

## 12.1 Architektura mikroprocesora Intel 8086

### 1. Wprowadzenie

Mikroprocesory można dzielić według różnych kryteriów. Najczęściej dzielimy je na

- mikroprocesory CISC (ang. Complex Instruction Set Computer)
- mikroprocesory RISC (ang. Reduced Instruction Set Computer)
- mikroprocesory sygnałowe (mikroprocesory DSP)
- mikrokontrolery (mikroprocesory jednokładowe)

Najbardziej znane firmy produkujące mikroprocesory to Intel, Motorola, AMD (Advanced Micro Devices), Cyrix, IDT, Texas Instruments, DEC (Digital Equipment Corporation), IBM, Harris, HP (Hewlett Packard), Samsung, Atmel, Phillips. Skoncentrujemy się na mikroprocesorach firmy Intel z uwagi na ich ogromną popularność.

Intel 4004 to pierwszy mikroprocesor na świecie. Został zaprojektowany w roku 1969. Był to procesor 4-bitowy. Intel 8080 to pierwszy 8-bitowy mikroprocesor na świecie.

Intel 8086 to mikroprocesor 16-bitowy, który zainicjował rodzinę mikroprocesorów x86. Został wprowadzony do produkcji w 1978 roku, zawierał około 29 tys. tranzystorów i w momencie wprowadzania do produkcji był taktowany zegarem 8 MHz. Równolegle z mikroprocesorem Intel 8086 był produkowany Intel 8088. Była to odmiana mikroprocesora 8086 z 8-bitową szyną danych. Z Intel 8086 i 8088 współpracował układ koprocesora numerycznego Intel 8087. Bardzo popularnym mikroprocesorem 16-bitowym był również Intel 80286 wprowadzony do produkcji w roku 1982. Intel 286 zawierał około 134 tys. tranzystorów i został w nim po raz pierwszy wprowadzony tzw. chroniony tryb pracy (ang. protected mode).

Intel 80386 był pierwszym 32-bitowym mikroprocesorem rodziny x86. Z kolei Intel 80486 był pierwszym mikroprocesorem rodziny x86 zawierającym wewnątrz chipu mikroprocesora koprocesor numeryczny oraz 8 kB pamięci notatnikowej typu cache. Kolejne mikroprocesory rodziny x86 to już zupełnie współczesne konstrukcje 32-bitowe: Pentium, Pentium II, Pentium III i Pentium IV.

We wszystkich mikroprocesorach Pentium mamy 2 pamięci cache L1 i L2 (pamięć cache pierwszego i drugiego poziomu), maksymalną pojemność pamięci operacyjnej 64 GB i 64-bitową szynę danych. Wszystkie mikroprocesory Pentium wyposażone są w koprocesor czyli FPU (8 rejestrów 80 bitowych). Pentium 4 ma dodatkowo 64 bitową jednostkę MMX (8 rejestrów 64 bitowych) i 128 bitową jednostkę SSE (Streaming SIMD Extensions). Przypominamy, że skrót SIMD oznacza Single Instruction Multiple Data (czyli przetwarzanie równoległe polegające na zastosowaniu tej samej instrukcji do różnych danych)

Bardzo ważną cechą rodziny x86 jest to, że programy pisane na wcześniejsze mikroprocesory rodziny mogą być uruchamiane na mikroprocesorach wyższych w hierarchii rodziny tzn. opracowanych później. Zatem wymiana procesora nie musi pociągać za sobą wymiany oprogramowania.

Warto jeszcze wspomnieć o mikrokontrolerach Intel 8051 i Intel 8052. To bardzo popularne, udane mikrokontrolery, których odpowiedniki produkowane są przez wiele firm.

## 2. Kody wykorzystywane w mikroprocesorach rodziny Intel x86

Najczęściej wykorzystywanym kodem alfanumerycznym jest kod ASCII w różnych odmianach.

