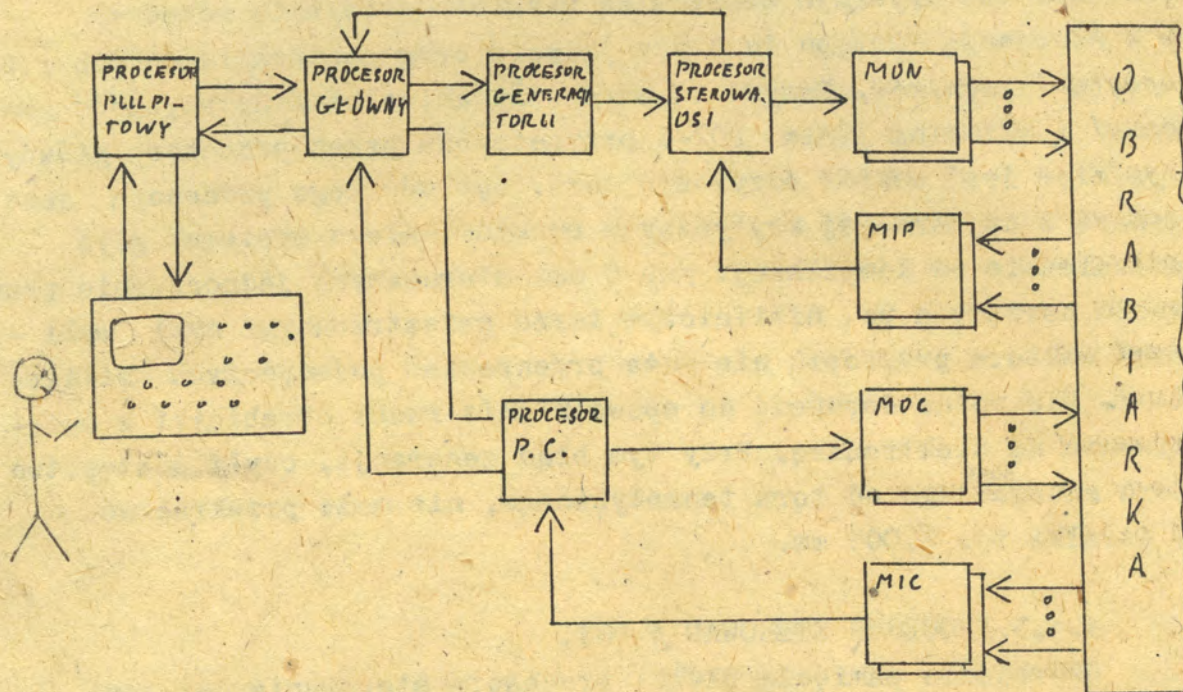
Płytką procesora generacji toru zawiera sześć układów pamięci

EPROM 2716 i dwa układy pamięci RAM - 4016 lub 6116. W ten sposób procesor ten uzyskuje maksymalną szybkość działania wchodząc w konkurencję dostępu do szyny jedynie przy pobieraniu danych lub odsyłaniu wyników. Danymi są parametry aktualne / np szybkość zadana/ i aktualna linia POT-u przygotowana przez procesor główny. Wynikiem jest wektor przyrostu toru. Szybkość tego procesora jest jednym z najbardziej krytycznych punktów całego systemu, gdyż niezależnie od komplikacji /np 5 osi sterowanych jednocześnie przy ruchu narzędzia po nieliniowym torze przestrzennym/ cykl wyliczeń wektora przyrostu nie może przekraczać pojedynczych milisekund. Nie można dopuścić do spowalniania ruchu obrabiarki w oczekiwaniu na elektronikę. Przy tym błąd generacji, czyli odstępstwo toru gener^{owa}lnego od toru teoretycznego, nie może przekraczać 1 działki tj. 0.001 mm.

4.1.3. PROCESOR STEROWANIA OSI.

Uzbrojenie pamięcią płytki procesora sterowania osi jest identyczne jak płytki procesora generacji toru tj 12 KB EPROM i 4 KB RAM. Istotną różnicą w stosunku do obu opisanych wyżej procesorów jest wykorzystanie złączy. Procesor ten posiada dodatkową szynę lokalną /rys.2 - LOCAL BUS/. Komunikacja z tą szyną odbywa się przez niewykorzystywany w standardzie, środkowy rząd styków w łączówkach. Fizycznie na druku wszystkich trzech procesorów 16-bitowych te połączenia istnieją ale jedynie w gnieździe procesora sterowania osi są one dołączone do druku plateru kasyety.

