POLSKIE TOWARZYSTWO INFORMATYCZNE
CENTRUM SZKOLENIA INFORMATYCZNEGO
ZETO ŁÓDŹ

MIKROKOMPUTERY 16 i 32-bitowe

MIKROKOMPUTEROWE STACJE ROBOCZE

MATERIAŁY IV SZKOŁY MIKROKOMPUTEROWEJ
pod redakcją WOJCIECHA CELLAREGO

15–17 grudnia 1987 r.

POLSKIE TOWARZYSTWO INFORMATYCZNE
CENTRUM SZKOLENIA INFORMATYCZNEGO

ZETO ŁÓDŹ

MIKROKOMPUTERY 16 i 32-bitowe

MIKROKOMPUTEROWE STACJE ROBOCZE

MATERIAŁY IV SZKOŁY MIKROKOMPUTEROWEJ
pod redakcją WOJCIECHA CELLAREGO

15–17 grudnia 1987 r.

S P I S T R E Ś C I

1. Wojciech Cellary — 3
   Słowo wstępne
2. Jerzy Brzeziński, Piotr Krzyżagórski — 5
   Szesnastobitowy system mikroprocesorowy Intel 80286
3. Paweł Krysztofiak — 28
   Mikroprocesor 32-bitowy Intel 80386
4. Andrzej Woźniak — 52
   Rodzina mikroprocesorów 16/32 - bitowych Motorola 68000
5. Jerzy Brzeziński — 66
   System mikroprocesorowy iAPX432
6. Marek Maniecki — 83
   Wprowadzenie do stacji roboczych
7. Andrzej Ziemkiewicz — 94
   Komputer Fortune 32:16
8. Elżbieta Jezierska - Ziemkiewicz — 102
   Architektura RISC i mikrokomputer IBM RT
9. Janusz Rybnik, Jerzy Solak — 115
   Rodzina komputerów SUN-3
10. Wacław Iszkowski — 138
    System personalny PS/2
11. Wacław Iszkowski — 149
    System operacyjny OS/2

S Ł O W O W S T Ę P N E

Słowem "mikrokomputer" określa się dzisiaj całą klasę maszyn cyfrowych. Na dolnym krańcu znajdują się mikrokomputery domowe, które służą właściwie do szeroko pojętej zabawy. W środku, znajdują się mikrokomputery personalne o mniejszej lub większej mocy obliczeniowej, których najpopularniejszym przedstawicielem jest mikrokomputer IBM PC w wersji XT i AT. Natomiast na jej górnym krańcu znajdują się tak zwane mikrokomputerowe stacje robocze. Mikrokomputerowe stacje robocze są mikrokomputerami o dużych mocach obliczeniowych. Stosowane są w nich mikroprocesory trzydziestodwubitowe. Ich pamięci operacyjne są rzędu 10MB, a pamięci zewnętrzne rzędu setek MB. Ich cechą charakterystyczną jest ukierunkowanie na graficzny kontakt z użytkownikiem. Mikrokomputerowe stacje robocze posiadają w tym celu zarówno odpowiedni sprzęt - monitory graficzne o dużej powierzchni ekranu, wysokiej rozdzielności i bogatej palecie barw, drukarki graficzne, różnego rodzaju manipulatory, pióra świetlne, digitizery, itd., - jak i odpowiednie oprogramowanie. Moc obliczeniowa stacji roboczych jest co najmniej porównywalna z mocami obliczeniowymi minikomputerów z końca lat siedemdziesiątych, ale zapewniony przez nie komfort pracy i jakość ich oprogramowania narzędziowego jest zdecydowanie wyższa, głównie ze względu na rozwiązania graficzne. Współczesne laboratorium informatyczne zbudowane przy wykorzystaniu mikrokomputerowych stacji roboczych składa się z kilku stanowisk pracujących w trybie jednodostępnym, połączonych w szybką sieć komputerową, do której są dołączone również inne maszyny - mikrokomputery i minikomputery stosowane w tym laboratorium.

Główny zakres zastosowań mikrokomputerowych stacji roboczych to przede wszystkim szeroko rozumiane wspomaganie projektowania i zarządzania. Możliwości graficzne mikrokomputerowych stacji roboczych , ich duża moc obliczeniowa wynikająca z zastosowania mikroprocesorów trzydziestodwubitowych i jednodostępnego trybu pracy oraz możliwości szybkiej wymiany informacji za pomocą sieci komputerowych powodują, że są one

aktualnie najlepszym narzędziem w tej dziedzinie.

Celem tej szkoły jest zaprezentowanie problematyki mikrokomputerowych stacji roboczych. Przyjęto taki program szkoły, aby objąć nią również mikrokomputery o mocy obliczeniowej większej od IBM PC AT nawet jeśli nie mają wszystkich cech mikrokomputerowych stacji roboczych. W programie szkoły wyróżniono dwa nurty. W pierwszym z nich zaprezentowano same mikroprocesory i ich cechy charakterystyczne umożliwiające budowę stacji roboczych. Omawiane są mikroprocesory Intel iAPX286, iAPX386, iAPX432, Motorola 68000 oraz architektura RISC (ang. Reduced Instruction Set Computer).

W drugim nurcie są omawiane i porównywane najbardziej znane mikrokomputerowe stacje robocze: Fortune, IBM RT, SUN i IBM PS/2. Warto zwrócić uwagę na ostatni z przedstawionych systemów - IBM PS/2 - działający pod kontrolą nowego systemu operacyjnego OS/2, któremu poświęcony jest osobny wykład. System ten, a ściśle rzecz biorąc - rodzina systemów, pretenduje do zajęcia w najbliższych latach takiej samej pozycji jak mikrokomputer IBM PC.

Założenia programowe tej szkoły nie uległy zmianie w stosunku do całego cyklu. Przypomnijmy, że Szkoła ma charakter informacyjny oraz że jest ona skierowana do osób zawodowo zajmujących się informatyką, znających problematykę mikrokomputerów i pragnących zapoznać się z problematyką mikrokomputerowych stacji roboczych.

Organizatorzy wyrażają nadzieję, że IV Szkoła Mikrokomputerowa przyczyni się do rozszerzenia wiedzy na temat mikrokomputerów i przełamania opinii jakoby "świat" kończył się na mikrokomputerze IBM PC wersja AT.

Wojciech Cellary

Poznań, październik 1987.

IV Szkoła CSI/PTI
Mikrokomputery 16 i 32-bitowe
Łódź, grudzień 1987

Szesnastobitowy system mikroprocesorowy Intel 80286

dr inż. Jerzy Brzeziński
mgr inż. Piotr Krzyżagórski
Instytut Automatyki
Politechniki Poznańskiej
ul. Piotrowo 3A
60-965 Poznań, tel.78-23-78

1.WPROWADZENIE

Na początku lat osiemdziesiątych mikroprocesory rodziny Intel 8086 znalazły zastosowanie w wielu systemach przemysłowych i profesjonalnych mikrokomputerach osobistych. Dalsze rozszerzanie ich zastosowań było jednak silnie ograniczone brakiem sprzętowych mechanizmów ochrony pamięci, stosunkowo niską efektywnością obliczeń i niewielkim rozmiarem przestrzeni adresowej. Podstawowe znaczenie miała pierwsza z wymienionych wad, która wykluczała w praktyce zasadność bezpośredniego stosowania tego mikroprocesora do realizacji złożonych systemów wielodostępnych. Spełnienie wymagań tego istotnego obszaru zastosowań wymagało konstrukcji nowego układu. Rozwiązaniem zaproponowanym przez firmę Intel był 16-bitowy mikroprocesor 80286. Elementem jakościowo nowym w mikroprocesorze 80286 jest rozbudowany sprzętowy mechanizm zarządzania pamięcią umożliwiający ochronę i wirtualizację pamięci. Przypomnijmy, że wirtualizacja pamięci polega na stworzeniu użytkownikowi wrażenia dysponowania praktycznie nieograniczoną pamięcią operacyjną (pamięcią wirtualną), przy użyciu ograniczonej rzeczywistej pamięci operacyjnej, pamięci zewnętrznej oraz specjalnych środków sprzętowych i programowych, zapewniających automatyczną wymianę informacji pomiędzy pamięcią operacyjną a zewnętrzną. Widziana przez programistę przestrzeń adresów wirtualnych jest transformowana automatycznie przez system w przestrzeń adresów fizycznych komórek pamięci operacyjnej. Punktem wyjścia dla realizacji pamięci wirtualnej jest jej podział

na rozłączne i spójne fragmenty o jednakowym lub dowolnym rozmiarze, nazywane odpowiednio stronami i segmentami. W pierwszym przypadku mówimy o stronicowanej pamieci wirtualnej, w drugim natomiast o pamięci segmentowanej. Należy podkreślić, że użytkownik nie musi być świadomy tego podziału. Ograniczymy się tu do krótkiego omówienia pamięci segmentowanej, z racji możliwości jej realizacji, przy użyciu mikroprocesora 80286.

W systemach z pamięcią wirtualną program operuje na adresach wirtualnych, które w przypadku pamięci segmentowanej składają się z numeru segmentu i adresu w segmencie. W trakcie wykonywania programu generowane są adresy wirtualne operandów (argumentów operacji i rozkazów). Adresy te muszą zostać dynamicznie odwzorowane w adresy fizycznych komórek pamięci operacyjnej zawierających żądane jednostki. Ilustrowany na rys.2 proces transformacji adresów wirtualnych w adresy fizyczne jest realizowany z wykorzystaniem tablicy, której poszczególne elementy, zwane deskryptorami, są przyporządkowane jednoznacznie segmentom pamięci wirtualnej. Deskryptory zawierają aktualne fizyczne adresy bazowe segmentów im przyporządkowanych oraz warunki dostępu do tych segmentów. Specjalizowana sprzętowa jednostka transformacji adresu, odczytuje automatycznie deskryptor adresowanego segmentu. Jeśli żądany segment znajduje się aktualnie w pamięci operacyjnej i spełniane są wymagania określone przez warunki dostępu, to zawarty w deskryptorze adres fizyczny segmentu zostaje zsumowany z adresem w segmencie, tworząc w ten sposób adres fizyczny odwzorowanego operandu. W ogólności, może jednak zaistnieć sytuacja, że żądanego segmentu brak jest w danej chwili w pamięci operacyjnej lub dostęp do niego jest niedozwolony. W obu przypadkach generowane są automatycznie przerwania, które powodują przekazanie sterowania do systemu operacyjnego w celu obsługi błędu dostępu lub sprowadzenia żądanego segmentu. Po dokonaniu tego sprawdzenia, uaktualnieniu deskryptora i wyznaczeniu adresu segmentu w pamięci opreracyjną - program może być oczywiście natychmiast kontynuowany. Należy podkreślić, że wszystkie operacje związane z transformacją adresów, kontrolą warunków dostępu, sprawdzaniem obecności segmentu w pamięci operacyjnej i ewentualnym jego sprowadzeniem - są realizowane bez jakiegokolwiek udziału użytkownika. Użytkownik ma zatem wrażenie, że wszystkie segmenty jego programu są ciągle w pamięci operacyjnej. Tym samym,

nawet dla dowolnie dużych programów jest on całkowicie zwolniony od konieczności zarządzania wymianą informacji między pamięcią operacyjną a pamięcią zewnętrzną, co stanowi podstawową zaletę wirtualizacji pamięci z punktu widzenia użytkownika.

Ważne znaczenie ma również fakt, że systemowe mechanizmy ochrony dostępu i wirtualizacji pamięci zapewniają też pełną seperację i niezależność zadań użytkowników. Oznacza to, że wszelkie (celowe lub przypadkowe) próby dostępu do danych czy programów innych użytkowników są w pełni kontrolowane przez odpowiednie mechanizmy sprzętowe i programowe. Bez akceptacji systemu operacyjnego dostęp do obszarów innych użytkowników nie jest zatem możliwy. Należy jeszcze dodać, że wirtualizacji wraz z efektywnymi mechanizmami przełączania zadania umożliwia również znaczny wzrost poziomu wykorzystania pamięci operacyjnej i innych zasobów systemu komputerowego, w wyniku zwiększenia liczby współbieżnie wykonywanych zadań.

Rozważając zalety pamięci wirtualnej trzeba też jednak wspomnieć, że realizacja takiej pamięci związana jest z różnymi kosztami systemowymi, do których można zaliczyć : opóźnienie wnoszone przez jednostkę transformacji adresów, koszt realizacji systemowych procedur wymiany informacji, koszt pamiętania tablic deskryptorów, itd.

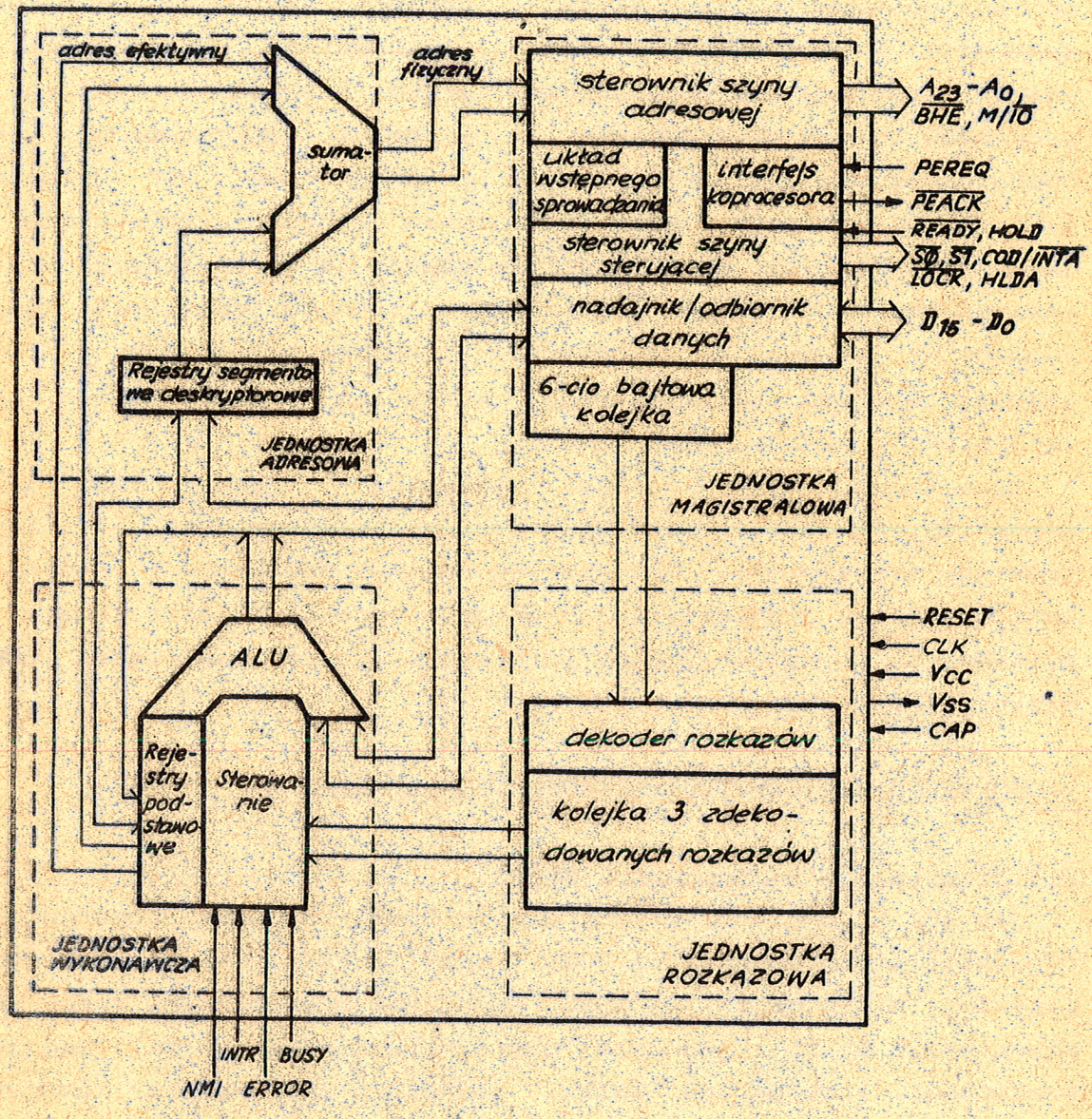
2. OGÓLNA CHARAKTERYSTYKA MIKROPROCESORA 80286

2.1 Struktura mikroprocesora 80286

Jak już wspomniano, mikroprocesor 80286 stanowi konstrukcyjne rozwinięcie mikroprocesora 8086 firmy Intel. Składa się on z czterech, a nie dwóch jak 8086, jednostek operacyjnych pracujących równolegle. Są to : jednostka magistralowa (bus unit), jednostka rozkazowa (instruction unit), jednostka wykonawcza (execution unit) oraz jednostka adresowa (address unit).
Schemat połączeń tych jednostek i struktura mikroprocesora 80286 przedstawia rys.1.

Podstawowym zadaniem jednostki magistralowej jest wysterowywanie i obsługa linii magistralowych, obejmujących: 24-bitową szynę adresową, 16-bitową szynę danych oraz szynę sterującą. Jednostka ta realizuje na żądanie innych jednostek operacje związane z zapisem i odczytem z pamięci operacyjnej

danych i rozkazów. Jednocześnie, jednostka magistralowa może w sposób autonomiczny wczytywać kolejne rozkazy do 6-bajtowej wewnętrznej kolejki.



Rys.1 Struktura mikroprocesora 80286.

Jednostka rozkazowa pobiera rozkazy z kolejki wewnętrznej jednostki magistralowej i realizuje operacje dekodowania rozkazów. Zdekodowane rozkazy zostają z kolei umieszczone w kolejce

wewnętrznej jednostki rozkazowej.

Zadaniem jednostki wykonawczej jest automatyczne pobieranie zdekodowanych rozkazów z wewnętrznej kolejki jednostki rozkazowej. Następnie rozkaz jest interpretowany, co wiąże się w ogólności wynaczeniem efektywnego adresu argumentu i realizacją odpowiadających rozkazowi operacji przetwarzania informacji.

Jednostka adresowa odwzorowuje adres efektywny w adres fizyczny przekazywany do jednostki magistralowej. Operacje związane z tym odwzorowaniem mogą być dość złożone i zostaną omówione szczegółowo w dalszej części artykułu. W tym miejscu powiemy tylko ogólnie, że jednostka adresowa zawiera podstawowe mechanizmy sprzętowe ochrony i wirtualizacji pamięci.

Wymienione cztery podstawowe jednostki operacyjne mikroprocesora Intel 80286 pracują równolegle i częściowo niezależnie. W efekcie, operacje pobierania, dekodowania i wykonywania kolejnych rozkazów są wykonywane jednocześnie. Zmniejsza to istotnie średni czas wykonywania rozkazu oraz częściowo uniezależnia procesor od dostępu do magistrali.