Z kodów numerycznych wykorzystywane są najczęściej 4 kody: kod NKB, kod uzupełnień do 2, kod BCD 8421 i zapis zmiennoprzecinkowy (w koprocesorze). Pamiętajmy, że w kodzie NKB dla  $a_n a_{n-1} \dots a_0$ ,  $a_i \in \{0,1\}$  mamy

$$a_n a_{n-1} \dots a_0 = \sum_{i=0}^n a_i 2^i$$

Dla kodu uzupełnień do 2 (czyli U2) dla  $a_n a_{n-1} \dots a_0$ ,  $a_i \in \{0,1\}$  mamy

$$a_n a_{n-1} \dots a_0 = -a_n \cdot 2^n + \sum_{i=0}^{n-1} a_i 2^i$$

Warto przypomnieć, że algorytmy podstawowych działań arytmetycznych w tym kodzie są dobrze dopasowane do sprzętu. Co przekładając na język praktyki oznacza, że są szybkie i łatwe do implementacji. Wyjaśnia to popularność zapisu U2 w mikroprocesorach.

Kod BCD (8421) wykorzystywany jest w dwu wersjach jako spakowany kod BCD (packed BCD) i niespakowany (unpacked BCD).

Zapis zmiennoprzecinkowy używany jest w kilku formatach zgodnych ze standardem IEEE 754.

Warto pamiętać, że znacznik OF =1 (OF to overflow flag) oznacza nadmiar w zapisie uzupełnień do 2, a nie w zapisie NKB.

## 3. Architektura mikroprocesora 8086

Intel 8086 jest jednym z pierwszych mikroprocesorów 16-bitowych. Został wprowadzony do produkcji przez firmę Intel w roku 1980.

Układ mikroprocesora 8086 składa się z 2 pracujących niezależnie głównych części BIU (Bus Interface Unit) i części wykonawczej EU (Execution Unit). W BIU znajduje się bufor mieszczący kolejkę rozkazów o długości 6 bajtów.

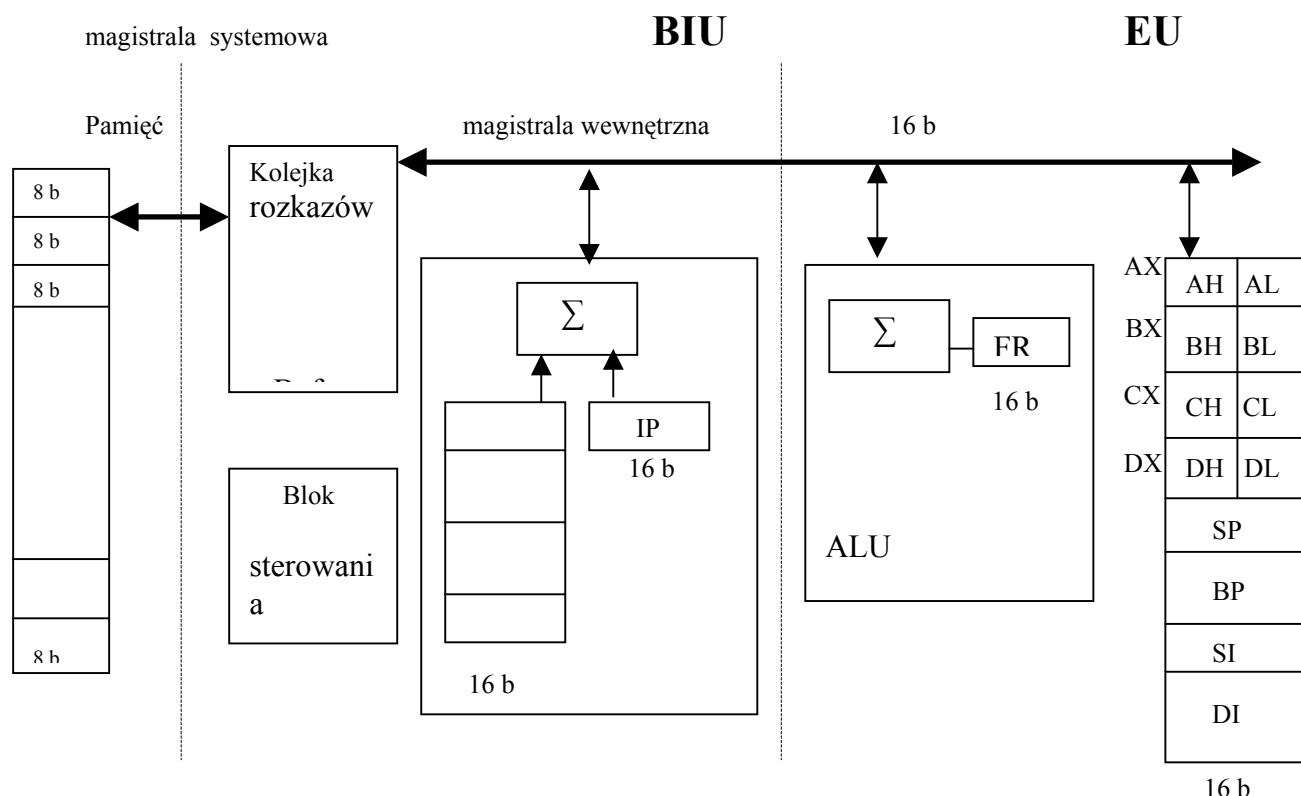
Licznik rozkazów PC (*Program Counter* lub *Program Pointer* lub *Instruction Pointer*) jest 16-bitowy.

