



ARCHITEKTURA SYSTEMÓW KOMPUTEROWYCH

PROJEKT ZALICZENIOWY

FILIP COPIJA

NR ALBUMU: 13618

GRUPA: Z/21 S lab1/1/IS

Organizacja procesora 8086

Jednostka sprzężenia z magistralą: Procesor komunikuje się z otoczeniem za pośrednictwem układu sprzężenia z magistralą. Układ ten realizuje wszystkie operacje dotyczące zewnętrznej magistrali procesora. Układ sterowania magistralą zawiera sumator oraz rejestry służące do obliczenia 20-bitowego adresu fizycznego. Dane przesyłane są po 16-bitowej magistrali danych. Magistrala adresowa jest 20-bitowa, co pozwala na bezpośrednie adresowanie pamięci operacyjnej o pojemności 1MB.

Jednostka wykonawcza: Jej głównym zadaniem jest dekodowanie oraz wykonywanie kolejnych rozkazów opuszczających, zapełnianą przez układ sprzężenia z magistralą, tzw. kolejkę rozkazów. Wykonanie rozkazów odbywa się przy udziale jednostki arytmetyczno-logicznej ALU (arithmetic logic unit).

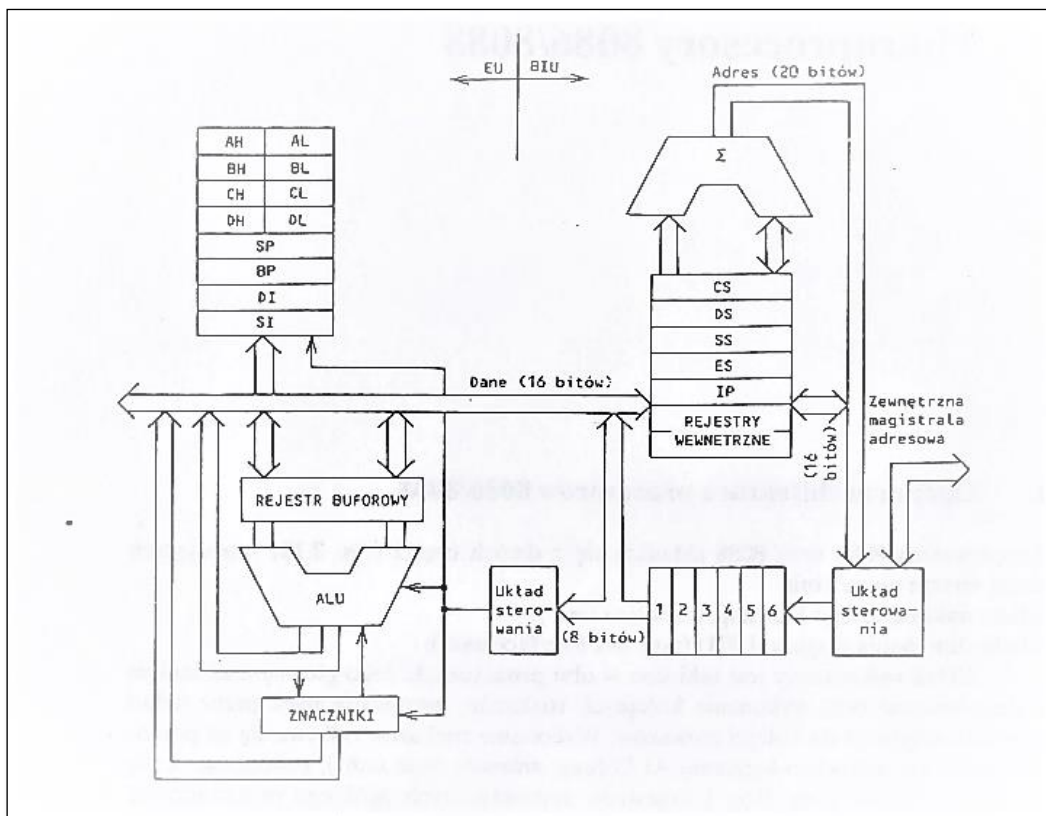
Jednostka arytmetyczno-logiczna: Jest tą częścią procesora, która wykonuje operacje arytmetyczne, takie jak dodawanie i odejmowanie słów danych, a także elementarne operacje logiczne.