W porównaniu z mikroprocesorem 8086, mikroprocesor 80286 zawiera szereg udoskonaleń i rozwinięć jakościowych. Do grupy udoskonaleń można zaliczyć : rozdzielenie magistrali adresowej i danych, rozszerzenie przestrzeni adresowej, zwiększenie efektywności mechanizmu synchronizacji z koprocesorem, zmniejszenie średniego czasu wykonywania instrukcji w wyniki zwiększenia liczby równolegle pracujących jednostek operacyjnych. W wyniku tych udoskonaleń uzyskano wielokrotny wzrost efektywności obliczeń w stosunku do standardowej wersji mikroprocesora 8086 (od 3.2 - dla rozkazów skoków warunkowych, do 10.4 - dla rozkazów dzielenia liczb 32-bitowych). Rozwinięcia jakościowe wiążą się natomiast z mechanizmami ochrony i wirtualizacji pamięci oraz sprzętowym wspomaganiem procesu przełączenia zadań.

2.2 Tryby pracy mikroprocesora 80286

Ochrona i wirtualizacja pamięci oraz sprzętowe mechanizmy przełączania zadań są szczególnie pożądane w wielodostępnych systemach komputerowych. Realizacja tych wszystkich mechanizmów w mikroprocesorze 80286 umożliwa zatem budowę z jego użyciem efektywnych i niezawodnych wielodostępnych systemów mikrokomputerowych. Jak wiadomo, takich możliwości bezpośrednio

nie dawał mikroprocesor 8086. W porównaniu z nim mikroprocesor 80286 spełnia zatem wymagania znacznie szerszego obszaru zastosowań. Warto w tym miejscu jednak przypomnieć, że mechanizmy ochrony i wirtualizacji wprowadzają jednak pewne dodatkowe koszty systemowe. Koszty te nie we wszystkich jednak zastosowaniach są uzasadnione. Ponadto, w wielu praktycznych przypadkach bardzo duża wartość ma zachowanie zgodności programowej z poprzednimi mikroprocesorami danej linii.

W celu pogodzenia przeciwstawnych w pewnym zakresie dążeń do poszerzania zakresu zastosowań, przez wprowadzanie nowych sprzętowych mechanizmów zarządzania pamięcią, z potrzebą zachowania w pewnych przypadkach zgodności ze starszymi rozwiązaniami, zastosowano w mikroprocesorze 80286 mechanizm programowej zmiany trybu pracy.

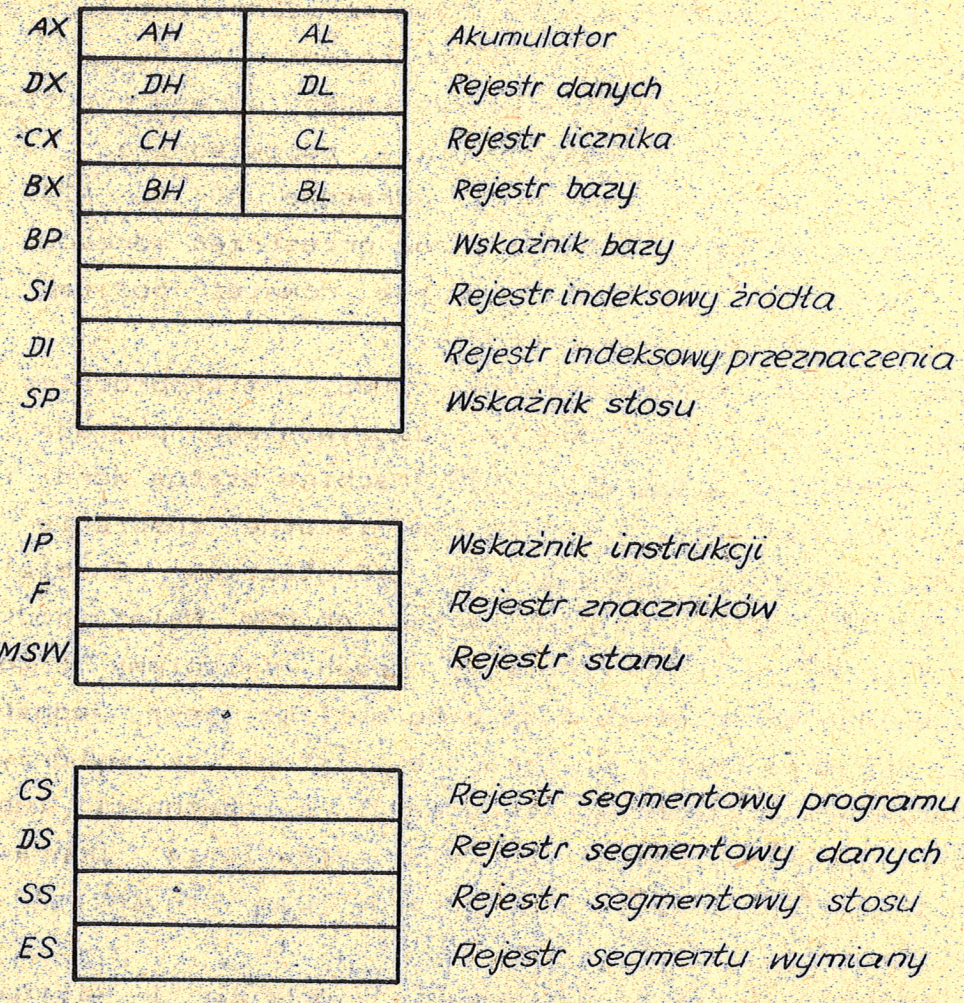
Po sprzętowym wyzerowaniu lub załączeniu zasilania mikroprocesor 80286 pracuje w trybie adresowania rzeczywistego (Real Address Mode), w którym jest całkowicie zgodny od strony architektonicznej z mikroprocesorem 8086. Oznacza to, że oba procesory posiadają takie same : zestawy rejestrów, przestrzenie adresowe, mechanizmy adresowania pamięci i systemy przerwań. Nowe możliwości mikroprocesora 80286 uzyskuje się po jego programowym przełączeniu w tryb pracy z ochroną i wirtualizacją pamięci. W trybie tym dostępne są mechanizmy sprzętowe umożliwiające realizację segmentowanej pamięci wirtualnej o maksymalnym rozmiarze 16GB dla każdego zadania. Rośnie przestrzeń adresów fizycznych z 1MB do 16MB. Dostępne stają się również sprzętowe mechanizmy przełączania zadań oraz dodatkowe rozkazy, umożliwiające efektywne operowanie na nowych sprzętowych mechanizmach zarządzania pamięcią.

3. ARCHITEKTURA MIKROPROCESORA 80286

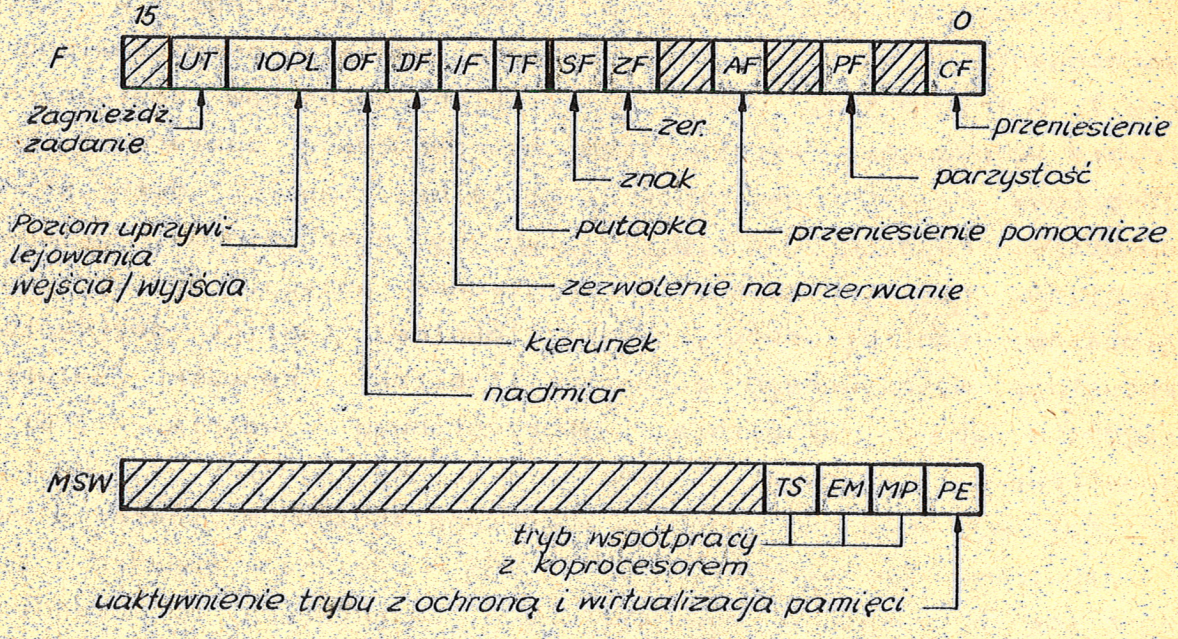
3.1 Architektura w trybie adresowania rzeczywistego

Po załączeniu zasilania lub wyzerowaniu mikroprocesor 80286 pracuje w trybie adresowania rzeczywistego, w którym jest architektonicznie zgodny z mikroprocesorem 8086. To ostatnie spostrzeżenie potwierdza niemal w pełni rys.2a, przedstawiający architekturę mikroprocesora 80286 w trybie adresowania rzeczywistego. Jak widać zestaw podstawowych rejestrów jest tu

a)



b)



Rys. 2. a) Architektura mikroprocesora w trybie pracy z adresacją rzeczywistą
b) Format rejestru znaczników i rejestru stanu.

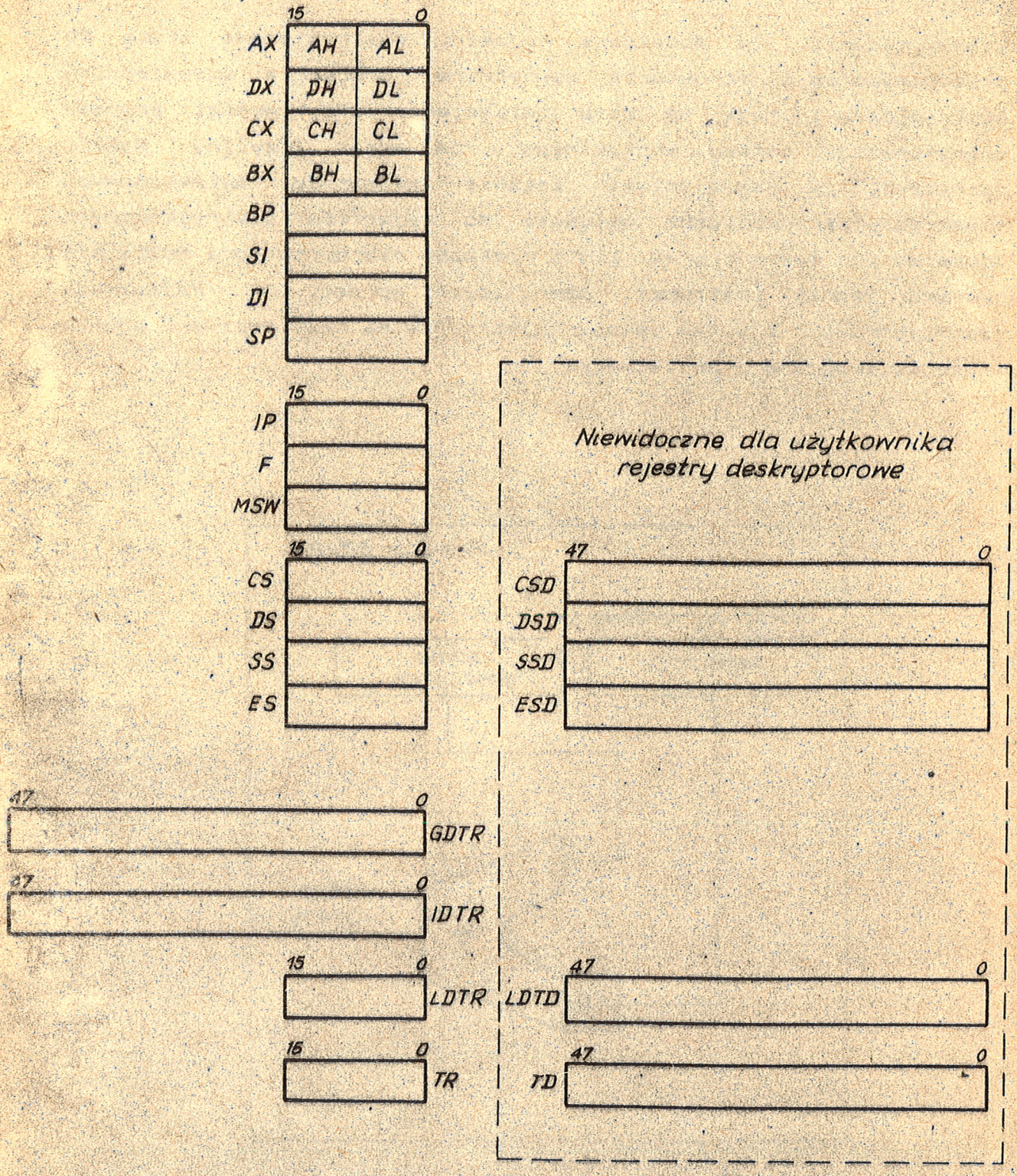
taki sam jak i w procesorze 8086. Również identyczne są dostępne tryby adresowania, jak i sposób obliczania adresu fizycznego. Tworzony jest on przez zsumowanie wynikowego adresu efektywnego z pomnożoną przez 16 zawartością odpowiedniego rejestru segmentowego. W trybie tym bity adresowe A0 do A23 nie są wysterowywane , co ogranicza fizyczną przestrzeń adresową pamięci do 1MB. Taki sam jak w 8086 jest również podział pamięci operacyjnej na obszary zarezerwowane.

W stosunku do mikroprocesora 8086, mikroprocesor 80286 pracujący w trybie adresowania rzeczywistego posiada jednak dodatkowo maszynowe słowo stanu MSW (Machine Status Word) i nieco bogatszą listę rozkazów. W trybie tym istotne znaczenie ma bit uaktywniania ochrony pamięci PE (Protection Enable) oraz instrukcja LMSW, wpisująca nową wartość do MSW. Należy podkreślić, że ustawiając PE w stan logiczny jeden zmieniamy tryb pracy mikroprocesora. Wyzerowanie tego bitu możliwe jest jednak tylko sprzętowo sygnałem RESET. Pozostałe bity stanu są wykorzystywane do przekazywania procesorowi informacji o obecności lub braku koprocesora arytmetycznego, w celu zapewnienia poprawnej i efektywnej realizacji złożonych operacji numerycznych.

3.2. Architektura w trybie adresowania z ochroną i wirtualizacja pamięci.

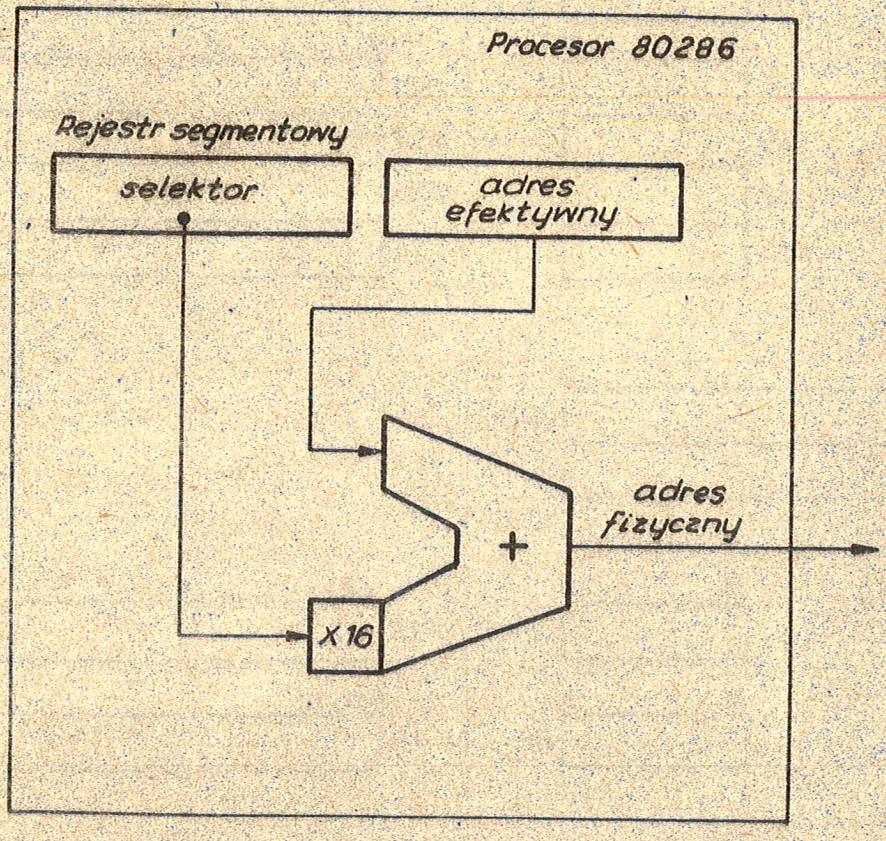
Architekturę mikroprocesora 80286 pracującego w trybie adresowania z ochroną i wirtualizacją pamięci przedstawiono na rys.3. W trybie tym, oprócz wszystkich rejestrów dostępnych w trybie adresowania rzeczywistego, mikroprocesor zawiera także : 48-bitowy rejestr deskryptora globalnej tablicy deskryptorów, 48-bitowy rejestr deskryptora tablicy deskryptorów przerwań, 16-bitowy rejestr selektora (względnej pozycji deskryptora w globalnej tablicy deskryptorów) lokalnej tablicy deskryptorów aktualnie wykonywanego zadania, 16-bitowy rejestr selektora segmentu stanu aktualnie wykorzystanego zadania, 3-bitowe dodatkowe pole w rejestrze flag. Wymienione rejestry są dostępne programowo i mogą być dowolnie zmieniane stosownymi instrukcjami.