Procesor sterowania osi jest realizacją wielowymiarowego wektora sumacyjnego zadanego toru narzędzia i toru przez to narzędzie wykonanego. Z procesora generacji toru przez **SYSTEM BUS** i bufor w pamięci procesora głównego napływają wektory wartości zadanej a z modułów wejść pomiarowych /MIP 10/ przez LOCAL BUS procesor osi wylicza wektor ruchu wykonanego. Wypracowane następnie na tej podstawie wyrtości prędkości zostają przez LOCAL BUS i moduły wyjść MON 10 zadane do zespołu napędowego.



Rys. 3.

Struktura przepływu informacji

Czas cyklu obliczeń procesora sterowania osi jest równie krytyczny dla całego systemu jak i w przypadku procesora generacji toru. Stąd dążenie do obciążenia tych procesorów kosztem procesora głównego od prac pomocniczych i administracyjnych. Synchronizacja cyklu pracy tych procesorów jest wspomagana przerwaniem zegarowymi. Fizycznie zegar systemowy jest zlokalizowany na wolnym miejscu w module MKI 10 realizującym przejście z szyny SYSTEM BUS na szynę SYSTEM BUS II. W prototypie przyjęto 20 - sto milisekundowy cykl zegara.

4.2.MPC 10 PAMIĘĆ RAM CMOS.

Procesor główny nie posiada lokalnej pamięci RAM i do celów operacyjnych wykorzystuje poprzez SYSTEM BUS moduł pamięci MCP 10. Zawiera on 20 k-bajtów pamięci RAM CMOS z czego 16 KB jest zorganizowane w słowa 16 - sto bitowe a 4 KB bajtowo z wykorzystaniem tylko parzystych adresów. Zajmuje w ten sposób 24 kB przestrzeni adre-

sowej. Przewiduje się dwa wykonania z 16 KB i 20 KB. Wykonanie 20 KB będzie miało na pakiecie podtrzymanie zasilania 4 KB, które są przewidziane do pamiętania parametrów obrabiarki. Przy wypełnieniu jedynie pierwszych 16 KB moduł ten jest zamiennikiem modułu MPR 10 pamięci RAM szyny SYSTEM BUS III.

4.3. MKI 10 PAKIET PRZEJŚCIA DO PC między szynami systemowymi SYSTEM BUS i SYSTEM BUS III. Jest to dwudostępna pamięć RAM, zrealizowana na układzie TMS 4016 o pojemności 4 KB lub 2 KB. Przy mniejszej pojemności pamięć zajmuje co drugi adres /parzysty lub nieparzysty bajt/. Dostęp jest semaforowany z przeplotem i z rozstrzygnięciem ewntualnych konfliktów na korzyść szyny głównej.

Ponadto moduł zawiera zegar systemowy generujący przerwania co 20 ms.

4.4. MKP 10 PAKIET PRZEJŚCIA DO SZYNY SYSTEM BUS III. Moduł ten zorganizowany podobnie jak poprzedni obsługuje ponadto WEJŚCIE / WYJŚCIE pulpitem z wyłączeniem obsługi ekranu oraz czytnika i perforatora taśmy papierowej.

4.5. MCP 80 MODUŁ PROCESORA 8-mio BITOWEGO.

Jest to "klasyczny" mikrokomputer^{jednopakietowy} zrealizowany na mikroprocesorze MCY 7880 posiadający na pakiecie 16 KB pamięci PROM i 4KB pamięci RAM, system przerwań zrealizowany na układzie 8214 i wejście/wyjście na układach 8255 i 8251.

PROCESOR P.C. /programmable controller/ posiada rozszerzoną do 64 KB pamięć programu na pakiecie MPP 10 i obsługuje 96 wejść i 96 wyjść cyfrowych przejmując kontrolę i sterowanie wszystkiego w obrabiarce poza sterowaniem napędu. Chodzi tu na przykład o czujniki krańcowe, magazyn narzędzi, napęd wrzeciona i.t.p. W połączeniu z innymi modułami szyny SYSTEM BUS II może stanowić samodzielny kontroler.

PROCESOR PRZYGOTOWANIA DANYCH

- procesor pulpitem służy do programowej obsługi pulpitu łącznie z ekranem monitorowym i urządzeń WEJŚCIA / WYJŚCIA zewnętrznych nośników danych. Wczytane dane - program obsługi technologicznej /POT/ jest następnie przetwarzany i przechowywany w PAMIĘCI PROGRAMU - na module MPR 10 lub MPC 10.