Rejestry: Rejestr jest częścią pamięci wewnętrznej procesora o niewielkiej (liczonej w bitach) pojemności. Jest to zwykle układ obwodów elektrycznych (przerzutników) służący do przechowywania informacji. Pewne rejestry procesora są związane z określonymi operacjami, a wykonanie niektórych rozkazów jest związane z określonymi rejestrami. Rejestry są na ogół szybsze niż układy pamięci operacyjnej. W porównaniu z liczbą komórek pamięci ich liczba jest niewielka. Stąd w części adresowej rozkazu niewielka liczba bitów jest potrzebna dla wskazania rejestru. Główne zastosowanie rejestrów polega na przechowywaniu adresów lub danych przed lub w trakcie ich przetwarzania. Rejestry mogą być używane do modyfikacji adresów, pamiętania adresów powrotu z podprogramu, jako liczniki rozkazów, akumulatory pomocnicze lub małe pamięci podręczne.

Cykl pracy procesora 8086

Rozkazy procesora związane są wykonaniem pewnych stałych kroków zwanych fazami cyklu. W zależności od rodzaju rozkazu mogą wystąpić cztery spośród pięciu faz cyklu:

- **pobranie** – kod rozkazu zostaje pobrany z pamięci do zestawu rejestrów tworzących kolejkę rozkazów skąd trafi do rejestru rozkazów IR procesora;
- **dekodowanie** – tutaj rozkaz jest dekodowany i ustalane są połączenia arytmometru z rejestrami, która mają dostarczyć argumentów operacji;
- **odczyt** – jeżeli rozkaz wymaga argumentu z pamięci operacyjnej, to zostaje obliczony jego adres, a następnie argument jest odczytywany i wprowadzany do procesora;
- **wykonanie** – w tej fazie operacja, która została zdefiniowana rozkazem zostaje wykonana;
- **zapis** – jeżeli wynik operacji ma być umieszczony w pamięci operacyjnej, to zostaje obliczony adres i wykonany zapis do pamięci.



Rys. 1 Ogólna budowa mikroprocesora 8086

Czym jest adresowanie?

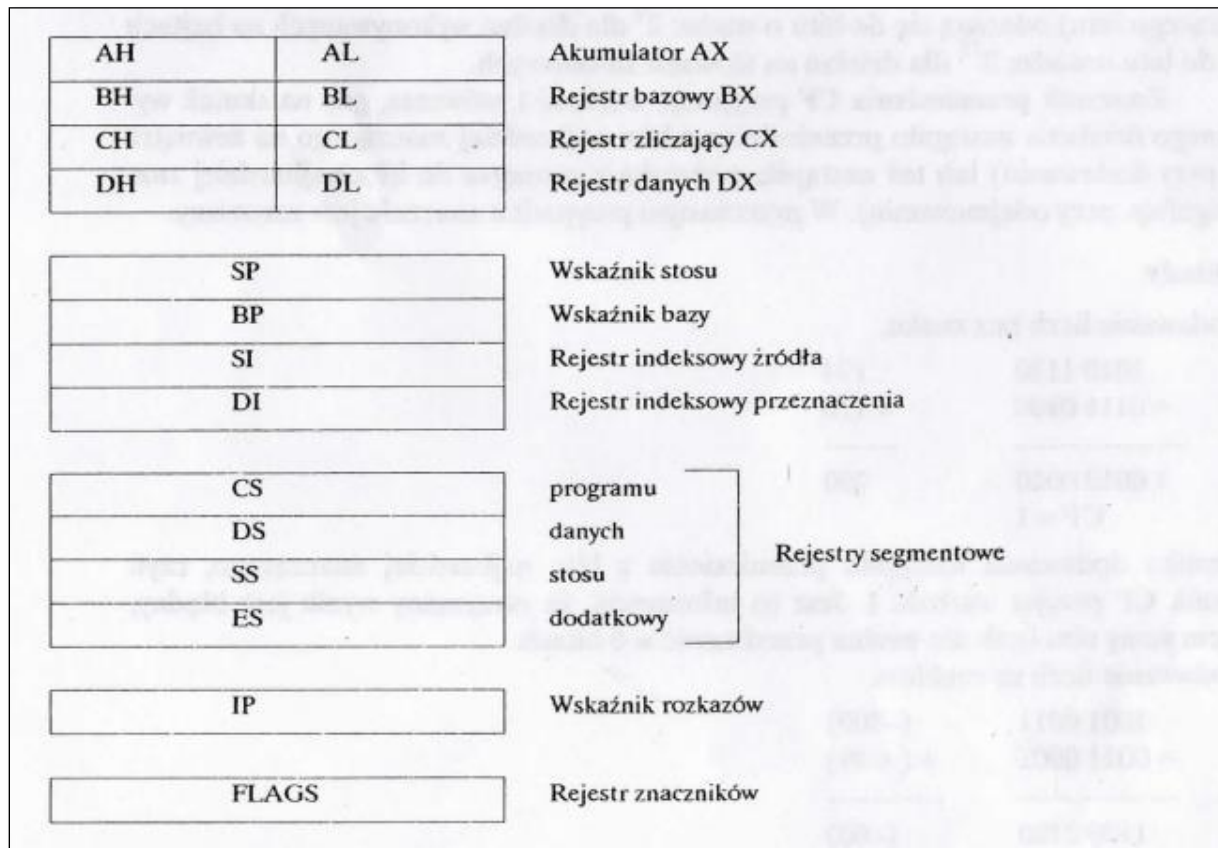
To format rozkazu wewnętrznego komputera, który określa podział słowa rozkazu na pola i sposób dekodowania poszczególnych pól. Najprostszy podział słowa rozkazu wyróżnia pole kodu operacji oraz pole adresowe. Pole adresowe może składać się z wielu pól określających jeden element kodu operacji rozkazu. Pole adresu zawiera adres pamięci operacyjnej, numer rejestru lub dane umieszczone w rozkazie. Tryb adresowania to rodzaj operacji jaką wykonuje procesor na zawartość pola adresowego rozkazu. Można go określić w jednym lub w kilku polach .