Pamięć operacyjna w systemie 8086 zorganizowana jest bajtowo, tzn. adresowany jest każdy bajt i ma maksymalną pojemność 1 MB tzn.  $2^{20}$  bajtów. Adres fizyczny (wystawiany na końcówkach szyny adresowej) ma 20 bitów.

Cztery 16 bitowe rejestry ogólnego przeznaczenia AX, BX, CX, DX mogą być traktowane jako pary rejestrów 8 bitowych tzn.

$$AX=(AH,AL), \quad BX=(BH,BL), \quad CX=(CH,CL), \quad DX=(DH,DL)$$

W naturalny sposób w mikroprocesorze 8086 można operować bajtami.



**Rys 1. Architektura mikroprocesora Intel 8086. CS, DS, SS, ES to 16-bitowe rejestry segmentowe. Blok rejestrów ogólnego przeznaczenia to 16-bitowe rejestry AX, BX, CX, DX. Blok rejestrów wskaźnikowych i indeksowych to 16-bitowe rejestry SP, BP, SI, DI.**

Układ 8086 zawiera osiem 16-bitowych *rejestrów roboczych*.

Pierwsza grupa tych rejestrów to 4 *rejestry ogólnego przeznaczenia*:

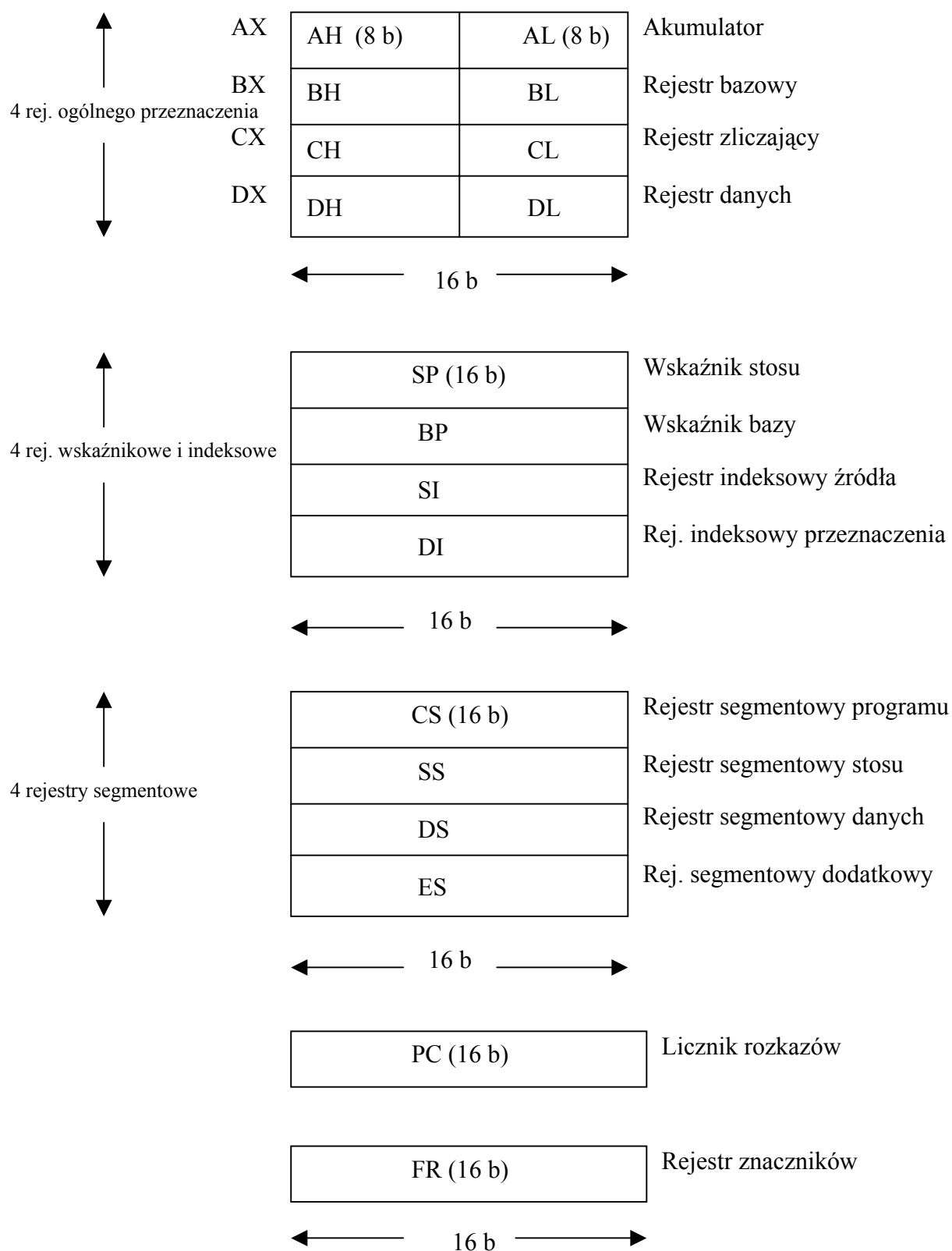
- AX – rejestr akumulatora, mówimy krótko akumulator
- BX – rejestr bazowy (base register)
- CX - rejestr zliczający (count register), służy do zliczania liczby obiegów pętli
- DX – rejestr danych (data register)