Zauważmy, że mikroprocesor 80286 zawiera ponadto sześć 48-bitowych rejestrów deskryptorowych skojarzonych odpowiednio z rejestrami segmentowymi (CS, DS, SS, ES), rejestrem selektora lokalnej tablicy deskryptorów oraz rejestrem selektora segmentu



Rys. 3. Architektura mikroprocesora 80286 w trybie pracy z ochroną i wirtualizacją pamięci.

stanu zadania. Te dodatkowe rejestry deskryptorowe służą do przechowywania w strukturze wewnętrznej procesora odpowiednich deskryptorów i mają na celu zwiększenie efektywności procesu transformacji adresu wirtualnego na adres fizyczny. Należy zaznaczyć, że jakiejkolwiek zmianie rejestrów selektorowych towarzyszy automatyczne wpisanie do rejestrów deskryptorowych odpowiednich deskryptorów. Lista rozkazów mikroprocesora 80286 nie zawiera jednak instrukcji operujących wprost na rejestrach deskryptorowych i w tym sensie rejestry te są niewidoczne z punktu widzenia programu i użytkownika.



Rys. 4. Mechanizm tworzenia adresu fizycznego w trybie z adresacją rzeczywistą.

4. MECHANIZMY ZARZĄDZANIA PAMIĘCIĄ W MIKROPROCESORZE 80286

4.1. Zarządzanie pamięcią w trybie adresowania rzeczywistego

W trybie adresowania rzeczywistego jednostka adresowa mikroprocesora 80286 pełni dokładnie te same funkcje co odpowiednia jednostka mikroprocesora 8086. Przypomnijmy więc tylko, że adres fizyczny tworzony jest w tym trybie przez zsumowanie wynikowego adresu efektywnego z przesuniętą o cztery pozycje w lewo zawartością odpowiedniego rejestru segmentowego, dając w efekcie 20-bitowy adres fizyczny. Z punktu widzenia użytkownika przestrzeń pamięci operacyjnej zawiera zatem cztery segmenty, każdy o rozmiarze 64 KB. Segmenty te nie muszą być w ogólności rozłączne. Obszar pamięci w przedziale adresów 0H do 3FFH zarezerwowany jest na tablicę wskaźników przerwań (IP i CS). Natomiast obszar w przedziale FFF0H do FFFFH jest zarezerwowany na początkowy fragment procedury inicjacji systemu (bajt o adresie FFF0H zawiera kod pierwszego rozkazu po wyzerowaniu systemu).

4.2. Zarządzanie pamięcią w trybie adresowania z ochroną i wirtualizacją pamięci

W trybie pracy z ochroną i wirtualizacją pamięci nową zupełnie rolę pełnią rejestry segmentowe. Pamiętane są w nich mianowicie tzw. selektory składające się z 13-bitowego numeru segmentu, bitu wyboru jednej z dwóch tablic deskryptorów oraz dwubitowego pola żądanego poziomu uprzywilejowania RPL. Wynikowy adres efektywny może być natomiast interpretowany jako adres w segmencie. Połączenie numeru segmentu z adresem w segmencie tworzy pełen adres wirtualny.

Zgodnie z ogólnym mechanizmem trasformacji adresu wirtualnego, selektor określa względną pozycję deskryptora w określonej tablicy deskryptorów. W przypadku, gdy adresowany segment znajduje się w pamięci operacyjnej, wskazany deskryptor zawiera fizyczny adres bazowy tego segmentu. W celu wyznaczania adresu fizycznego operandu należy zatem dodać do adresu bazowego adres w segmencie.

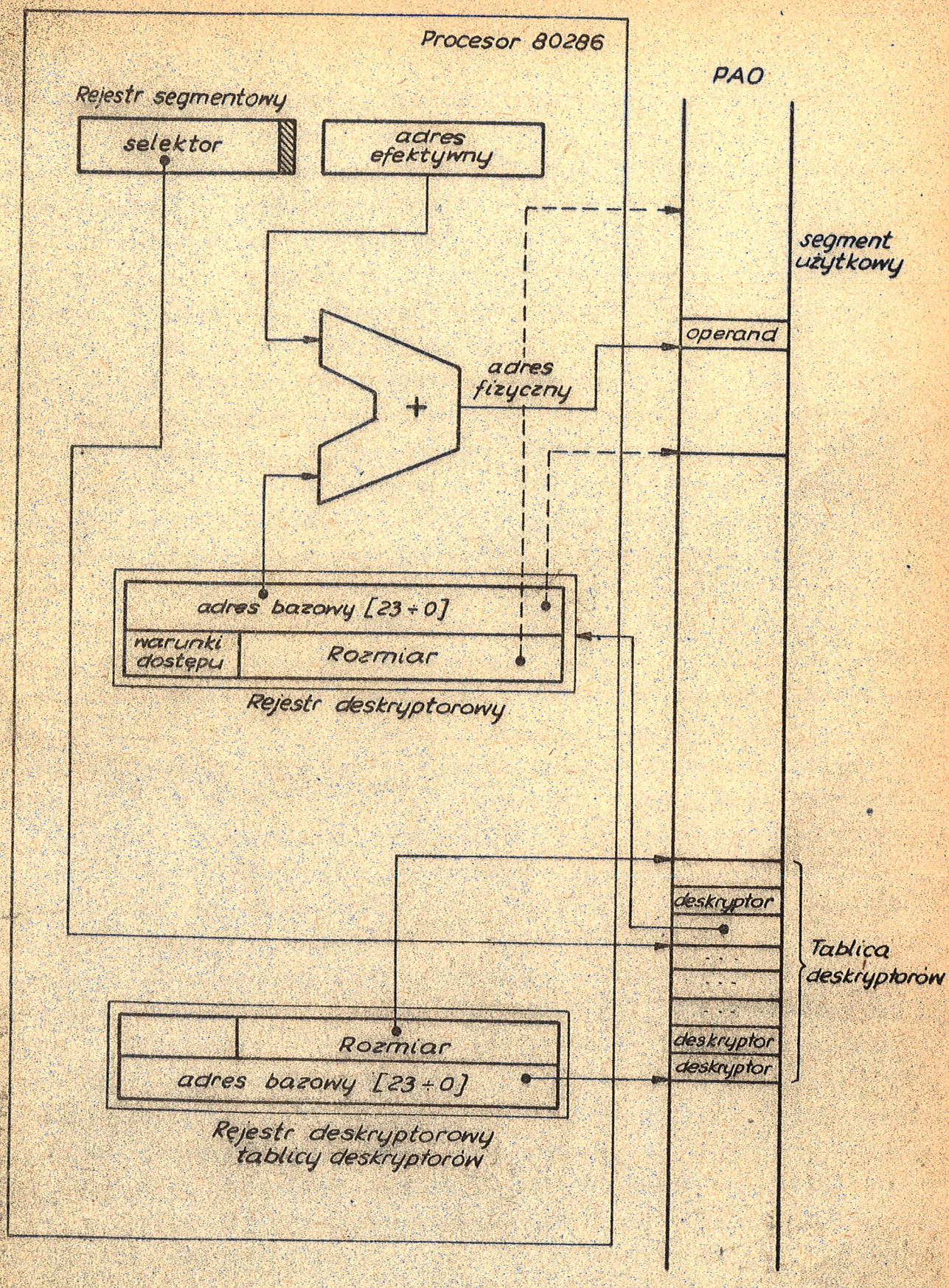
Ten klasyczny schemat transformacji adresu wirtualnego został zrealizowany w mikroprocesorze 80286 bardzo efektywnie przy użyciu niewidocznych dla użytkownika rejestrów deskryptorowych. Jak już wspomniano, do rejestrów tych wpisywany jest automatycznie nowy

deskryptor przy każdej zmianie zawartości rejestru segmentowego, a więc - przy każdej zmianie selektora. Jednostka zarządzania pamięcia może zatem korzystać z bieżącej wartości odpowiedniego rejestru deskryptorowego, bez konieczności każdorazowego odczytywania deskryptora z pamięci operacyjnej (zob. rys. 5). Opisywany mechanizm w połączeniu z rozróżnieniem niezależnych rejestrów segmentowych kodu, danych, stosu i wymiany - zapewnia w praktyce wysoką efektywność procesu transformacji adresu wirtualnego.

W dotychczasowej prezentacji omówione zostało jedynie zastosowanie deskryptora w procesie wyznaczania adresu fizycznego adresowanego operandu. Należy jednak zaznaczyć, że deskryptory pełnić mogą również szereg ważnych funkcji związanych z ochroną dostępu do poszczególnych segmentów, wywoływaniem procedur obsługi przerwań , przełączaniem współbieżnie wykonywanych zadań, itd. Bliższe omówienie tych funkcji poprzedzimy jednak bardziej szczegółową prezentacją samych deskryptorów.

W systemie 80286 wyróżnia się dwa podstawowe typy deskryptorów: deskryptory segmentów użytkowych (segmentów danych i kodów) oraz deskryptory systemowe (sterujące i segmentów danych systemowych). Deskryptory segmentów użytkowych składają się z 20-bitowego fizycznego adresu bazowego segmentu, 16-bitowego rozmiaru, 8-bitowego pola charakterystycznego oraz 16-bitowego pola rezerwowego. Format deskryptora tego typu przedstawiono na rys. 6a. Pola charakterystyczne deskryptora tworzą:

- bit P (Present) - obecność segmentu w pamięci fizycznej (jeżeli P=0, to pozostałe pola nie są wykorzystywane w znaczeniu opisywanym poniżej);
- dwubitowe pole DPL (Descriptor Privilage Level) - poziomu uprzywilejowania deskryptora, wykorzystywane przez mechanizm ochrony dostępu do segmentów;
- bit S (Segment Descriptor) - stanowiący wyróżnik deskryptora segmentu użytkowego (dla omawianego typu deskryptora S=1);
- bit E (Executable) - rodzaju segmentu (danych - E=0 albo kodu - E=1);
- dwubitowe pole OP (Operation) - dopuszczalnych operacji na segmencie (dla segmentów danych specyfikuje się atrybut zezwolenia na zapis i znacznik segmentu stosu, natomiast dla segmentów kodu - atrybut zezwolenia na odczyt i atrybut



Rys. 5 Mechanizm tworzenia adresu fizycznego w trybie z ochroną i wirtualizacją pamięci.

weryfikacji wymaganego poziomu uprzywilejowania zadania ubiegającego się o dostęp do danego segmentu);

- bit A (Accessed) - znacznika odwołania do segmentu (A=0 oznacza, że od momentu wyzerowania bitu A nie nastąpiło jeszcze żadne odwołanie do segmentu opisywanego przez ten deskryptor).

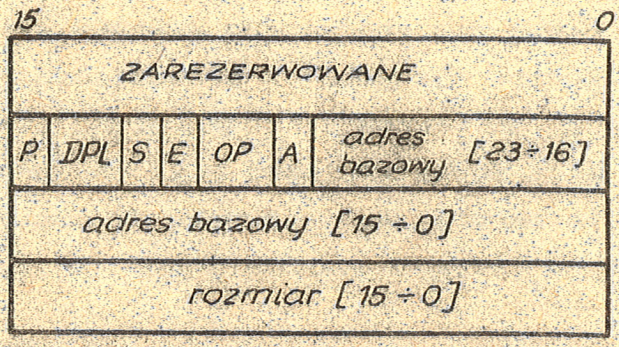
Wskaźnik P obecności segmentu w pamięci fizycznej jest sprawdzany automatycznie przez jednostkę zarządzania pamięcią mikroprocesora 80286 w momencie zmiany zawartości rejestru segmentowego. Jeżeli adresowany segment nie znajduje się w pamięci operacyjnej, to generowany jest automatycznie błąd segmentu i tym samym sterowanie zostaje przekazane procedurom systemu operacyjnego, odpowiedzialnym za sprowadzanie żądanego segmentu do pamięci operacyjnej. Jeżeli w chwili wystąpienia błędu segmentu brak jest nadto w pamięci operacyjnej wolnego obszaru spójnego o rozmiarze niemniejszym od rozmiaru żądanego segmentu, to system operacyjny dokonuje relokacji bądź usunięcia pewnych segmentów, w celu utworzenia w pamięci wymaganego obszaru wolnego. Dopiero wówczas, może zostać zainicjowana operacja wczytania z pamięci zewnętrznej żądanego segmentu.

Wybór segmentu, który należy usunąć ma istotny wpływ na efektywność systemów. Zauważmy bowiem, że usunięcie segmentu, który będzie niezbędny w najbliższej przyszłości, prowadzi do kolejnego błędu segmentu i opóźnienia związanego z jego obsługą. System operacyjny powinien oczywiście dążyć do minimalizacji kosztów wymiany informacji. Tak postawiony problem jest w ogólności bardzo złożony. Stąd też w praktyce powszechnie stosuje się heurystyczne algorytmy wymiany wykorzystujące znaczniki odwołania do segmentów np. algorytm FINUFO (First-In-Not-Used-First-Out). Efektywną realizację tego typu algorytmów w systemach z mikroprocesorem 80286 umożliwia mechanizm ustawiania bitu A w stan 1, przy każdorazowym wczytaniu deskryptora do rejestru deskryptorowego procesora.

Jak już wspomniano drugą klasę deskryptorów w systemie 80286 tworzą deskryptory systemowe. Opisują one segmenty specjalnych danych systemowych oraz mechanizmy przekazywania sterowania, uwzględniające ochronę pamięci.

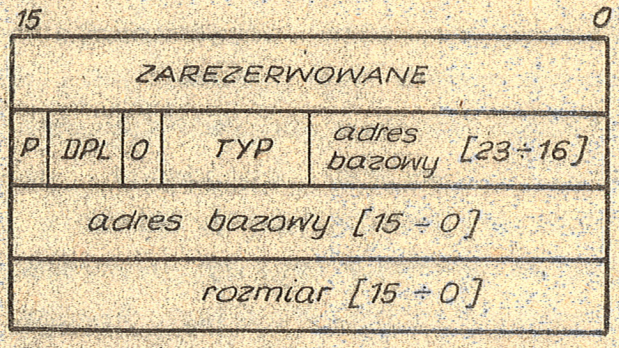
Deskryptory segmentów specjalnych definiują segmenty lokalnych tablic deskryptorów (tablic zawierających deskryptory dostępne tylko dla jednego zadania użytkowego) oraz segmenty stanu zadań (z

a)



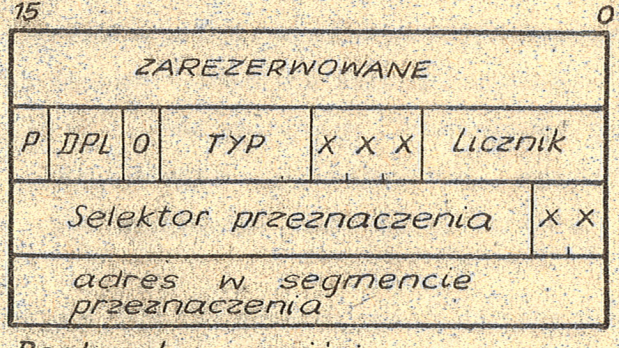
Deskryptor segmentu użytkowego

b)



Deskryptor tablic i segmentów systemowych

c)



Deskryptor przejścia

Rys. 6. Deskryptory w systemie 80286

każdym zadaniem skojarzony jest segment stanu zadania, zawierający informacje niezbędne dla poprawnego uaktywniania zadania). Format deskryptora segmentów systemowych przedstawia rys. 6b. Jak widać jest on podobny do formatu deskryptora segmentu użytkowego. Różnice polegają na innej wartości bitu S oraz innym znaczeniu pola TYP. Polę to określa typ segmentu (1-dostępny segment stanu zadania, 2-lokalna tablica deskryptorów, 3-zajęty segment stanu zadania).

Deskryptory przekazywania sterowania opisują cztery typy przejść: wywołania podprogramu (call), przełączenie zadania (task), przerwanie (interrupt), pułapka (trap). Odpowiednie przejście jest inicjowane pojedynczą instrukcją (lub zdarzeniem) zmieniającą zawartość rejestru segmentowego programu. Przejście typu wywołanie podprogramu jest używane przy odwołaniu do segmentu wymagającego wyższego poziomu uprzywilejowania zadania. Przejście typu przełączenie zadania stosowane jest przy operacji przełączania zadań. Przejścia typu przerwanie i pułapka określają natomiast wejścia do procedur obsługi przerwań. Odwołanie do nich powoduje zablokowanie systemu przerwań. Format deskryptora przejść przedstawia rys. 6c. Jak wiadać, deskryptor ten różni się znacznie od pozotałych. Pole TYP określa tu typ przejścia. Deskryptory przejść typu wywołanie podprogramu, przerwanie i pułapka - zawierają: selektor wskazujący deskryptor adresowanego segmentu oraz adres w segmencie. Rozwiązanie to w pełni zabezpiecza przed odwołaniem do niedozwolonych adresów segmentu programu. Przy przejściach typu wywołanie podprogramu, 5-bitowe pole licznika słów określa liczbę parametrów, które mają być automatycznie przepisane ze stosu programu wywołującego na stos procedury wywoływanej.

Deskryptor przejścia typu przełączenie zadania może odwoływać się jedynie do deskryptora segmentu stanu zadania. Użycie przejścia tego typu powoduje wykonanie przez procesor operacji przełączenia zadań.

Podkreślmy, że w każdym przypadku, gdy deskryptor przejścia wskazuje deskryptor niewłaściwego typu, generowane jest przerwanie wewnętrzne, uniemożliwiające pełne wykonanie zainicjowanej operacji.

Wszystkie opisane dotąd deskryptory przechowywane są w pamięci w postaci tablic. Rozróżnia się trzy typy tablic: globalną tablicę

deskryptorów GDT (Global Descriptor Table), lokalną tablicę deskryptorów LDT (Local Descriptor Table) i tablicę deskryptorów przerwań IDT (Interrupt Descriptor Table).

Globalna tablica deskryptorów zawiera deskryptory dostępne dla wszystkich zadań wykonywanych w systemie. W tablicy tej mogą znajdować się wszystkie typy deskryptorów z wyjątkiem deskryptorów przejść typu przerwanie i pułapka. Tablica ta jest inicjowana instrukcją LGDT, wpisującą deskryptor tej tablicy do rejestru deskryptorowego GDTD. Instrukcja ta może być wykonana tylko przez zadanie o najwyższym poziomie uprzywilejowania (system operacyjny).

Każde zadanie dysponuje prywatną lokalną tablicą deskryptorów, co zapewnia pełną separację zadań. Lokalna tablica deskryptorów może zawierać jedynie deskryptory segmentów użytkowych oraz deskryptory przejść typu wywołania podprogramu i przełączenie zadania. Tablica ta jest inicjowana instrukcją LLDT, wpisującą selektor tej tablicy do rejestru LDTS. Selektor ten wskazuje pozycję w GDT deskryptora LDT. Instrukcja LLDT może być wykonana tylko przez zadanie o najwyższym poziomie uprzywilejowania.

Tablicę deskryptorów przerwań tworzą tylko deskryptory przejść typu przerwanie, pułapka lub przełączenie zadania. Tablica ta jest inicjowana w sposób podobny do GDT. Zauważmy, że możliwość dowolnego usytuowania tej tablicy jak i funkcje jej elementów istotnie różnią procesor 80286 od 8086.