- **adresowanie natychmiastowe** - pole adresowe zawiera bezpośrednio operand, a Długość operandu zależy od kodu operacji lub od kodu operacji i dodatkowego pola sterującego w rozkazie. Ten tryb adresowania stosujemy, gdy np. chcemy bezpośrednio z rozkazu (bez odwoływania się do pamięci) załadować do rejestru prostą daną - bajt lub parę bajtów.
- **adresowanie bezpośrednie** - jest najbardziej podstawowym trybem adresowania. a. W tym trybie zawartość pola adresowego stanowi już finalny adres argumentu rozkazu. Ten tryb stosujemy, gdy nie zależy nam na tym, aby nasz program był przesuwalny w pamięci operacyjnej, lecz jest przeznaczony do wykonania przy zapisie w ściśle określone miejsce w pamięci.
- **adresowanie pośrednie** - rozkaz zawiera adres komórki pamięci operacyjnej, w której zawarty jest finalny adres operandu rozkazu. Ten tryb stosujemy, gdy chcemy, aby finalny adres operandu rozkazu mógł być dynamicznie wstawiony do komórki pośredniczącej w adresowaniu w czasie wykonywania programu.
- **adresowanie indeksowe** - jest inaczej nazywane modyfikacją adresu przez indeksowanie. W tym trybie wykorzystuje się specjalne rejestry procesora tzw. rejestry indeksowe (ang. index registers), które zawierają przesunięcie, który trzeba dodać do adresu istniejącego w rozkazie aby wyliczyć adres finalny operandu. Ten tryb adresowania pozwala przesunąć adres zawarty w rozkazie o wartość rejestru indeksowego. Używając tego trybu we wszystkich rozkazach programu, można osiągnąć możliwość wykonania programu przy załadowaniu w dowolne miejsce pamięci.