Druga grupa to *rejestry indeksowe i wskaźnikowe*:

- SP – wskaźnik stosu (stack pointer)
- BP – wskaźnik bazowy (base pointer)
- SI – rejestr indeksowy źródła (source index register)
- DI – rejestr indeksowy przeznaczenia (destination index register)

Grupa 4 *rejestrów segmentowych* to:

- CS – rejestr segmentu programu (code segment register)
- DS – rejestr segmentu danych (data segment register)
- SS – rejestr segmentu stosu (stack segment register)
- ES – rejestr segmentu dodatkowego



Rys. 2. Rejestry mikroprocesora 8086

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU	NU	NU	NU	OF	DF	IF	TF	SF	ZF	NU	AF	NU	PF	NU	CF

**Rys. 3. Rejestr znaczników FR (Flag Register) i usytuowanie znaczników w rejestrze FR;**  
symbolem NU (ang. *not used*) oznaczone są niewykorzystane pozycje rejestru FR

Struktura 16-bitowego *rejestru znaczników* FR (ang. Flag Register) pokazana jest na rys.3. Znacznik (ang. *flag* lub *status flag*) to pozycja rejestru znaczników pozwalająca zapamiętać sytuację taką jak np. wystąpienie nadmiaru lub pozwalająca ustawić parametr taki jak DF (kierunek inkrementacji) decydujący o sposobie wykonania pewnych instrukcji. W słowie 16-bitowym mamy 9 znaczników o następującym znaczeniu:

- OF - nadmiar w zapisie U2 (ang. *overflow flag*). OF = 1 oznacza wystąpienie nadmiaru
- DF - kierunek inkrementacji (ang. *direction flag*), ustawiany jest rozkazem STD a zerowany rozkazem CLD. DF wskazuje czy zawartości rejestrów SI i DI mają być zwiększane o1 czy zmniejszane o1.
- IF - znacznik włączający i wyłączający system przerwań (ang. *interrupt flag*) IF = 1 zezwala na przyjmowanie przerwań, IF = 0 oznacza brak zezwolenia na przyjmowanie przerwań
- TF - znacznik pracy krokowej (ang. *trap flag*)
- SF - znacznik znaku (ang. *sign flag*), SF = 0 liczba dodatnia, SF = 1 liczba ujemna
- ZF - znacznik zera (ang. *zero flag*), ZF = 0 wynik różny od 0, ZF = 1 wynik równy 0
- AF - przeniesienie pomocnicze (ang. *auxiliary carry flag*), jest to przeniesienie z bitu 3 na 4
- PF - znacznik parzystości (ang. *parity flag*), PF = 0 liczba jedynek wyniku jest nieparzysta, PF = 1 liczba jedynek wyniku jest parzysta
- CF - znacznik przeniesienia (ang. *carry flag*)