5. KONTROLA DOSTĘPU DO PAMIĘCI OPERACYJNEJ W MIKROPROCESORZE 80286

W mikroprocesorze 80286 wyróżnia się cztery poziomy uprzywilejowania (priorytety) zadań. Priorytet zadania określają dwa najmłodsze bity selektora zadania. Selektor ten w trakcie wykonywania zadania znajduje się w rejestrze segmentowym programu (CS). Priorytet aktualnie wykonywanego zadania określa bieżący poziom uprzywilejowania CPL (Current Privilege Level). Zadania o najwyższym priorytecie mogą uzyskać dostęp do wszystkich segmentów danych określonych przez LDT oraz GDT. Mogą też korzystać z pełnego zestawu instrukcji procesora.

Poziom uprzywilejowania (priorytet) skojarzony jest również z każdym deskryptorem. Jak wspomniano, jest on określony przez pole DPL deskryptora. Poziom uprzywilejowania deskryptora określa

wymagany priorytet zadania, które może korzystać z segmentu opisanego danym deskryptorem, bez interwencji systemu operacyjnego.

Dostęp do danych można dodatkowo ograniczyć korzystając z selektorów segmentów danych. Efektywny poziom uprzywilejowania w odniesieniu do tych segmentów jest uzależniony bowiem nie tylko od priorytetu zadania lecz również od poziomu uprzywilejowania danego selektora (RPL- Requested Privilege Level). Wynikowy poziom uprzywilejowania określa mniejszy z tych dwóch poziomów.

Przedstawiony powyżej system priorytetów stanowi tylko jeden z elementów mechanizmu ochrony dostępu do pamięci. Ogólnie, mechanizm ten obejmuje:

- ograniczenie zestawu dopuszczalnych operacji na segmencie (np: brak zezwolenia na zapis do segmentu danych, brak zezwolenia na odczyt z segmentu kodu),
- ograniczenie dopuszczalnych wartości adresu w segmencie do wartości odpowiadających rozmiarowi segmentu,
- separacja przestrzeni adresowych zadań przez wyłączność dostępu do lokalnych tablic deskryptorów,
- ograniczenie wyboru wejść do segmentów programu przez mechanizm przekazywania sterowań (deskryptory przejść),
- wprowadzenie systemu priorytetów zadań, deskryptorów i selektorów, który ogranicza niekontrolowany dostęp zadań użytkowych do segmentów systemowych,
- uzależnienie zestawu dostępnych instrukcji od poziomu uprzywilejowania zadania (instrukcje LIDT, LLDT, LGDT, LTR, LHSW, CTS i HLT mogą być wykonywane tylko przez zadania o najwyższym poziomie uprzywilejowania),
- uzależnienie możliwości wykonywania instrukcji wejścia/wyjścia od poziomu uprzywilejowania zadania w stosunku do poziomu uprzywilejowania wejścia/wyjścia IOPL (instrukcje INS, IN, OUTS, OUT, STI, CLI i LOCK mogą być wykonywane tylko przez zadania, których poziom uprzywilejowania jest większy lub równy poziomowi uprzywilejowania IOPL, określonemu przez 14 i 13 bit rejestru znaczników F).

Należy podkreślić, że kontrola dostępu realizowana jest sprzętowo, a naruszenie jakichkolwiek warunków dostępów powoduje wystąpienie przerwania i tym samym, przekazanie

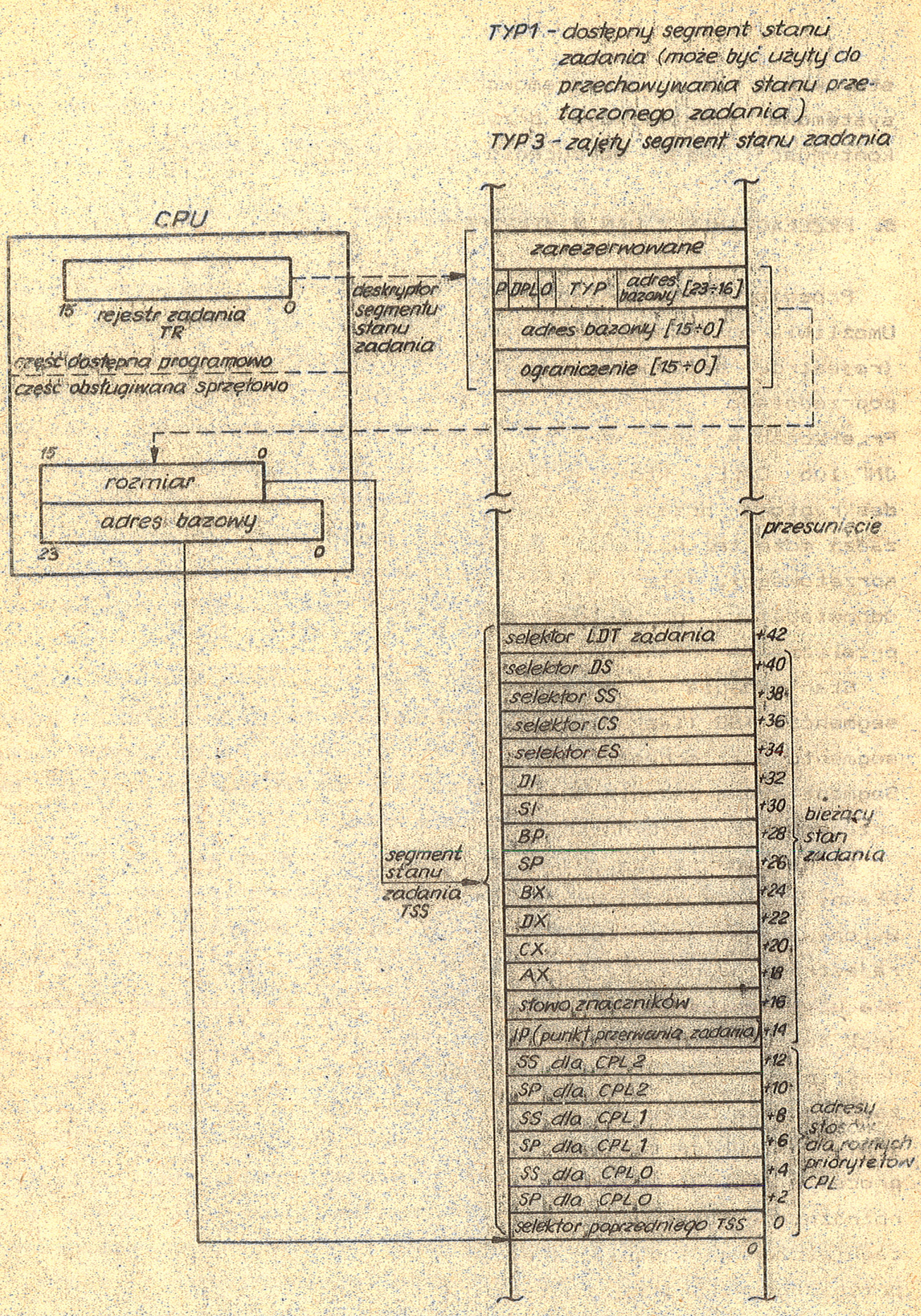
sterowania do oprogramowania systemowego. Oprogramowanie systemowe identyfikuje przyczynę przerwania i decyduje o kontynuacji bądź odrzuceniu zawieszonego zadania.

6. PRZEŁĄCZANIE ZADAŃ W MIKROPROCESORZE 80286

Procesor 80286 posiada wbudowany mechanizm przełączania zadań. Umożliwia on automatyczne zapamiętanie stanu zawieszonego zadania (rejestrów procesora, przestrzeni adresowej oraz łącznika do poprzedniego zadania) i uaktywnienie następnego zadania. Przełączanie zadań może być wywołane międzysegmentową instrukcją JMP lub CALL, która odwołuje się do segmentu stanu lub do deskryptora przejścia typu przełączenie zadania. Przełączanie zadań może też nastąpić w wyniku przerwania (programowego lub sprzętowego), jeżeli w tablicy deskryptorów przerwań deskryptor odpowiadający przyjętemu przerwaniu będzie deskryptorem typu przełączenie zadania.

Stan zadania w momencie przełączenia przechowywany jest w segmencie TSS (Task State Segment) stanu zadania. Format takiego segmentu oraz schemat odwołania się do niego przedstawia rys.7. Segment stanu zadania musi być większy od 002BH, aby możliwe było przechowanie niezbędnych elementów stanu.

Każde zadanie znajdujące się w systemie musi mieć do dyspozycji własny segment stanu zadania. Segment stanu zadania aktualnie wykonywanego jest identyfikowany przez selektor pamiętany w rejestrze zadania TR. Z rejestrem tym skojarzony jest niewidoczny dla użytkownika rejestr deskryptorowy TD, do którego wczytywany jest automatycznie przy każdej zmianie zawartości rejestru TR deskryptor segmentu stanu zadania. Powrót do zadania, które zainicjowało ostatnią operację przełączenia realizuje instrukcja IRET. Instrukcja ta jest jednak również stosowana w celu powrotu z procedur obsługi przerwań. Obie te sytuacje muszą być oczywiście rozróżnione. Dlatego też bit 14 w słowie znaczników, zwany bitem zagnieżdżonego zadania (nested task - NT), jest ustawiany w przypadku wykonania operacji przełączania zadań i służy do sterowania wykonaniem instrukcji IRET. Jeżeli NT=0, to instrukcja IRET jest interpretowana jako typowy powrót z procedury przerwania. W przeciwnym razie, instrukcja IRET inicjuje przełączenie zadań. I w tym celu wykorzystywane jest słowo o



Rys. 7. Mechanizm przełączania zadań

adresie 0 w segmencie stanu zadania. Należy zaznaczyć, że podczas przełączania segmenty stanu przełączanych zadań są deklarowane jako zajęte, a wartość selektora zadania zawieszonego jest zapisywana do segmentu stanu zadania uaktywnionego. Podczas powrotnego przełączania zadań bit NT jest zerowany. Przerwanie, które nie powoduje przełączenia zadań zeruje oczywiście bit NT, aby uniknąć błędnej interpretacji instrukcji IRET.

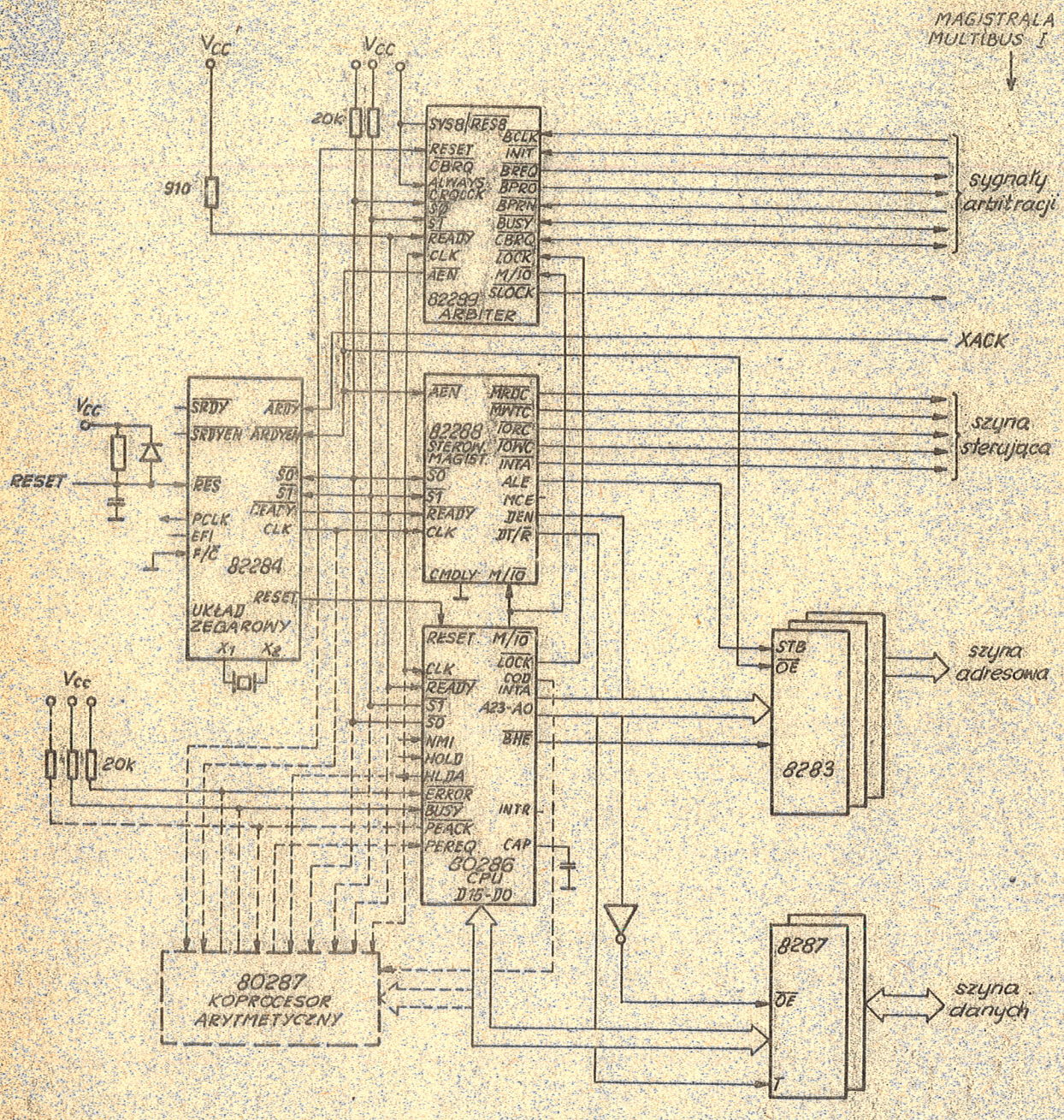
7. RODZINA UKŁADÓW 80286

Podobnie jak miało to miejsce w systemie 8086, mikroprocesorowi centralnemu towarzyszy cały szereg układów pomocniczych, które umożliwiają tworzenie mikrokomputerów o różnej strukturze i własnościach funkcjonalnych. Do podstawowych układów pomocniczych należą: generator sygnałów zegarowych 82284, koprocesor arytmetyczny 80287, sterownik magistrali 82288, arbiter magistrali 82289, nadajniki/odbiorniki danych 8286/8287 i rejestry buforowe szyny adresowej 8282/8283. Układ 82284 generuje sygnał zegara systemowego oraz synchronizuje sygnały READY/ oraz RESET/. Układ koprocesora arytmetycznego 80287 stanowi zmiennoprzecinkową jednostkę arytmetyczno-logiczną, charakteryzującą się bardzo dużą szybkością i precyzją obliczeń. Układ 82288 dokonuje konwersji lini statusu mikroprocesora na sygnały szyny sterującej magistali Multibus. Układ 82289 wypracowuje natomiast sygnały arbitracji tej magistrali.

Jak łatwo zauważyć rodzina układów 80286 stanowi funkcjonalny odpowiednik rodziny 8086. Można zatem oczekiwać w pełni analogicznych struktur systemowych. Potwierdzeniem tego przypuszczenia jest niewątpliwie struktura systemu zbudowanego z użyciem układów rodziny 80286, a przypominająca strukturę systemu 8086 pracującego w trybie maksymalnym.
Przykładową strukturę systemu zbudowanego z użyciem układów rodziny 80286 przedstawia rys.8.

8. PODSUMOWANIE

Nie ulega wątpliwości, że mikroprocesor 80286 jest ważnym i ciekawym rozwinięciem rodziny 8086. Zaawansowane mechanizmy ochrony i wirtualizacji pamięci, systemowe wspomaganie procesu



Rys 8. Struktura systemu 80286 z magistralą systemową Multibus I

uzasadniają zastosowanie tego mikroprocesora w różnych wielozadaniowych i wielodostępnych systemach mikrokomputerowych. Spośród bardziej popularnych systemów tego typu zbudowanych z użyciem mikroprocesora 80286 należy wymienić: System 310 i System 380 firmy Intel, mikrokomputery IBM PC/AT i wprowadzane obecnie IBM PS/2 (Model 50 i 60), graficzne stacje robocze M-28 firmy Olivetti, itd. Zestawienie powyższe stanowi praktyczne potwierdzenie atrakcyjności mikroprocesora 80286.

BIBLIOGRAFIA

1. Edmund Strauss: Inside the 80286, Brandy Communications Company, 1986.

2. iAPX 286 Hardware Reference Manual, Technical Report 210760, Intel Corporation, Santa Clara, California, 1983.

3. iAPX 286 Programmer Reference Manual, Technical Report 210498, Intel Corporation, Santa Clara, California, 1983.

4. Microsystem Component Handbook, vol.I, Intel Corporation, Santa Clara, California, 1984.

5. iAPX 286 Operating Systems Writer's Guide, Intel Corporation, Santa Clara, California, 1983.

IV Szkoła CSI/PTI
Mikrokomputery 16 i 32-bitowe
Łódź, grudzień 1987

Mikroprocesor 32-bitowy Intel 80386

mgr inż. Paweł Krysztofiak
Instytut Automatyki
Politechnika Poznańska
ul. Piotrowo 3A
60-965 Poznań, tel.78-25-29

1. WSTĘP

W czasie kilkunastoletniej historii rozwoju mikrokomputerów obserwuje się stałe rozszerzanie zakresu ich zastosowań. Od zastosowań związanych z wykonywaniem prostych obliczeń i sterowaniem bezpośrednim układami automatyki, udało się przejść do zastosowań zarezerwowanych dotychczas dla minikomputerów. Tak gwałtowne rozszerzenie zakresu zastosowań mikrokomputerów zostało spowodowane głównie wzrostem mocy obliczeniowej mikroprocesorów i zastosowaniem w nich nowych rozwiązań architektonicznych. W tym drugim przypadku chodzi o wprowadzenie mechanizmów sprzętowych niezbędnych w bardziej zaawansowanych zastosowaniach. Wielodostępne i wielozadaniowe systemy komputerowe wymagają na przykład sprzętowych mechanizmów ochrony pamięci dla zagwarantowania nienaruszalności lokalnych (prywatnych) obszarów pamięci współbieżnie wykonywanych zadań. W systemach tego typu bardzo pożądane są również mechanizmy wirtualizacji pamięci i wspomaganego sprzętowo przełączania zadań.

Pierwszym mikroprocesorem, który spełniał wszystkie wymienione wymagania systemów wielodostępnych był 16-bitowy mikroprocesor Intel 80286. Jego istotnym rozwinięciem jest 32-bitowy mikroprocesor Intel 80386, którego prezentacji poświęcony jest niniejszy artykuł.