- **adresowanie względne** - polega na modyfikacji adresu zawartego w rozkazie przez aktualną zawartość licznika rozkazów. Ten tryb adresowania dostarcza innego sposobu osiągnięcia dynamicznej przesuwalności adresów dostępu do danych, tj. gdy nie chcemy lub nie możemy znać przesunięcia całości programu w stosunku do adresu zerowego. Przy tym trybie adresowania, finalny adres danej jest wyliczany względem bieżącej zawartości licznika rozkazów, a więc do rozkazu wstawiamy przesunięcie danej w programie względem adresu następnego rozkazu.
- **adresowanie pośrednie indeksowe** - zapewnia jednoczesną możliwość zastosowania w programie adresowania pośredniego z modyfikacją adresu odczytanego z komórki pośredniczącej poprzez zawartość rejestru indeksowego. Adresowanie pośrednie indeksowe pozwala na dwupoziomowe dynamiczne określanie adresu danych w wyniku obliczeń wykonanych w poprzedzających fragmentach programu, metodą wpisania pewnego adresu bazowego a następnie modyfikacji tego adresu bazowego przez rejestr indeksowy.
- **adresowanie indeksowe pośrednie** - zapewnia najpierw modyfikację adresu zawartego w rozkazie przez zawartość rejestru indeksowego a następnie tak otrzymany adres jest stosowany do wskazania komórki pamięci, w której jest przechowywany finalny adres operandu rozkazu. Adresowanie indeksowe pośrednie pozwala na dynamiczne określanie adresów danych w programie w czasie wykonania programu, przy zapewnieniu przesuwalności w pamięci całego programu. Dla tego celu, program piszemy począwszy od adresu zerowego w pamięci stosując adresowanie indeksowe do wszystkich rozkazów z wyjątkiem tych, które działają na danych wybieranych dynamicznie.
- **adresowanie rejestrowe** - stosuje się, gdy dana dla rozkazu jest przechowywana w rejestrze. Część adresowa rejestru zawiera wtedy jedno lub więcej pól, w których znajdują się identyfikatory rejestrów. Adresowanie rejestrowe jest często stosowane w tym samym rozkazie razem z innymi trybami adresowania dotyczącymi pamięci operacyjnej. Wówczas rozkaz dotyczy zarówno pamięci operacyjnej jak i rejestrów procesora.
- **adresowanie pośrednie rejestrowe** - polega na tym, że jako miejsce pobrania finalnego adresu operandu rozkazu stosuje się rejestr procesora, którego identyfikator umieszczony jest w polu adresowym rozkazu. Przy pomocy tego trybu adresowania można dynamicznie określić finalny adres operandu poprzez odpowiednie załadowanie zawartości rejestru, w zależności od przebiegu obliczeń w programie.

Rejestry procesora 8086

Rejestry te są wykorzystywane jako uniwersalne do przechowywania danych i wykonywania różnych operacji, ale jednocześnie pełnią pewne funkcje specjalne odpowiadające ich nazwom:



Rys. 2 Rejestry dostępne programowo

Rejestry podstawowe

AX (accumulator) – akumulator

BX (basis register) – rejestr bazowy

CX (count register) – rejestr zliczający

DX (data register) – rejestr danych

Rejestry te są 16-bitowe. Przez niektóre rozkazy mogą być traktowane jako pary rejestrów 8-bitowych z niezależnym dostępem do części mniej oraz bardziej znaczącej (**low/high**). Odpowiednie rejestry przyjmują wtedy nazwy:

- AL, AH;
- BL, BH;
- CL, CH;
- DL, DH.

Akumlator **AX** to rejestr, w którym umieszczony jeden z argumentów operacji arytmetycznej po wykonaniu rozkazu może być zamieniony przez wynik operacji. Rozkazu wykorzystujące rejestr akumlatora mają zawarty kod, a czas ich wykonania jest krótszy.

Rejestr bazowy **BX** służy do przechowywania adresu bazowego. Jego zawartość przez niektóre rozkazy może być potraktowana jako odniesienie przy obliczaniu adresów podczas wykonywania programu komputera.

Rejestr zliczający to rejestr **CX**. Może być wykorzystywany jako licznik sterujący wykonywaniem pętli programowych i do operacji na łańcuchach (ciągach słów).

Rejestr **DX** jest stosowany jako starsza część słowa danych podczas wykonywania operacji mnożenia i dzielenia. Kiedy mnożymy dwie wartości 16-bitowe do zapamiętywania 34-bitowego wyniku potrzebne są dwa rejestry **DX**. Z kolei w przypadku dzielenia w rejestrach tych umieszczana jest 32-bitowa dzielna.

Rejestry wynikowe i indeksowe

Mikroprocesor 8086 posiada dwa rejestry wskaźnikowe i dwa rejestry indeksowe. Wszystkie są rejestrami 16-bitowymi. Przechowywane w nich dane mogą być użyte jako argumenty większości rozkazów arytmetycznych i logicznych.

SP (stack pointer) – wskaźnik stosu

BP (base pointer) – wskaźnik bazy

DI (source index register) – rejestr indeksowy źródła

SP jest to rejestr wskaźnikowy stosu stosowany do adresowania danych w obrębie wydzielonego obszaru pamięci, zwanego segmentem stosu. Jest modyfikowany przy standardowych operacjach prowadzonych z wykorzystaniem stosu w taki sposób, że zawiera adres ostatnio zapamiętanej danej.