#### 4. Segmentacja pamięci operacyjnej

Pamięć operacyjna z punktu widzenia programisty podzielona jest na tzw. segmenty (spójne fragmenty pamięci) o maksymalnym rozmiarze 64 kB, które mogą na siebie zachodzić.

Procesor 8086 ma 4 rejestry segmentowe CS,DS,SS,ES. Rejestr segmentowy kodu programu CS wskazuje segment, z którego pobierane są kolejne rozkazy do wykonania. Rejestr segmentowy danych DS wskazuje segment, w którym zapamiętywane są dane używane w programie. Rejestr segmentowy stosu SS wskazuje segment pamięci w którym zdefiniowany jest stos. Rejestr segmentu dodatkowy ES wskazuje dodatkowy segment danych

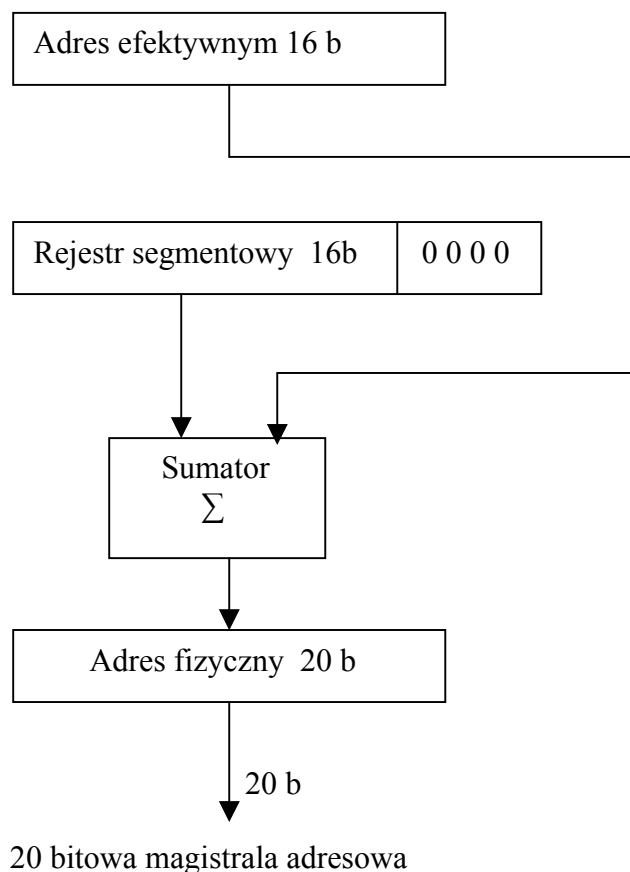
Każde odwołanie się do pamięci musi zawierać adres segmentu, czyli adres początkowy segmentu (symbolicznie oznaczany jako SEGMENT) i przemieszczenie w ramach segmentu (symbolicznie oznaczamy to przemieszczenie symbolem OFFSET).

Powszechnie stosowaną notacją dla oznaczania adresu jest zapisanie najpierw adresu segmentu a za nim po dwukropku przesunięcia.. Pełny fizyczny adres w pamięci operacyjnej notujemy więc w postaci

SEGMENT : OFFSET

przy czym jeden adres fizyczny można przedstawić na wiele sposobów w postaci pary uporządkowanej `SEGMENT:OFFSET`. Na przykład adres pewnej danej w pamięci operacyjnej możemy zapisać tak: `40H:12H` co oznacza, że dana znajduje się w segmencie o adresie `40H` a przesunięcie w ramach segmentu wynosi `12H`.