4.6. MPP 10 PAMIĘĆ EPROM 64 KB.

Jest to pamięć programu stałego procesora P.C. Przewiduje się wykonania 16 KB, 32 KB i 64 KB. Dla celów uruchomionych można do modułu tego dołączyć programator EPROM-ów lub zamienić go modułem pamięci RAM z podtrzymaniem.

4.7. MIC 10 WEJŚCIA CYFROWE.

Moduł zawiera 32 wejścia 1-no bitowe z optoelektroniczną separacją galwaniczną. Charakteryzuje się wysyłaniem przerwania przy wykryciu zmiany stanu sygnałów wejściowych i filtracją zakłóceń $T = 20 \text{ ms/}$. Parametry sygnału wejściowego - 24V/6 mA. System zawiera trzy takie moduły.

4.8. MOC 10 WYJŚCIA CYFROWE.

Moduł zawiera 32 wyjścia 1-no bitowe o parametrach 24V/300 mA odseparowane optoelektronicznie od układu sterowania. System zawiera trzy takie moduły zainstalowane, podobnie jak i wejścia cyfrowe, na szynie P.C.

4.9. MPR 10 PAMIĘĆ PROGRAMU - DRAM 64 KB.

Moduł zawiera 64 KB pamięci dynamicznej 2118. Będzie stosowany w szynie procesora pulpituowego /SYSTEM BUS III/ do przechowywania programu obróbki technologicznej /POT/ w wypadku gdy zastosowanie obrabiarki będzie wymagać programów POT nie mieszczących się w 16 KB pamięci. Zasilanie pamięci programów POT będzie podtrzymywane z zewnątrz przez minimum 48 godzin.

4.10. MTV 10 STEROWANIE MONITOREM EKRAKOWYM.

Pulpit operatora został wyposażony w ekran monitorowy, do obsługi którego służy osobny moduł. Moduł zawiera generator znaków i pamięć treści ekranu. Ekran zawiera 12 linii po 40 pól znakowych. Każdy znak jest matrycą 8 x 12 punktów. Repertuar znaków obejmuje 128 znaków alfanumerycznych i pseudo-graficznych.

4.11. MIP 10 WEJŚCIA POMIAROWE.

Moduł ten zasila i odbiera sygnały z przetworników do pomiaru położenia. Moduł zawiera układy wytwarzania sygnałów zasilających rezolwer oraz układy odbierające sygnał z rezolwera i przetwarza

jące go na wartość cyfrową. Moduł obsługuje cztery zespoły pomiarowe.

4.12. MON 10 WYJŚCIA NAPĘDOWE.

Moduł wyjściowy sterowania napędem osi przetwarza cyfrową wartość prędkości, zadawaną z procesora sterowania osi, na sygnał analogowy ± 10 V do sterowania układów napędu posuwów z silnikami prądu stałego.

5. MAPA ADRESOWANIA SYSTEMU.

Komunikacja wewnątrz systemu, również z urządzeniami WE/WY - - pulpit i ekran monitora, odbywa się przez odpowiednie adresowanie pamięciowe. Pamięć lokalna modułu "przysłania" szynę systemową. Wystawienie przez procesor adresu spoza obszaru lokalnego powoduje odwołanie się poprzez tę szynę do innego modułu.

Szyny danych są 16-bitowe i organizacja modułów pamięciowych też jest 16 bitowa. Moduły procesorów 8-bitowych dokonują odpowiedniej komutacji parzystego i nieparzystego bajtu danych.

Mapę pamięci szyn systemowych pokazuje rysunek 4.

/ rysunek nr 4/

6. UWAGI KOŃCOWE.

Mechanicznie system NUXON 500 składa się z czterech modułów.