2. CHARAKTERYSTYKA MIKROPROCESORA 80386

2.1. Dane podstawowe

Mikroprocesor 80386 posiada 32-bitowe rejestry, oraz 32-bitowe szyny: adresową i danych. Fizyczna przestrzeń adresowa wynosi 4 GB, natomiast przestrzeń wirtualna dla każdego zadania 64 TB. Mikroprocesor może być sterowany zegarem o częstotliwości 12,5 lub 16 MHz. W tym drugim przypadku przepustowość szyny danych wynosi 32 MB na sekundę. Mikroprocesor zawiera równolegle pracujące jednostki operacyjne, pozwalające na zrównoleglenie operacji pobierania, dekodowania i wykonywania rozkazów. Dzięki temu, między innymi, mikroprocesor może wykonywać 3 do 4 miliony operacji na sekundę. Zintegrowana z procesorem jednostka zarządzania pamięcią operacyjną umożliwia realizację pamięci wirtualnej segmentowanej i stronicowanej, ochronę obszarów pamięci oraz efektywną realizację wielozadaniowości. W zastosowaniach obliczeniowych efektywność pracy systemu może być istotnie zwiększona w wyniku wykorzystania szybkiego koprocesora arytmetycznego 80287 lub 80387. W stosunku do 80286, mikroprocesor 80386 posiada dodatkowo mechanizmy autotestów (wykonywanych po włączeniu napięcia zasilającego) oraz mechanizmy wspomagające uruchamianie programów. Omawiany mikroprocesor zachowuje zgodność na poziomie programów wynikowych z wszystkimi mikroprocesorami z rodziny iAPX 86.

2.2. Tryby pracy mikroprocesora 80386

Pogodzenie w mikroprocesorze 80386 nowych mechanizmów zarządzania pamięcią operacyjną oraz zgodności programowej ze starszymi przedstawicielami rodziny mikroprocesorów iAPX 86, stało się możliwe przez wprowadzenie dwóch trybów pracy: trybu

adresowania rzeczywistego (*Real Address Mode*) i trybu pracy z ochroną i wirtualizacją pamięci (*Protected Virtual Address Mode*).

Tryb pracy z adresowaniem rzeczywistym przyjmowany jest każdorazowo po włączeniu lub wyzerowaniu mikroprocesora. W tym trybie mikroprocesor 80386 posiada taką samą strukturę logiczną jak mikroprocesory 8086 i 80186 oraz 80286 pracujący w trybie z adresowaniem rzeczywistym. Takie same są też mechanizmy adresowania pamięci, przestrzeń adresowa pamięci operacyjnej oraz system przerwań.

Pełne możliwości mikroprocesora 80386 można uzyskać po przejściu do trybu pracy z protekcją i wirtualizacją pamięci. W tym trybie zmieniają się mechanizmy adresowania, a fizyczna przestrzeń adresowa wzrasta z 1 MB do 4 MB. Każdy użytkownik (zadanie) dysponuje przestrzenią wirtualną o rozmiarze 64 TB. Dostępne stają się dodatkowe mechanizmy zapewniające bardzo efektywne przełączanie zadań, co ma istotne znaczenie w systemach wielozadaniowych. W tym trybie mogą być również wykonywane wszystkie programy napisane dla mikroprocesora 8086.

2.3. Zarządzanie pamięcią operacyjną i zadaniami

Podstawowym mechanizmem zarządzania pamięcią operacyjną jest segmentacja. Segment jest spójnym obszarem pamięci wirtualnej zawierającym logicznie powiązane z sobą informacje. Na przykład: podprogramy, tablice, obszary danych i inne programowe struktury logiczne mogą tworzyć odrębne segmenty. Informacje o danym segmencie pamiętane są w 8 bajtowej strukturze danych, zwanej deskryptorem (*descriptor*). Wszystkie deskryptory występujące w systemie grupowane są w tablice deskryptorów.

Segmentacja umożliwia realizację złożonych mechanizmów ochrony pamięci. Dla uzyskania tego celu skojarzono z każdym segmentem bajt określający warunki dostępu poszczególnych zadań do danego segmentu.

W mikroprocesorze 80386 oprócz segmentacji, został zrealizowany mechanizm stronicowania pamięci. Strona jest spójnym obszarem pamięci wirtualnej o stałej długości. W ogólności nie wszystkie strony aktywnego segmentu muszą w danej chwili rezydować w pamięci operacyjnej. Podobnie jak w przypadku segmentów, również strony mogą posiadać określone warunki

dostępu, pozwalające na realizację ochrony pamięci.

W celu efektywnej realizacji pracy wieloprogramowej, niezbędne są mechanizmy szybkiego przełączania zadań wykonywanych współbieżnie przez procesor. Sprzętowe mechanizmy przełączania zadań umożliwiają składowanie wszystkich elementów stanu zadania (stany rejestrów procesora, przestrzeń adresowa, wskaźnik do poprzednio wykonywanego zadania, itd) oraz wczytywanie nowych wartości odpowiadających uaktywnianemu zadaniu.

3. STRUKTURA MIKROPROCESORA 80386

3.1. Struktura funkcjonalna mikroprocesora

W mikroprocesorze można wyróżnić cztery jednostki operacyjne, pracujące równolegle: jednostkę wykonawczą (*execution unit*), jednostkę rozkazową (*instruction unit*), jednostkę zarządzania pamięcią (*memory management unit*) oraz jednostkę magistralową (*bus interface unit*). Taka struktura mikroprocesora pozwala na równoczesne pobieranie, dekodowanie i wykonywanie rozkazów. Zmniejsza to istotnie średni czas wykonania pojedynczego rozkazu oraz w pewnym stopniu uniezależnia procesor od dostępu do magistrali. Struktura funkcjonalna mikroprocesora jest przedstawiona na rysunku 3.1.

Jednostka wykonawcza zawiera osiem 32 bitowych rejestrów ogólnego przeznaczenia oraz 64 bitowy rejestr przesuwny, służący do realizacji szybkich operacji arytmetycznych.

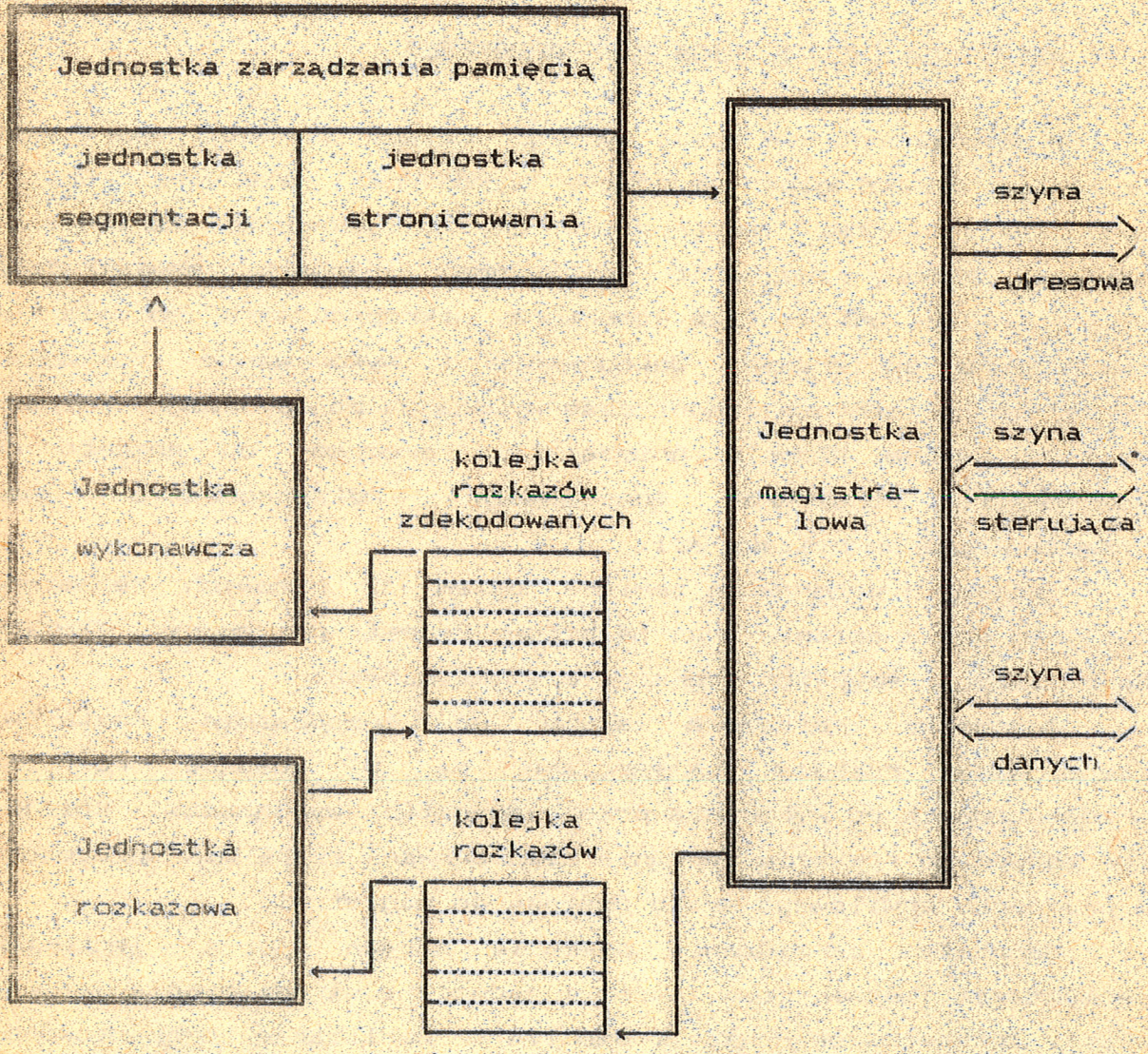
Jednostka rozkazowa służy do dekodowania rozkazów. Zdekodowane rozkazy umieszczane są w kolejce rozkazów przeznaczonych do wykonania przez jednostkę wykonawczą. Kolejka zdekodowanych rozkazów zapewnia nieprzerwaną pracę mikroprocesora w przypadku chwilowego braku dostępu do magistrali.

Jednostka zarządzania pamięcią składa się z jednostki segmentacji (*segmentation unit*) i jednostki stronicowania (*paging unit*). Jednostka segmentacji umożliwia realizację segmentowanej pamięci wirtualnej. Każdy segment, o maksymalnym rozmiarze 4 GB, posiada atrybuty informujące o jego położeniu, rozmiarze, typie oraz warunkach dostępu. Każde zadanie może posiadać do 16381 segmentów, a więc użytkownik dysponuje pamięcią wirtualną o

rozmiarze do 64 TB.

Jednostka stronicowania pozwala na dzielenie segmentu na strony o rozmiarze 4 KB. Należy wyjaśnić, że mechanizm stronicowania pamięci operacyjnej jest niewidoczny dla procesu segmentacji.

Jednostka magistralowa zapewnia pobieranie i kolejkowanie rozkazów, dynamiczne sterowanie szerokością szyny danych oraz wybór dpwolnego bajtu szyny danych.



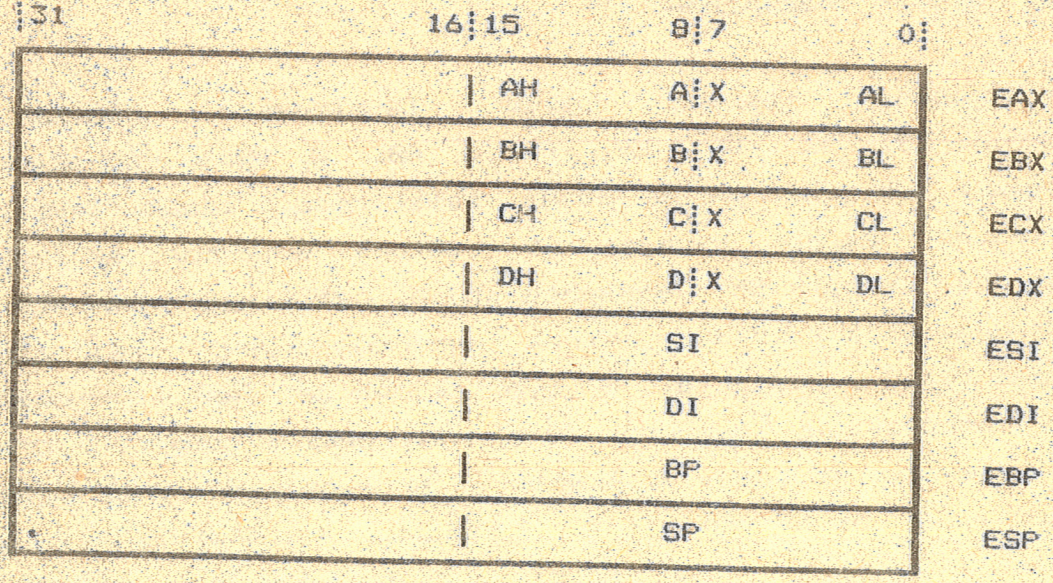
Rys. 3.1. Struktura funkcjonalna mikroprocesora 80386

3.2. Struktura logiczna mikroprocesora

Mikroprocesor 80386 posiada 32-rejestry, podzielone na następujące grupy:

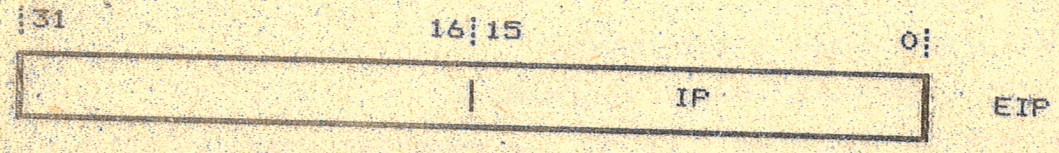
- rejestry ogólnego przeznaczenia,
- wskaźnik rozkazów i rejestr znaczników,
- rejestry segmentowe,
- rejestry sterujące,
- rejestry segmentów systemowych,
- rejestry uruchomieniowe i testowe.

W mikroprocesorze 80386 występuje osiem 32-bitowych rejestrów ogólnego przeznaczenia. Służą one do przechowywania wartości danych i adresu. Pozwalają na wykonywanie operacji na danych o długości 1, 8, 16, 32 i 64 bitów oraz na polach danych o długości od 1 do 32 bitów. Rejestry ogólnego przeznaczenia oznaczono: EAX, EBX, ECX, EDX, ESI, EDI, EBP i ESP. Młodszych 16 bitów każdego z tych rejestrów może być wykorzystywane niezależnie, przez podanie nazwy AX, BX, CX, DX, SI, DI, BP lub SP. Dzięki temu, istnieje pełna zgodność programowa między mikroprocesorem 80386, a 80286 i 8086. Podobnie jak w tych ostatnich mikroprocesorach, cztery pierwsze z wymienionych rejestrów 16-bitowych, mogą być wykorzystywane jako pary rejestrów 8-bitowych. Rejestry ogólnego przeznaczenia są przedstawione na rysunku 3.2.



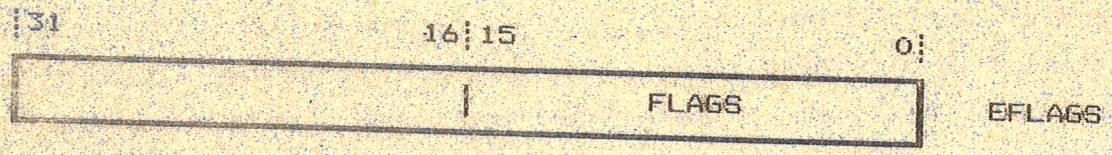
Rys. 3.2. Rejestry ogólnego przeznaczenia.

Wskaźnik rozkazów jest 32-bitowym rejestrem EIP. W tym rejestrze jest przechowywany adres rozkazu, który ma być wykonany. Jest to adres relatywny w stosunku do adresu segmentu. Młodszych 16 bitów rejestru EIP zawiera licznik rozkazów wykorzystywany przy adresowaniu 16-bitowym. Wskaźnik rozkazów jest przedstawiony na rysunku 3.3.



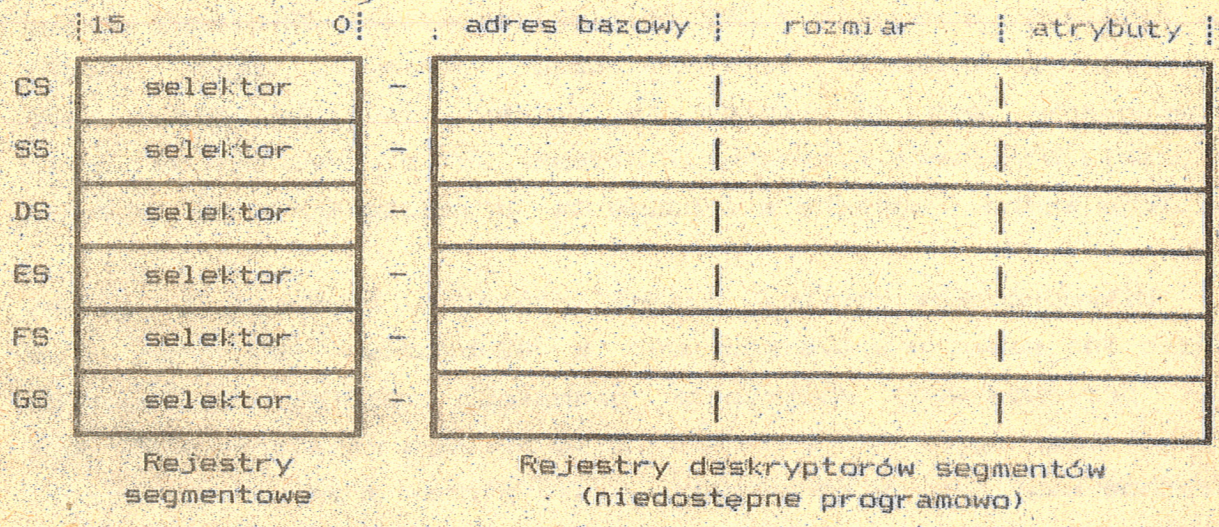
Rys. 3.3. Wskaźnik rozkazów.

Rejestr znaczników EFLAGS jest rejestrem 32-bitowym. Młodszych 16 bitów tego rejestru jest zgodne z rejestrem znaczników w mikroprocesorach 80286 i 8086. Z pozostałych 16 bitów wykorzystane są tylko dwa, o nazwach *Virtual Mode* i *Resume Flag*. Rejestr znaczników jest przedstawiony na rysunku 3.4.