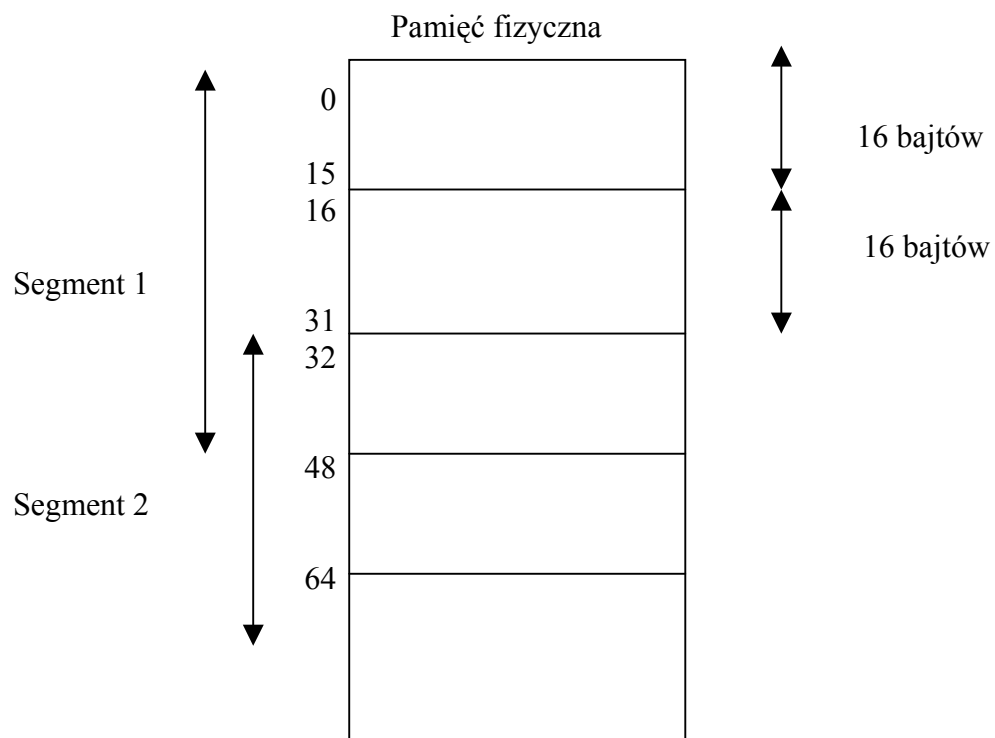
Możemy również używać w tym zapisie nazw rejestrów segmentowych. Możemy np. zapisać `DS:24` co oznacza, że dana znajduje się w segmencie danych (o adresie: zawartość `DS` razy  $2^4$ ) z przesunięciem w segmencie równym `24`.



**Rys. 4. Adresowanie pamięci;** adres fizyczny uzyskujemy przez dodanie arytmetyczne 16-bitowego adresu efektywnego (jest to tzw. `OFFSET` lub przemieszczenie wewnątrz segmentu) do wartości zapisanej w rejestrze segmentu (w kodzie NKB) pomnożonej przez  $2^4$ ; adres efektywny uzyskujemy zależnie od trybu adresowania

Rysunek 4. ilustruje zasadę segmentacji pamięci. Każdy segment zaczyna się na granicy będącej wielokrotnością 16 (a więc 0, 16, 32, 48 itd.) co pozwala zaadresować segmenty 16-bitowym słowem. Tworzenie 20-bitowego adresu fizycznego polega na uzupełnieniu słowa 16-bitowego 4-ma zerami od strony najmniej znaczących bitów.. Procesor automatycznie przyjmuje, że 4 najmłodsze bity z 20-bitowego adresu segmentu są zerami. Odwołując się do danej procesor dodaje do 20-bitowego adresu segmentu 16-bitowe przesunięcie tej danej w segmencie. Jest to fizyczny adres danej w pamięci operacyjnej.

**Uwaga:** System operacyjny MS-DOS automatycznie wypełnia rejestry `CS`, `ES`, i `SS`. Natomiast o rejestr `DS` program użytkownika musi zadbać sam, tzn. program użytkownika musi sam załadować rejestr `DS` adresem segmentu danych programu. Jeśli dane programu nie mieszczą się w 64 kB obszarze pamięci przeznaczony standardowo na dane, to program użytkownika musi sam modyfikować zawartość rejestrów `DS` i `ES`.



**Rys.5. Segmentacja pamięci; segment jest podobny do „okienka” przesuwanego w górę i w dół pamięci, ze skokiem 16**