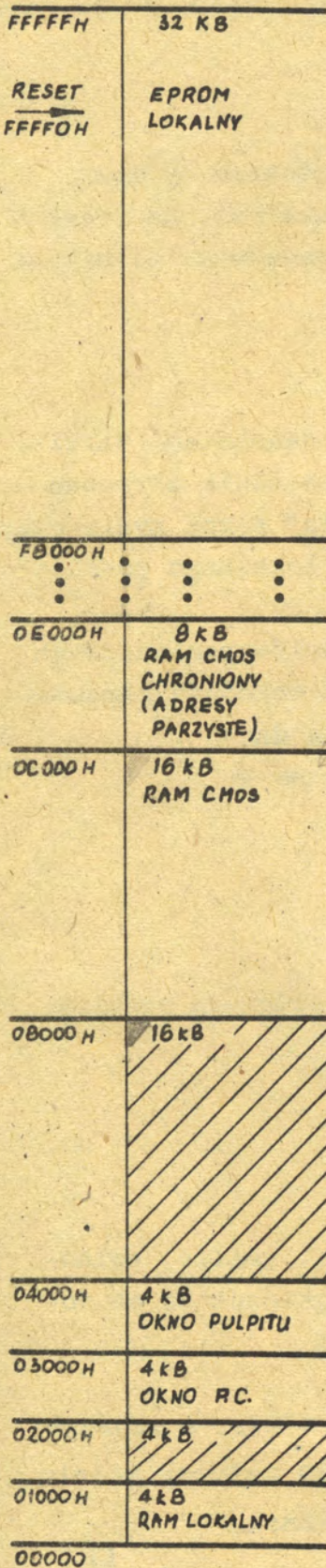
Są to:

- kasetta logiki /standard 19"/
- pulpit operatora
- czytnik ew. i/lub pamięć kasetowa
- zasilacz o łącznej mocy ok 500 VA.

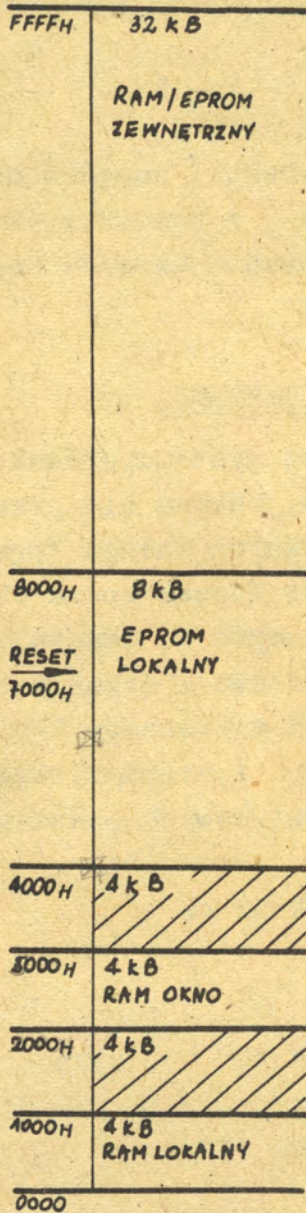
Moduły te mogą być montowane w szafce zawieszanej, szafie wolno stojącej lub rozdzielnie w obudowie obrabiarki. Wymagane są obudowy hermetyczne z zamkniętym obiegiem powietrza.

Plan przewiduje zestawienie modelu do końca 1984r. a wejście do produkcji w początku 1987 roku. Podany tu opis dotyczy modelu. W prototypach i produkcji może ulec istotnym zmianom.

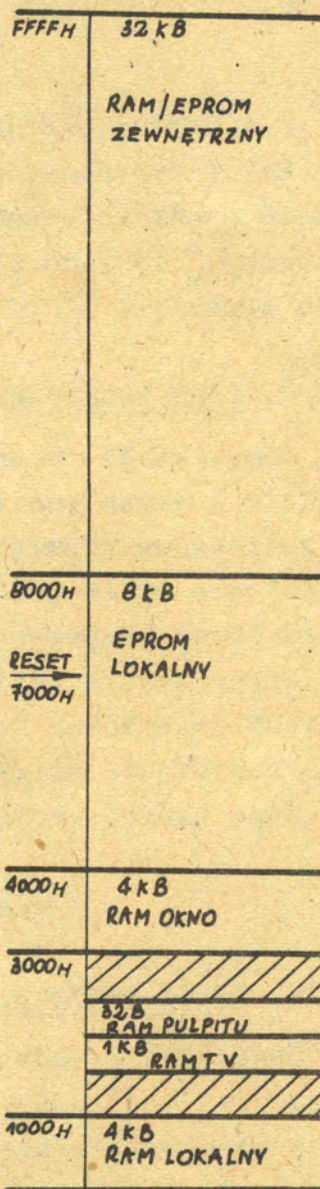
□



SYSTEM BUS



SYSTEM BUS II
(start programu od 7000H rozkazem JMP)



SYSTEM BUS III

RYS. 4.

MAPA ADRESOW
SZYN SYSTEMOWYCH

AUFWIEDERSEHN