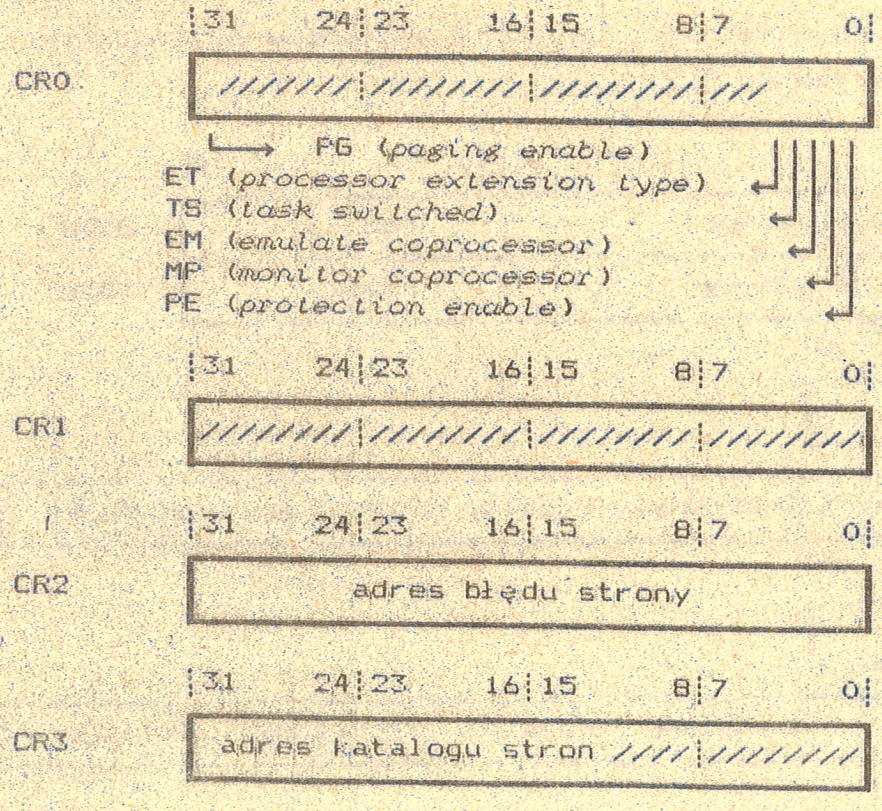
Rys. 3.4. Rejestr znaczników.

Mikroprocesor posiada sześć 16-bitowych rejestrów segmentowych: CS, SS, DS, ES, FS i GS. W tych rejestrach przechowywana jest informacja zwana selektorem, identyfikująca bieżący segment pamięci. Rejestr CS zawiera selektor segmentu kodu programu, rejestr SS - segmentu stosu, natomiast pozostałe - segmentów danych. Z każdym rejestrem segmentowym jest skojarzony rejestr deskryptora segmentu. W tym rejestrze jest pamiętany 32-bitowy adres bazowy segmentu, 32-bitowy rozmiar segmentu oraz atrybuty segmentu. Rejestry deskryptorów segmentów są niedostępne programowo. Ich zawartość jest ładowana automatycznie po programowej zmianie zawartości odpowiedniego rejestru segmentowego. Rejestry segmentowe oraz skojarzone z nimi rejestry deskryptorów segmentów są przedstawione na rysunku 3.5.



Rys. 3.5. Rejestry segmentowe.

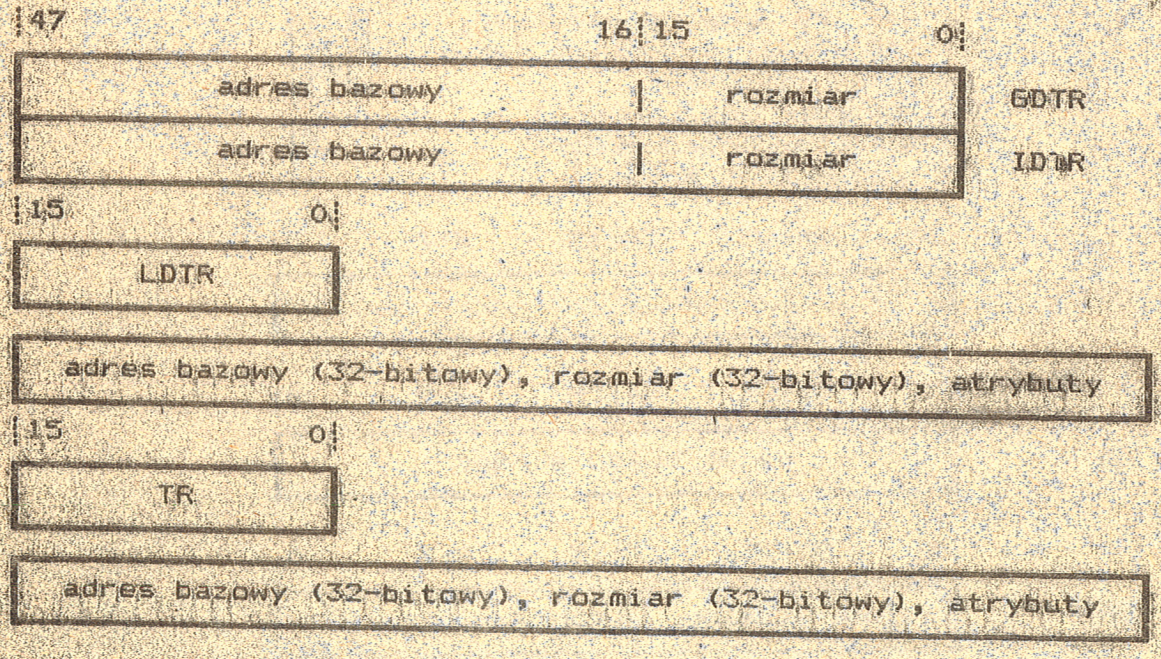
Mikroprocesor posiada cztery 32-bitowe rejestry sterujące: CR0, CR1, CR2 i CR3. W tych rejestrach są przechowywane informacje dotyczące stanu procesora. W rejestrze CR0 jest wykorzystanych sześć bitów dotyczących trybu pracy procesora oraz współpracy z koprocesorem arytmetycznym. Rejestr CR1 nie jest



Rys. 3.6. Rejestry sterujące.

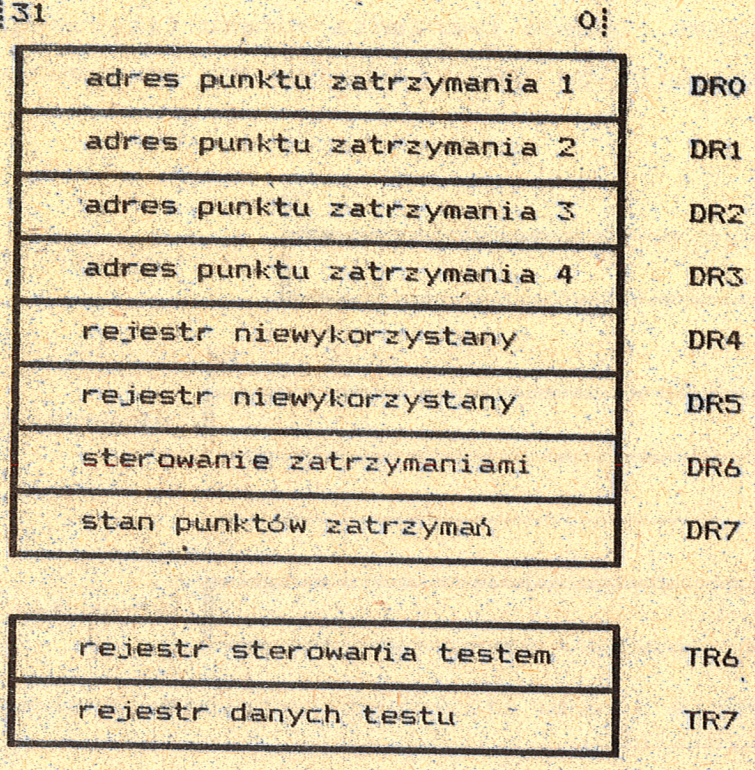
wykorzystany. W rejestrze CR2 jest przechowywany 32-bitowy adres, przy którym wystąpił ostatnio błąd strony. (Błąd strony spowodowany jest odwołaniem do strony nie znajdującej się aktualnie w pamięci fizycznej.) Rejestr CR3 zawiera bazowy adres katalogu stron. Rejestry sterujące są przedstawione na rysunku 3.6.

Mikroprocesor posiada cztery rejestry zawierające adresy tablic lub segmentów systemowych. W 48-bitowym rejestrze GDTR jest przechowywany deskryptor globalnej tablicy deskryptorów (*Global Descriptor Table - GDT*). W 48-bitowym rejestrze IDTR jest przechowywany deskryptor tablicy deskryptorów przerwań (*Interrupt Descriptor Table - IDT*). W 16-bitowym rejestrze LDTR jest przechowywany selektor lokalnej tablicy deskryptorów (*Local Descriptor Table -LDT*). W 16-bitowym rejestrze TR jest przechowywany selektor segmentu stanu aktualnie wykonywanego zadania (*Task State Segment-TSS*). Z rejestrami LDTR i TR są skojarzone rejestry deskryptorów segmentów. W tych rejestrach są pamiętane 32-bitowe adresy bazowe segmentów, 32-bitowe rozmiary segmentów oraz atrybuty segmentów. Rejestry deskryptorów segmentów są niedostępne programowo. Ich zawartość jest ładowana automatycznie po programowej zmianie zawartości odpowiedniego rejestru segmentu systemowego. Rejestry segmentów systemowych są przedstawione na rysunku 3.7.



Rys. 3.7. Rejestry segmentów systemowych.

Mikroprocesor posiada osiem 32-bitowych rejestrów uruchomieniowych. Cztery z nich, o nazwach DR0 do DR3, zawierają adresy punktów zatrzymań pracy programu. Rejestry DR4 i DR5 są niewykorzystane. Rejestr DR6 jest wykorzystywany do ustawiania punktów zatrzymań, natomiast rejestr DR7 wskazuje aktualny stan rejestrów uruchomieniowych. Dwa 32-bitowe rejestry testowe pozwalają na sterowanie testowaniem części pamięci operacyjnej wykorzystywanej przez system operacyjny. Rejestry uruchomieniowe i testowe są przedstawione na rysunku 3.8.



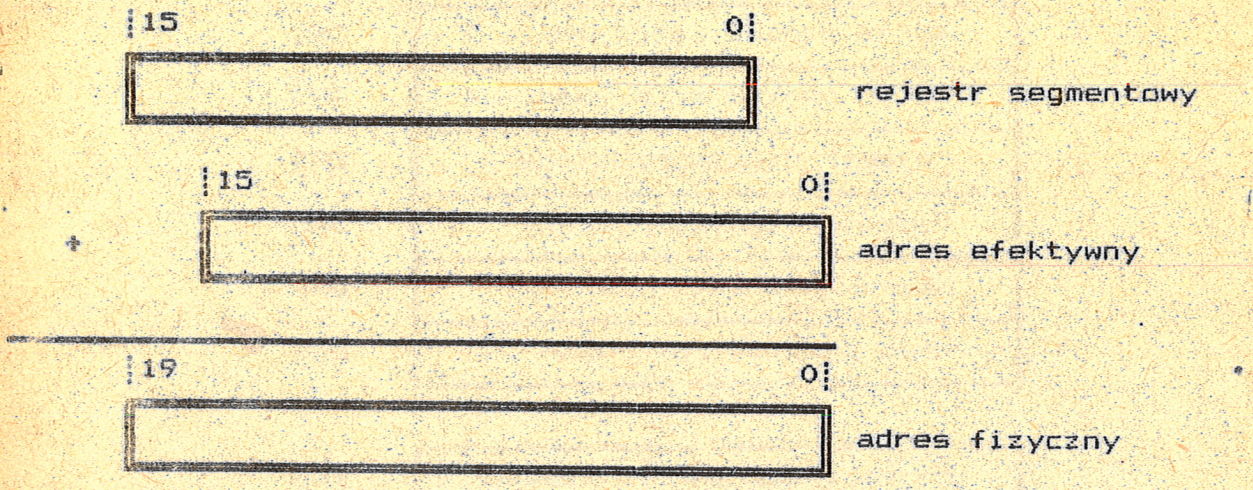
Rys. 3.8. Rejestry uruchomieniowe i testowe.

4. MECHANIZMY ZARZĄDZANIA PAMIĘCIĄ W MIKROPROCESORZE 80386

Zarządzanie pamięcią operacyjną w mikroprocesorze 80386 uzależnione jest od wybranego przez użytkownika trybu pracy. Pełne możliwości mikroprocesora wykorzystuje się w trybie pracy z ochroną i wirtualizacją pamięci operacyjnej.

4.1. Zarządzanie pamięcią w trybie adresowania rzeczywistego

W trybie adresowania rzeczywistego rozmiar pamięci operacyjnej jest ograniczony do 1 MB. W tym trybie stronicowanie pamięci nie jest wykorzystywane. Fizyczny adres pamięci jest tworzony przez dodanie zawartości odpowiedniego rejestru segmentowego, przesuniętego o cztery bity w lewo, do adresu efektywnego. Rejestr segmentowy oraz adres efektywny są 16 bitowe, a więc adres fizyczny jest 20 bitowy. W trybie adresowania rzeczywistego wszystkie segmenty fizyczne posiadają równą wielkość, wynoszącą 64 KB. Segmenty te nie muszą być w ogólności rozłączne. Sposób tworzenia adresu fizycznego w trybie adresowania rzeczywistego jest przedstawiony na rysunku 4.1.

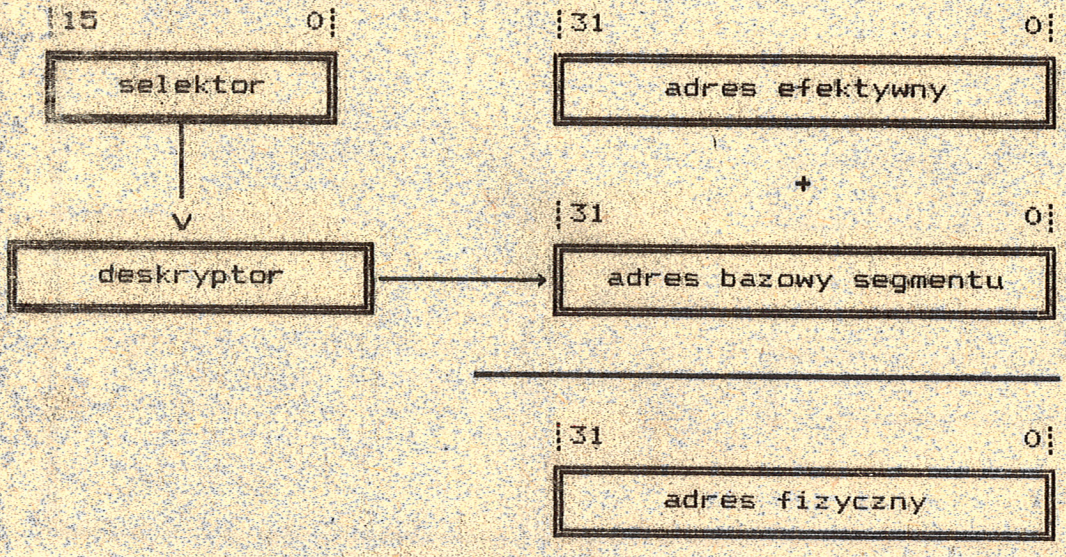


Rys. 4.1. Tworzenie adresu fizycznego w trybie adresowania rzeczywistego.

4.2 Zarządzanie pamięcią w trybie z protekcją i wirtualizacją

W trybie pracy z protekcją i wirtualizacją pamięci fizyczna przestrzeń adresowa wynosi 4 GB, natomiast przestrzeń wirtualna każdego zadania - 64 TB. Pojedyncze zadanie może więc wykorzystywać przestrzeń adresową wynoszącą 64 TB. W tym trybie pracy fizyczny adres jest tworzony na podstawie 16-bitowego selektora i 32-bitowego adresu efektywnego. Jednakże wartość selektora nie jest bezpośrednio sumowana z adresem efektywnym. Selektor służy do zidentyfikowania, umieszczonego w odpowiedniej tablicy, deskryptora adresowanego segmentu. Deskryptor ten

zawiera między innymi 32-bitowy adres bazowy segmentu. Adres logiczny jest tworzony przez zsumowanie 32-bitowego adresu bazowego segmentu i 32-bitowego przesunięcia. W przypadku pracy z wyłączonym mechanizmem stronicowania, adres logiczny jest równoważny adresowi fizycznemu. W przypadku pracy z wykorzystaniem stronicowania pamięci, adres logiczny jest dodatkowo zamieniany przez mechanizmy stronicowania w adres fizyczny. Sposób tworzenia adresu fizycznego z wykorzystaniem mechanizmów segmentacji pamięci jest przedstawiony na rysunku 4.2.

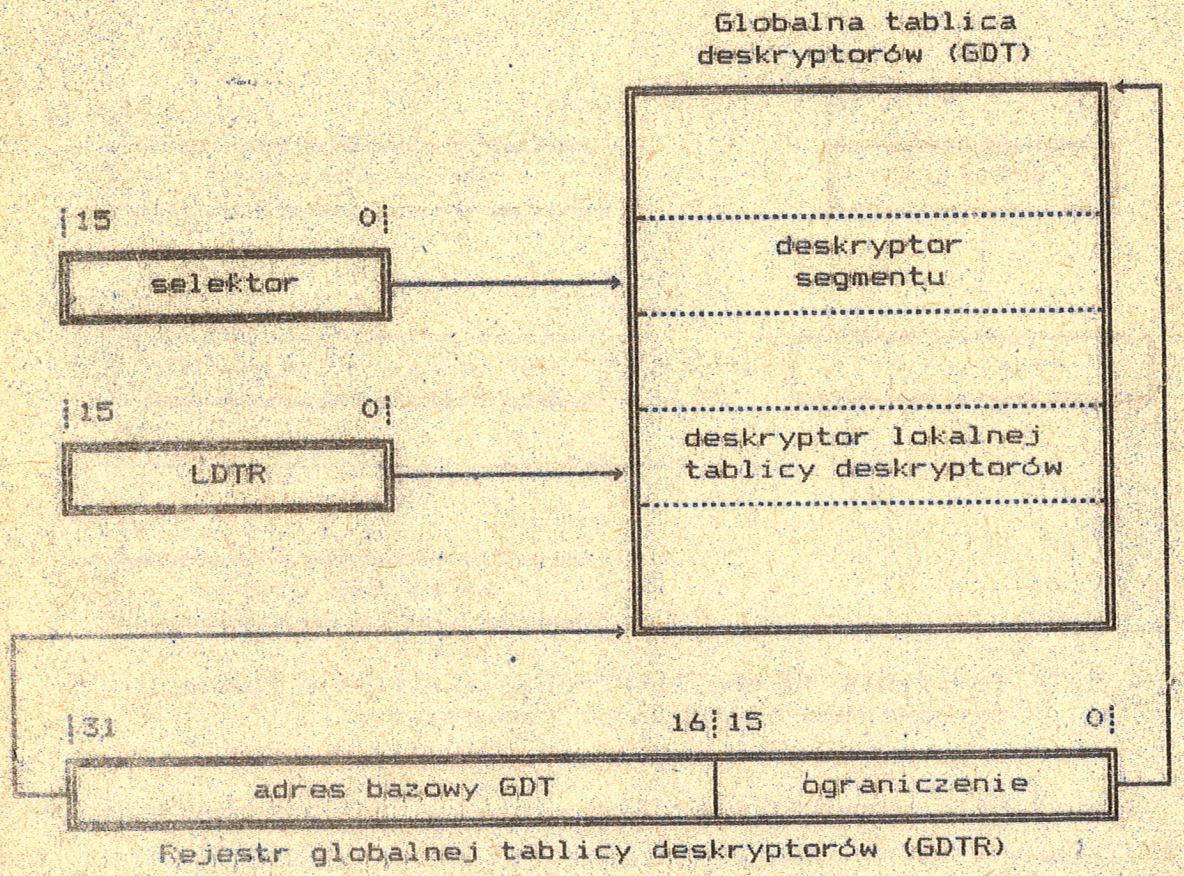


Rys. 4.2. Tworzenie adresu fizycznego z wykorzystaniem mechanizmów segmentacji pamięci.