Rejestr **BP** może dostarczać jeden ze składników adresu bazowego. Adresem efektywnym nazywamy, wyznaczony podczas wykonywania niektórych instrukcji, adres segmentu odczytywanego z pamięci lub adres komórki, w której zapisywany jest wynik. Wskaźnik **BP** wykorzystywany też jest podczas operacji niestandardowych np. podczas pobierania parametrów przekazywanych przez stos.

Rejestry segmentowe

Te rejestry wykorzystywane są do adresowania pamięci operacyjnej. Pamięć o maksymalnej wielkości 1 MB. Poszczególne segmenty są identyfikowane przy pomocy adresów bazowych przechowywanych w czterech wyspecjalizowanych rejestrach segmentowych:

- **CS;**
- **DS;**
- **ES;**
- **SS.**

CS (code segment register) – rejestr segmentowy program wskazujący segment programu, z którego aktualnie pobierane są kolejne rozkazy do wykonania

DS (data segment register) – rejestr segmentowy danych, który wskazuje segment, w którym zapamiętywane są zmienne używane w programie

ES (extra segment register) – rejestr segmentowy dodatkowy wskazuje położenie dodatkowego segmentu danych

SS (stack segment register) – rejestr segmentowy stosu wskazuje segment pamięci, w którym pamięć może być adresowana za pośrednictwem wskaźnika stosu SP (stack pointer)

Licznik rozkazów

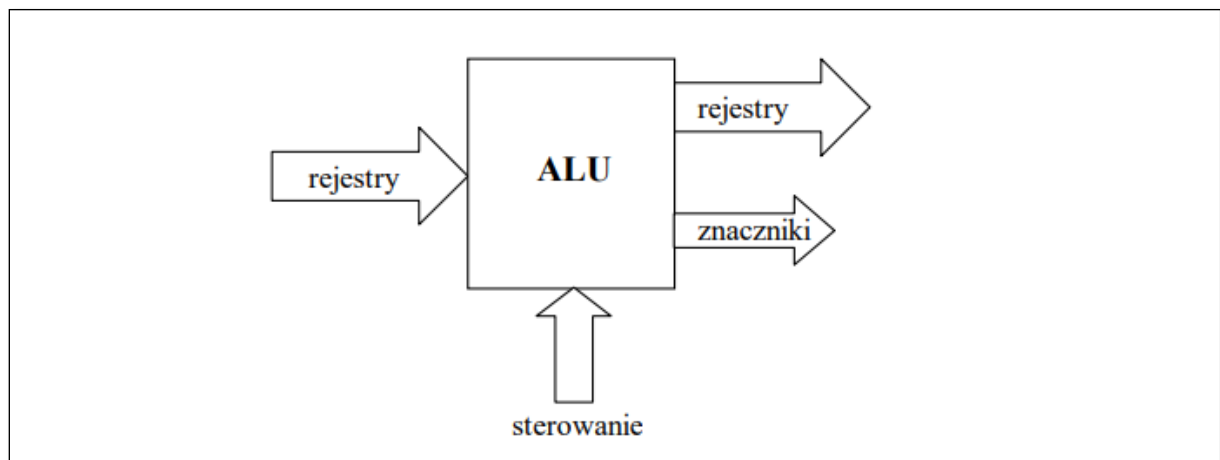
PC (program counter) razem z rejestrem segmentowym **CS** (code segment register) adresuje kolejne rozkazy przeznaczone do wykonania. **PC** wskazuje początek segmentu programu.

Znaczni

Każdy znacznik, czyli **flag** jest bitem w rejestrze znaczników, który wskazuje czy wystąpił określony stan. Niektóre instrukcje w wyniku działania ustawiają znaczniki, inne pozwalają badać ich stan.

Jednostka arytmetyczno-logiczna

Ta część jednostki wykonawczej procesora 8086 skupiona jest z układów wykonujących operacje arytmetyczne i logiczne. Argumentami mogą być bajty lub słowa. Źródłem argumentów operacji wykonywanych przez **ALU** może być dostępny programowo rejestr procesora, komórka pamięci lub operand bezpośredni w instrukcji programu. W fazie wykonania dane wejściowe jednostki znajdują się w rejestrach, a wyniki kierowane są do rejestrów i znaczników arytmetycznych.



Rys. 3 Wejścia i wyjścia jednostki arytmetyczno-logicznej