Wszystkie segmenty wykorzystywane w mikroprocesorze 80386 zdefiniowane są w tablicach deskryptorów. Istnieją trzy typy takich tablic: tablica deskryptorów globalnych, tablica deskryptorów lokalnych i tablica deskryptorów przerwań. Rozmiar pojedynczej tablicy może wynosić od 8 bajtów do 64 KB, w zależności od liczby pamiętanych w danej tablicy deskryptorów. Indeksem do odpowiedniej tablicy deskryptorów jest 13 starszych bajtów selektora. Z każdą tablicą są skojarzone rejestry, które mogą być modyfikowane przez system operacyjny za pomocą specjalnych instrukcji.

Globalna tablica deskryptorów (*GDT*) zawiera deskryptory opisujące wszystkie zadania występujące w systemie, za wyjątkiem zadań obsługi przerwań. Każdy system z mikroprocesorem 80386

zawiera tablicę GDT opisującą segmenty programu i danych systemu operacyjnego, segmenty stanu zadań oraz deskryptory lokalnych tablic deskryptorów. Występujące w systemie selektory stanowią wskaźniki do odpowiednich deskryptorów w globalnej tablicy deskryptorów. Z tablicą GDT jest skojarzony 48 bitowy rejestr GDTR, zawierający 32-bitowy adres bazowy segmentu oraz 16-bitowy rozmiar segmentu. Globalna tablica deskryptorów jest przedstawiona na rysunku 4.3.



Rys. 4.3. Globalna tablica deskryptorów.

Lokalna tablica deskryptorów (*LDT*) zawiera deskryptory związane z danym zadaniem. Lokalna tablica deskryptorów pozwala na separację segmentów wykorzystywanych przez dane zadanie od segmentów wykorzystywanych przez pozostałe zadania oraz - od segmentów wspólnych dla wszystkich zadań. Żaden z segmentów nie może być wykorzystywany przez zadanie, jeżeli deskryptor tego segmentu nie znajduje się w lokalnej lub globalnej teblicy deskryptorów. Z tablicą LDT jest skojarzony 16-bitowy rejestr LDTR, zawierający selektor do deskryptora danej tablicy LDT w

tablicy GDT.

Tablica deskryptorów przerwań (IDT) zawiera deskryptory wszystkich procedur obsługi przerwań występujących w systemie. Każde przerwanie występujące w systemie musi posiadać oddzielne pole w tablicy IDT.

W systemie 80386 występują dwie klasy deskryptorów: deskryptory systemowe i deskryptory segmentów użytkowych. Wśród deskryptorów systemowych wyróżnia się: deskryptory lokalnych tablic deskryptorów, deskryptory segmentów stanu zadań i deskryptory przejść. Deskryptory segmentów użytkowych opisują segmenty programu i danych. Każdy deskryptor złożony jest z ośmiu bajtów, opisujących dany segment pamięci. Deskryptor zawiera 32-bitowy adres bazowy segmentu, 20-bitowy rozmiar segmentu (wyrażony w bajtach lub stronach) oraz 12-bitowe pole charakterystyczne segmentu. W tym ostatnim polu opisu segmentu umieszczonesą następujące informacje:

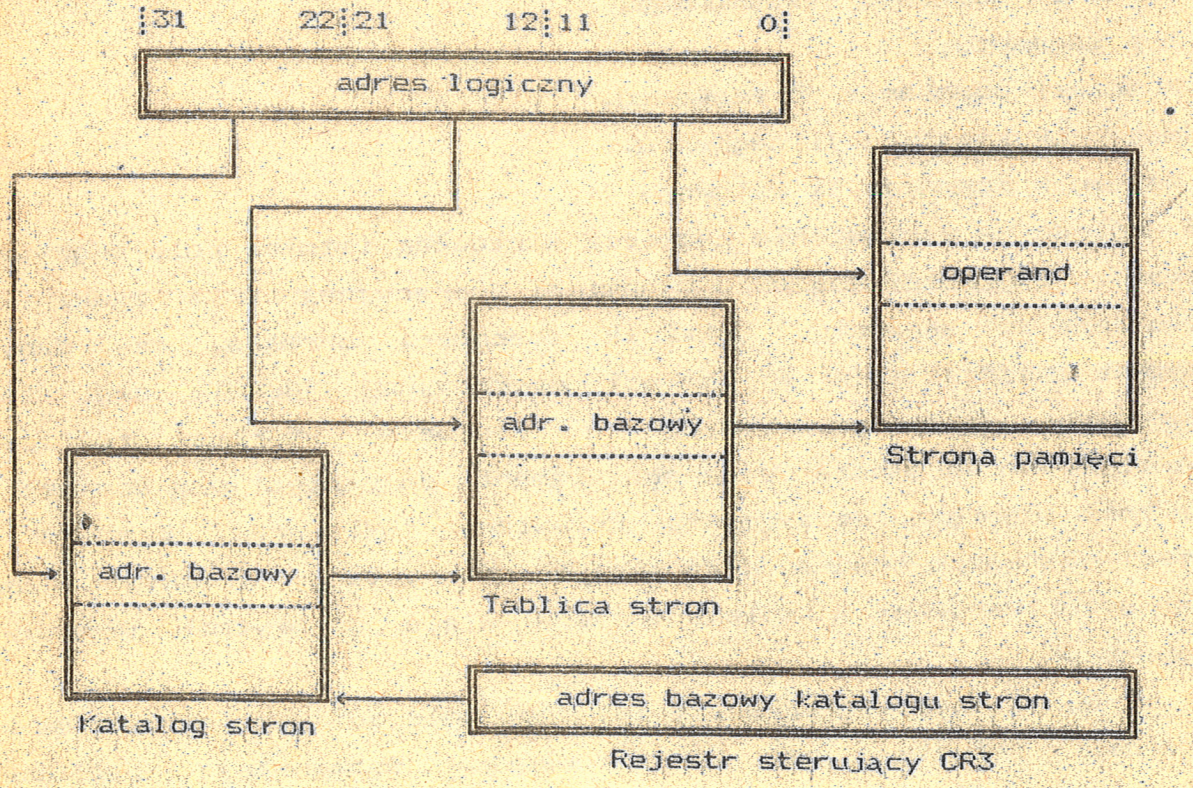
- wskaźnik obecności segmentu w pamięci operacyjnej,
- poziom uprzywilejowania segmentu,
- wyróżnik segmentu systemowego,
- typ segmentu,
- znacznik odwołania do segmentu,
- wskaźnik kwantyzacji segmentu,
- wskaźnik operacji 32 bitowej.

Ze względu na zarządzanie pamięcią wirtualną istotną rolę odgrywa wskaźnik obecności segmentu w pamięci operacyjnej oraz znacznik odwołania do segmentu. Wskaźnik obecności określa, czy dany segment znajduje się w pamięci operacyjnej. Jeżeli nie, to generowane jest przerwanie, w obsłudze którego zostaje zainicjowane sprowadzenie danego segmentu do pamięci operacyjnej. Znacznik odwołania do segmentu ustawiany jest przez procesor, jeżeli nastąpiło odwołanie do danego deskryptora. Znacznik ten jest wykorzystywany w implementacjach wielu algorytmów wymiany segmentów.

Drugim, obok segmentacji, istotnym mechanizmem zarządzania pamięcią wirtualną jest stronicowanie. Segmentacja zakłada podział programów i danych na różnej długości moduły związane z logiczną organizacją informacji. Stronicowanie wprowadza podział segmentów na wiele stron o niewielkim, stałym rozmiarze. W systemie 80386 rozmiar każdej strony wynosi 4 KB. Strony nie są

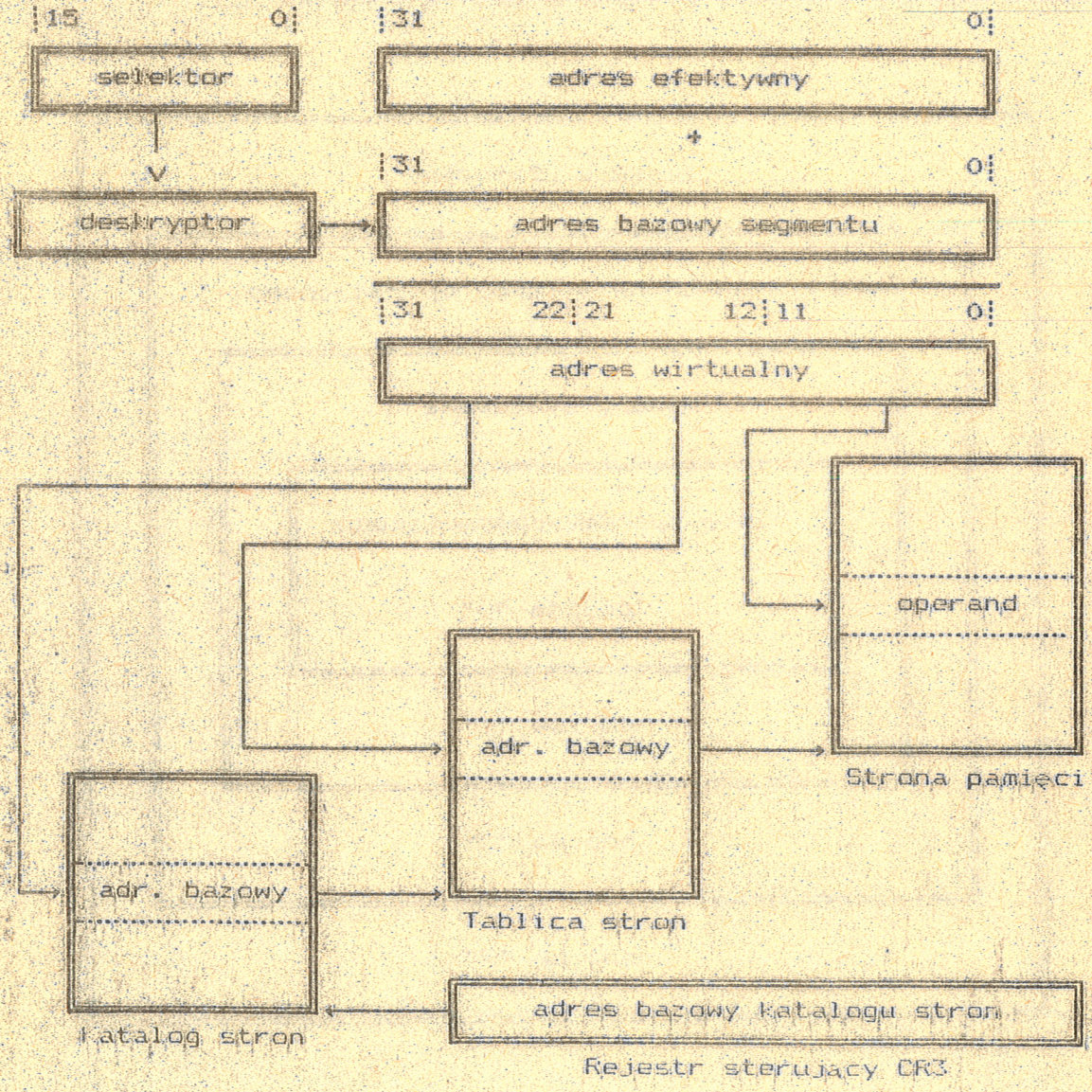
powiązane z logiczną organizacją informacji. Jednolita wielkość wszystkich stron znacznie upraszcza algorytmy alokacji pamięci oraz pozwala uniknąć niekorzystnego zjawiska fragmentacji zewnętrznej.

Mikroprocesor 80386 wykorzystuje dwa poziomy tablic do translacji adresu logicznego, uzyskanego w wyniku działania mechanizmów segmentacji, na adres fizyczny. Pierwszy poziom stanowi katalog stron, natomiast drugi poziom - tablica stron. Katalog stron zajmuje 4 KB pamięci i posiada 1024 32-bitowe rekordy. Adres bazowy katalogu stron przechowywany jest w rejestrze sterującym CR3. Starszych 10 bitów adresu logicznego jest wykorzystywane jako indeks wskazujący odpowiedni rekord katalogu stron. Starszych 20 bitów każdego rekordu katalogu stron jest adresem bazowym tablicy stron. Tablica stron posiada strukturę identyczną jak katalog stron. Bity adresowe A12 do A21 adresu logicznego stanowią indeks wskazujący odpowiedni rekord tablicy stron. Starszych 20 bitów rekordu tablicy stron w konkatenacji z 12 młodszymi bitami adresu logicznego tworzą



Rys. 4.4. Mechanizm stronicowania pamięci.

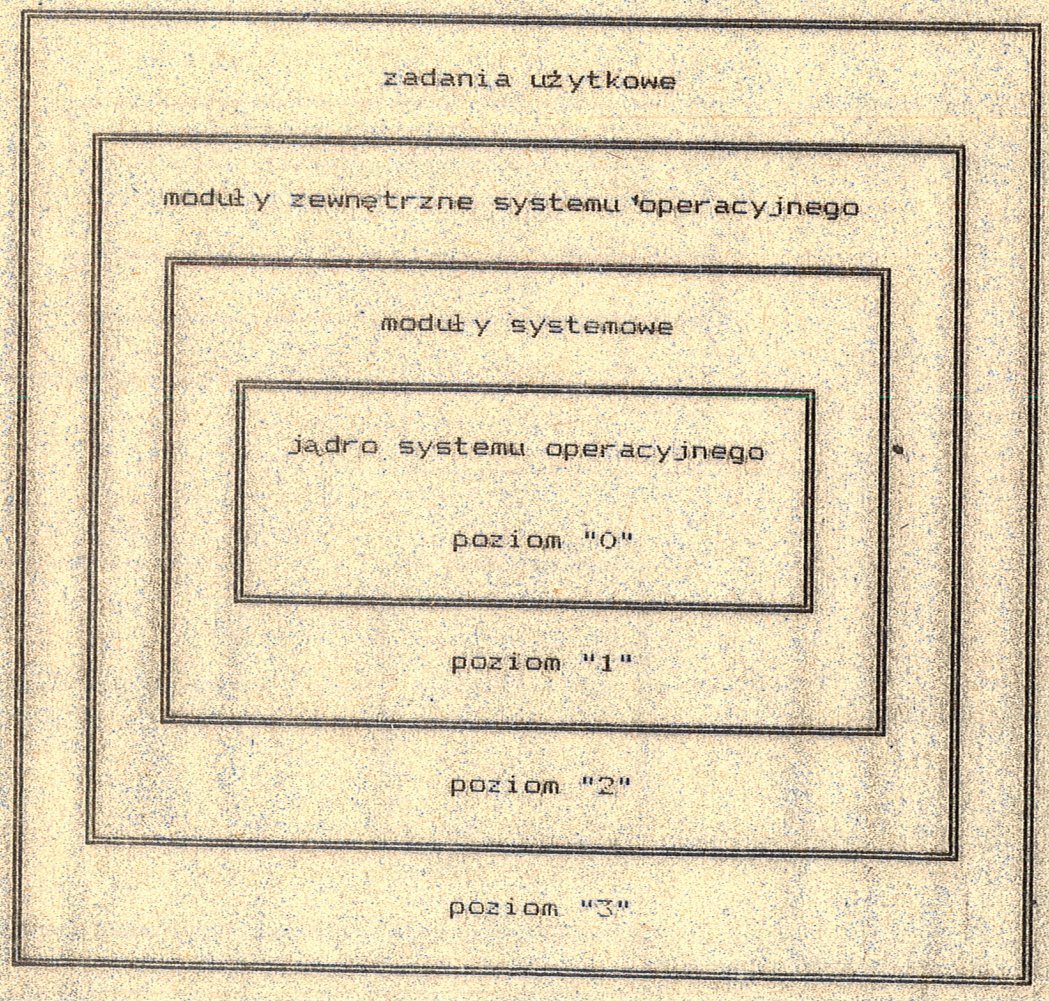
fizyczny adres pamięci. Rekordy katalogu stron i tablicy stron posiadają identyczną strukturę. Oprócz 20 bitów adresu zawierają informacje dotyczące wskazywanych rekordów tablicy stron lub wskazywanych stron. Informacje te dotyczą wykorzystania danego rekordu oraz mechanizmów protekcji dostępu do stron. Część bitów każdego rekordu może być wykorzystywana dowolnie przez system operacyjny. Sposób tworzenia adresu fizycznego z wykorzystaniem mechanizmów stronicowania jest przedstawiony na rysunku 4.4. Natomiast na rysunku 4.5. jest przedstawiony sposób tworzenia adresu fizycznego z wykorzystaniem mechanizmów segmentacji i stronicowania.



Rys. 4.5. Adresowanie pamięci z wykorzystaniem mechanizmów segmentacji i stronicowania.

5. OCHRONA PAMIĘCI W MIKROPROCESORZE 80386

W mikroprocesorze 80386 zrealizowano czteropoziomowy system protekcji (poziomów uprzywilejowania) zadań umożliwiający izolację i ochronę programów systemowych przed niepożądaną ingerencją zadań użytkowych. System priorytetów zapewnia ochronę instrukcji zastrzeżonych oraz ochronę dostępu do segmentów pamięci i deskryptorów. Mechanizmy ochrony pamięci wchodzą w skład jednostki zarządzania pamięcią. Ochrona pamięci w mikroprocesorze 80386 dotyczy zarówno segmentów jak i stron. Sposób wykorzystywania systemu priorytetów w wielozadaniowym systemie operacyjnym jest przedstawiony na rysunku 5.1.



Rys. 5.1. Wykorzystanie struktury priorytetów mikroprocesora 80386 w wielozadaniowym systemie operacyjnym.

5.1. Ochrona pamięci na poziomie segmentów

W systemie 80386 występują cztery poziomy priorytetów ponumerowane od 0 do 3. Najwyższy priorytet posiada poziom 0. Każdemu zadaniu wykonywanemu w systemie jest przyporządkowany priorytet. Dane przechowywane w segmencie o określonym priorytecie mogą być wykorzystywane, bez interwencji oprogramowania systemowego, jedynie przez zadania posiadające priorytet wyższy lub równy priorytetowi segmentu. Podobnie procedury przechowywane w segmencie kodu o określonym priorytecie mogą być wykorzystywane przez zadania posiadające taki sam lub wyższy priorytet.

Priorytety dotyczą zadań, deskryptorów, selektorów i instrukcji wejścia/wyjścia. Priorytety zadań pozwalają na ochronę dostępu poszczególnych zadań do procedur i danych innych zadań, w szczególności do procedur systemu operacyjnego. Priorytet, z którym wykonywane jest określone zadanie, zwany jest priorytetem bieżącym (*Current Privilege Level*). Bieżący priorytet zadania może być zmieniony jedynie przez przekazanie sterowania do segmentu posiadającego inny priorytet. Ponieważ przekazywanie sterowania odbywa się za pośrednictwem deskryptorów przejścia, system operacyjny kontroluje dostęp do zadań o priorytetach wyższych, niż priorytet danego zadania. Priorytety selektorów pozwalają na zwiększoną kontrolę dostępu poszczególnych zadań do segmentów danych. Priorytety instrukcji wejścia/wyjścia pozwalają na ustalenie grupy zadań, które mają prawo korzystać z tych instrukcji.

5.2. Ochrona pamięci na poziomie stron

Mechanizmy stronicowania wyróżniają dwa poziomy protekcji: poziom użytkownika i poziom systemu. Poziom użytkownika odpowiada najniższemu priorytetowi ochrony segmentów, natomiast poziom systemu odpowiada pozostałym priorytetom ochrony segmentów. Na obu poziomach protekcji strony mogą posiadać atrybut "tylko do odczytu" lub "odczyt i zapis". Każdy z rekordów katalogu stron i tablicy stron zawiera dwa bity, o nazwach U/S i R/W, określające protekcję dostępu do stron pamięci. Atrybuty protekcji w

rekordzie katalogu stron dotyczą wszystkich stron opisywanych w tablicy stron wskazywanej przez dany rekord, natomiast atrybuty protekcji w rekordzie tablicy stron dotyczą jedynie strony wskazywanej przez dany rekord.

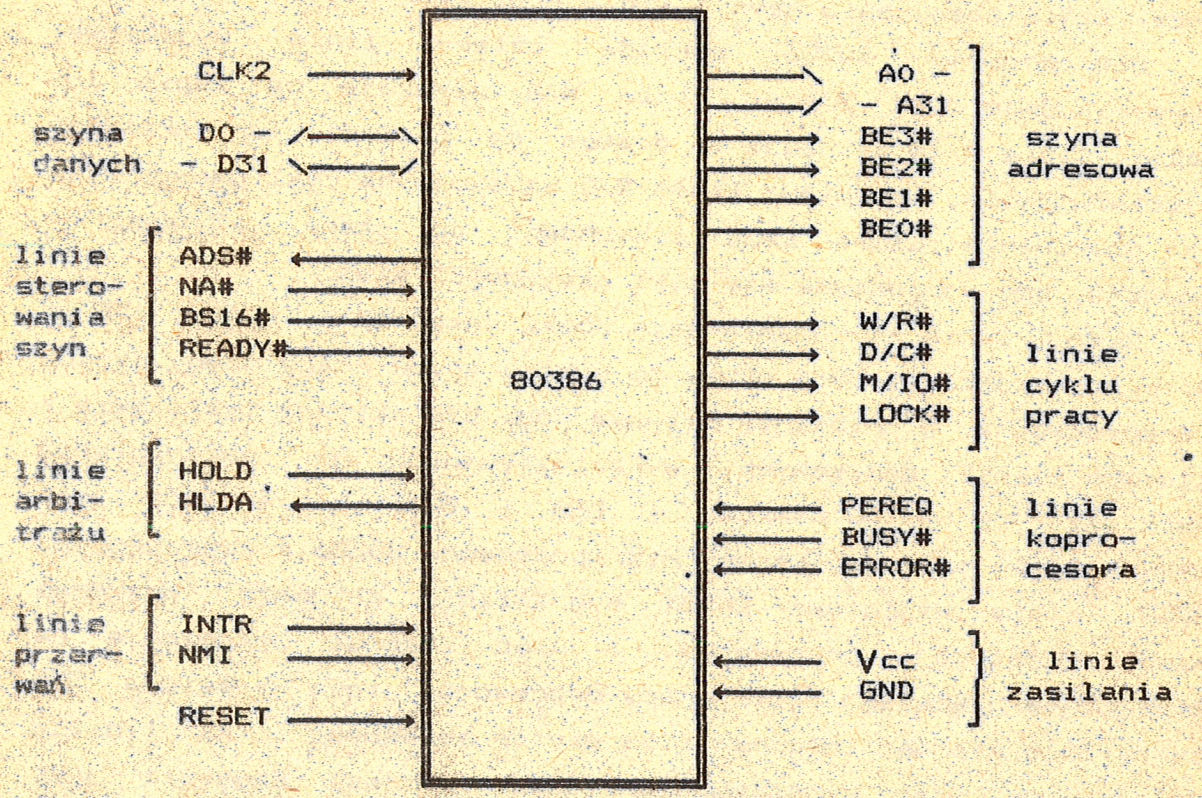
6. ZARZĄDZANIE ZADANIAMI

Mikroprocesor 80386 posiada bardzo efektywne sprzętowe mechanizmy przełączania zadań wykonywanych przez procesor. Zmiana wykonywanego zadania dokonywana jest w ciągu 17 mikrosekund. W tym czasie następuje przerwanie wykonywania bieżącego zadania, składowanie wszystkich niezbędnych elementów stanu zadania (stany rejestrów, przestrzeń adresowa i wskaźnik do poprzednio wykonywanego zadania) oraz załadowanie wartości odpowiadających nowemu zadaniu, sprawdzenie protekcji i przejście do wykonywania tego zadania. Przełączanie zadań następuje w wyniku wykonania instrukcji skoku lub wywołania podprogramu, pod warunkiem, że sterowanie ma być przekazane do programu znajdującego się w innym segmencie niż program wywołujący. Przełączenie może także nastąpić w przypadku wystąpienia przerwania programowago lub sprzętowego.

Istotną rolę w procesie przełączania zadań pełni rejestr segmentu stanu zadania TR (*Task State Segment Register*) oraz segment stanu zadania TSS (*Task State Segment*). Każde zadanie w systemie posiada własny segment stanu TSS, w którym przechowywane są wszystkie elementy stanu zadania. Bieżące zadanie identyfikowane jest przez rejestr TR, zawierający selektor deskryptora segmentu stanu tego zadania. Po napotkaniu instrukcji powrotu do poprzednio wykonywanego zadania, procesor składuje stan bieżącego zadania w jego segmencie stanu. Następnie procesor pobiera z tego segmentu informację o poprzednio wykonywanym zadaniu i zmienia zawartość rejestru stanu zadania. Na podstawie deskryptora stanu zadania procesor identyfikuje segment stanu zadania i odtwarza stan systemu sprzed przełączenia zadania.

7. ROZWIĄZANIA SPRZĘTOWE MIKROPROCESORA 80386

Mikroprocesor 80386 posiada wiele rozwiązań sprzętowych znacznie upraszczających konstruowanie układów współpracujących z nim. Procesor posiada oddzielne szyny: adresową i danych, 10 bitową szynę sterowania, trzy linie współpracy z koprocesorem, dwa wejścia przerwań oraz wejścia sygnału zegarowego i sygnału zerowania. Obudowa mikroprocesora posiada 132 wyprowadzenia. Wszystkie wyprowadzenia mikroprocesora są przedstawione na rysunku 7.1.



Rys. 7.1. Sygnały mikroprocesora 80386.

Szyna adresowa składa się z 34 linii. Cztery linie tej szyny (BE0 do BE3) są wykorzystywane bezpośrednio do uaktywniania poszczególnych bajtów pamięci operacyjnej. Pozostałe linie (A2 do A31) pozwalają na generowanie adresu z krokiem co 4 bajty. Taka organizacja szyny adresowej znacznie upraszcza budowę bloków pamięci współpracujących z mikroprocesorem oraz pozwala na wykorzystywanie pojedynczych bajtów jako operandów instrukcji wykonywanych przez procesor. Szyna adresowa jest wykorzystywana

do adresowania pamięci operacyjnej, urządzeń wejścia/wyjścia oraz do komunikacji z koprocesorem. Przestrzeń adresów wejścia/wyjścia wynosi 64 KB (od adresu 0 do FFFFH). Przy komunikacji z koprocesorem wykorzystywane są adresy od 800000F8H do 800000FFH z przestrzeni adresów wyjścia/wyjścia.

Szyna danych zawiera 32 dwukierunkowe linie. Odczyt danych z pamięci może być prowadzony z wykorzystaniem 16 lub 32 bitów. Natomiast w trakcie operacji zapisu mikroprocesor 80386 wysterowuje zawsze 32 linie, wysyłając tę samą informację na młodszej i starszej części szyny.

Szyna sterująca jest podzielona na trzy części: linie cyklu pracy, linie sterowania szyn i linie arbitrażu szyn.

Mikroprocesor 80386 posiada cztery linie wyjściowe, identyfikujące cykl pracy. Linia W/R wskazuje na zapis lub odczyt, linia M/IO wskazuje na dostęp do pamięci lub urządzeń wejścia/wyjścia, natomiast linia D/C wskazuje na cykl danych lub cykl sterowania. Linia LOCK przechodzi do stanu aktywnego w momencie, gdy rozpoczyna się cykl aktywności szyn.

Cztery linie sterowania szyn noszą nazwy BS16, READY, ADS i NA. Linia wejściowa BS16 służy do wyboru trybu pracy szyny danych mikroprocesora. W trybie 16 bitowym mogą być aktywne tylko dwie z czterech linii adresowych, wykorzystywanych do uaktywniania poszczególnych bajtów pamięci - BE0 i BE1. Linia BS16 jest aktywna także przy współpracy mikroprocesora 80386 z koprocesorem 80287. Linia wejściowa READY sygnalizuje gotowość urządzeń współpracujących z procesorem, a więc stanowi zezwolenie na zakończenie bieżącego cyklu pracy procesora. Linia wyjściowa ADS sygnalizuje ważność informacji na szynie adresowej i na liniach cyklu pracy. Linia wejściowa NA (*Next Address Request*) jest wykorzystywana do realizacji specjalnego trybu pracy szyny adresowej (*Address Pipelining*). Linia ta jest wysterowywana przez urządzenia współpracujące z mikroprocesorem. Po pojawieniu się na niej sygnału aktywnego, procesor wysyła na szynę adresową i linie cyklu pracy nowe wartości, pomimo że poprzedni cykl pracy nie został jeszcze zakończony. Taka organizacja pracy szyn mikroprocesora pozwala na skrócenie czasu dostępu do pamięci.

Linie arbitrażu szyn noszą nazwy HOLD (*Bus Hold Request*) i HLDA (*Bus Hold Acknowledge*). Linie te wykorzystywane są w sytuacji, gdy w systemie występuje więcej niż jeden układ mogący

przejmować sterowanie magistralą lokalną mikroprocesora. Linie arbitrażu szyn wykorzystywane są głównie w przypadku współpracy mikroprocesora z układem DMA oraz przy tworzenia układów tolerujących błędy sprzętowe, wykorzystujących dublowanie mikroprocesora.

Mikroprocesor posiada trzy wejściowe linie współpracy z koprocesorem arytmetycznym. Linie te noszą nazwy PEREQ (*Coprocessor Request*), BUSY (*Coprocessor Busy*) i ERROR (*Coprocessor Error*). Linia BUSY ma także dodatkowe znaczenie. W zależności od poziomu sygnału na tej linii w czasie trwania sygnału RESET, mikroprocesor wykonuje, lub nie, test wewnętrzny.

Mikroprocesor posiada dwie linie zgłoszenia przerwań: linię INTR i NMI. Linia INTR służy do zagłaszania przerwań maskowalnych. Przerwania mogą być maskowane programowo za pomocą jednego z bitów rejestru znaczników. Linia NMI służy do zgłaszania przerwań niemaskowalnych. W typowych aplikacjach linia przerwania niemaskowalnego jest dołączona do układu kontroli obecności zasilania, który generuje przerwanie w przypadku zaniku napięcia zasilającego.

Sygnał zegarowy steruje pracą wewnętrznej struktury mikroprocesora. W zależności od wykonania, mikroprocesor może być sterowany sygnałem zegarowym o częstotliwości 12,5 lub 16 MHz.

Sygnał zerowania mikroprocesora RESET jest sygnałem o najwyższym priorytecie. W czasie trwania tego sygnału wszystkie wyprowadzenia mikroprocesora są w stanie nieaktywnym. Po zakończeniu trwania tego sygnału mikroprocesor, w zależności od stanu linii BUSY, może rozpocząć wykonywanie autotestu. Testowane są wszystkie elementy mikroprocesora, które po sygnale zerowania posiadają zdefiniowaną wartość. Elementy te stanowią w przybliżeniu połowę struktury mikroprocesora. Autotest trwa około 30 milisekund. Wynik testu umieszczany jest w rejestrach EAX i EDX. Zerowe wartości tych rejestrów wskazują na poprawne zakończenie testu. Po sygnale zerowania mikroprocesor rozpoczyna realizację programu od adresu FFFFFFF0H. Po wykonaniu pierwszego skoku lub wywołaniu podprogramu wymagającego zmiany segmentu, linie adresowe A20 do A31 przyjmują wartość zero i mikroprocesor ogranicza przestrzeń adresową do pierwszego megabajtu fizycznej pamięci. Tym samym przechodzi do trybu pracy z adresowaniem rzeczywistym.

8. PROGRAMOWA ZGODNOŚĆ MIKROPROCESORÓW RODZINY iAPX 86

Do rodziny iAPX 86 należą mikroprocesory 8086, 80186, 80286 i 80386. Trzy pierwsze są mikroprocesorami 16 bitowymi, ostatni jest mikroprocesorem 32 bitowym. Charakterystyczną cechą tej rodziny jest fakt, że wszystkie kolejne mikroprocesory są zgodne ze swymi poprzednikami na poziomie programów wynikowych. Zgodność ta została utrzymana mimo wprowadzenia znacznych rozszerzeń funkcjonalnych, aż do zmiany długości słowa maszynowego włącznie. Celowość utrzymywania zgodności na poziomie programu wynikowego jest szczególnie widoczna na przykładzie mikroprocesora 80386. Pełne wykorzystanie możliwości mikroprocesora o tak złożonej strukturze wymaga bardzo rozbudowanego oprogramowania. Tworzenie takiego oprogramowania zajmuje często kilka lat. Jednak mikroprocesor nie musi "czekać" na pisane specjalnie dla niego systemy operacyjne. Może być wykorzystywany do realizacji już istniejącego oprogramowania. Znane są mikrokomputery klasy IBM PC wyposażone w mikroprocesory 80386. Oczywiście nie wykorzystują one wszystkich możliwości procesora lecz mimo to pracują znacznie szybciej i efektywniej od oryginalnych mikrokomputerów IBM PC XT/AT.

Firma Intel zapowiada, że również kolejne wersje procesorów rodziny iAPX 86 będą zachowywały zgodność ze swymi poprzednikami. Potwierdzeniem tej zapowiedzi jest fakt, że w mikroprocesorze 80386 wiele bitów rejestrów sterujących pozostaje niewykorzystane, z przeznaczeniem dla przyszłych zastosowań. Pisząc programy dla tego mikroprocesora należy pamiętać, aby przy odczytywaniu rejestrów maskować bity zarezerwowane, natomiast przy zapisie do rejestrów w miejsca zarezerwowane wpisywać zera. Przestrzeganie tej zasady pozwoli bezpośrednio przenosić programy napisane dla 80386 na następne generacje mikroprocesorów firmy Intel.

9. UWAGI KOŃCOWE

Przedstawiony powyżej opis mikroprocesora 80386 jest z konieczności niepełny, jednak pozwala zorientować się w funkcjonowaniu większości mechanizmów tego układu. W opisie nie uwzględniono między innymi listy rozkazów mikroprocesora, trybów

adresowania oraz mechanizmów obsługi przerwań. Wynika to z faktu, że oba te zagadnienia rozwiązane są podobnie jak w starszych wersjach mikroprocesorów. Główne różnice między mikroprocesorem 80286 i 80386 polegają na wprowadzeniu w pełni 32 bitowej struktury wewnętrznej (z zachowaniem jednak możliwości współpracy z 16 bitową magistralą zewnętrzną), na wprowadzeniu mechanizmów stonicowania pamięci operacyjnej oraz na znacznym rozszerzeniu przestrzeni adresów fizycznych i wirtualnych. Wydaje się, że przestrzeń adresowa jaką dysponuje procesor 80386 jeszcze przez wiele lat będzie dostatecznie duża nawet dla realizacji bardzo złożonych systemów programowych. Trudno nie zgodzić się z firmą IBM, która wybrała mikroprocesor 80386 na jednostkę centralną przyszłościowej serii mikrokomputerów IBM PS/2. Należy również zauważyć, że potencjalne możliwości mikroprocesora 80386 są ogromne. potrzebny jest jednak system operacyjny efektywnie te możliwości wykorzystujący.

BIBLIOGRAFIA

1. Edmund Strauss: Inside the 80286, Brandy Communications Company, 1986.
2. iAPX 286 Hardware Reference Manual, Technical Report 210760, Intel Corporation, Santa Clara, California, 1983.
3. iAPX 286 Programmer Reference Manual, Technical Report 210498, Intel Corporation, Santa Clara, California, 1983.
4. Microsystem Component Handbook, vol.I, Intel Corporation, Santa Clara, California, 1984.
5. iAPX 286 Operating Systems Writer's Guide, Intel Corporation, Santa Clara, California, 1983.
6. 80386 Hardware Reference Manual, Technical Report 231630, Intel Corporation, Santa Clara, California, 1985.