Operacje na łańcuchach

Operacje na łańcuchach

- MOVS/MOVSB/MOVSW/MOVSD/MOVSQ Prześlij łańcuch/bajtów/słów/podwójnych słów/poczwórnych słów
- CMPS/CMPSB/CMPSW/CMPSD/CMPSQ Porównaj łańcuchy/bajtów/słów/podwójnych słów/poczwórnych słów
- SCAS/SCASB/SCASW/SCASD/SCASQ Skanuj łańcuch/bajtów/słów/podwójnych słów/poczwórnych słów
- LODS/LODSB/LODSW/LODSD/LODSQ Ładuj łańcuch/bajtów/słów/podwójnych słów/poczwórnych słów
- STOS/STOSB/STOSW/STOSD/STOSQ Zapamiętaj łańcuch/bajtów/słów/podwójnych słów/poczwórnych słów
- REP Powtarzaj dopóki ECX nie jest zerem
- REPE/REPZ Powtarzaj dopóki equal/zero
- REPNE/REPNZ Powtarzaj dopóki not equal/not zero

Wpływa na flagi: -

Instrukcja MOVS/MOVSB

movs byte ptr [(r|e)di], [(r|e)si]
movsb

Przesyła bajt z pamięci ds:(r|e)si do pamięci es:(r|e)di. Rejestry (r|e)di/(r|e)si są zwiększane/zmniejszane o 1 w zależności od flagi DF (0/1).

[(r|e)s:edi] := [ds:(r|e)si]
(r|e)di := (r|e)di ±1
(r|e)si := (r|e)si ±1

movsb

Wpływa na flagi: -

Instrukcja MOVS/MOVSW

movs word ptr [(r|e)di], [(r|e)si]
movsw

Przesyła słowo z pamięci ds:(r|e)si do pamięci es:(r|e)di. Rejestry (r|e)di/(r|e)si są zwiększane/zmniejszane o 2 w zależności od flagi DF (0/1).

[es:(r|e)di] := [ds:(r|e)si]
(r|e)di := (r|e)di ±2
(r|e)si := (r|e)si ±2

movsw

Wpływa na flagi: -

Instrukcja MOVS/MOVSD

movs dword ptr [(r|e)di], [(r|e)si]
movsd

Przesyła podwójne słowo z pamięci ds:esi do pamięci es:edi. Rejestry (r|e)di/(r|e)si są zwiększane/zmniejszane o 4 w zależności od flagi DF (0/1).

[es:(r|e)di] := [ds:(r|e)si]
(r|e)di := (r|e)di ±4
(r|e)si := (r|e)si ±4

movsd

Wpływa na flagi: -

Instrukcja MOVS/MOVSQ

movs qword ptr [(r|e)di], [(r|e)si]
movsq

Przesyła poczwórne słowo z pamięci ds:(r|e)si do pamięci es:(r|e)di. Rejestry (r|e)di/(r|e)si są zwiększane/zmniejszane o 8 w zależności od flagi DF (0/1).

[es:(r|e)di] := [ds:(r|e)si]
(r|e)di := (r|e)di ±8
(r|e)si := (r|e)si ±8

movsq

Wpływa na flagi: OSZAPC

Instrukcja CMPS/CMPSB

cmps byte ptr [(r|e)si], [(r|e)di]

cmpsb

Porównuje bajt z pamięci ds:(r|e)si i z pamięci es:(r|e)di. Rejestry (r|e)di/(r|e)si są zwiększane/zmniejszane o 1 w zależności od flagi DF (0/1).

[ds:(r|e)si] - [es:(r|e)di]

(r|e)di := (r|e)di ±1

(r|e)si := (r|e)si ±1

cmpsb

Wpływa na flagi: OSZAPC

Instrukcja CMPS/CMPSW

cmps word ptr [(r|e)si], [(r|e)di]

cmpsw

Porównuje słowo z pamięci ds:(r|e)si i z pamięci es:(r|e)di. Rejestry (r|e)di/(r|e)si są zwiększane/zmniejszane o 2 w zależności od flagi DF (0/1).

[ds:(r|e)si] - [es:(r|e)di]

(r|e)di := (r|e)di ± 2

(r|e)si := (r|e)si ± 2

cmpsw

Wpływa na flagi: OSZAPC

Instrukcja CMPS/CMPSD

cmps dword ptr [(r|e)si], [(r|e)di]

cmpsd

Porównuje podwójne słowo z pamięci ds:(r|e)si i z pamięci es:(r|e)di. Rejestry (r|e)di/(r|e)si są zwiększane/zmniejszane o 4 w zależności od flagi DF (0/1).

[ds:(r|e)si] - [es:(r|e)di]

(r|e)di := (r|e)di ± 4

(r|e)si := (r|e)si ± 4

cmpsd

Wpływa na flagi: OSZAPC

Instrukcja CMPS/CMPSQ

cmps qword ptr [(r|e)si], [(r|e)di]

cmpsq

Porównuje poczwórne słowo z pamięci ds:(r|e)si i z pamięci es:(r|e)di. Rejestry (r|e)di/(r|e)si są zwiększane/zmniejszane o 8 w zależności od flagi DF (0/1).

[ds:(r|e)si] - [es:(r|e)di]

(r|e)di := (r|e)di ± 8

(r|e)si := (r|e)si ± 8

cmpsq

Wpływa na flagi: OSZAPC

Instrukcja SCAS/SCASB

scas byte ptr [(r|e)di]

scasb

Porównuje bajt akumulatora AL i pamięci es:(r|e)di. Rejestr (r|e)di jest zwiększany/zmniejszany o 1 w zależności od flagi DF (0/1).

AL. - [es:(r|e)di]

(r|e)di := (r|e)di ± 1

scasb

Wpływa na flagi: OSZAPC

Instrukcja SCAS/SCASW

scas word ptr [(r|e)di]

scasw

Porównuje słowo akumulatora AX i pamięci es:(r|e)di. Rejestr (r|e)di jest zwiększany/zmniejszany o 2 w zależności od flagi DF (0/1).

AX-[es:(r|e)di]

(r|e)di:=(r|e)di ± 2

scasw

Wpływa na flagi: OSZAPC

Instrukcja SCAS/SCASD

scas dword ptr [(r|e)di]

scasd

Porównuje podwójne słowo akumulatora EAX i pamięci es:(r|e)di. Rejestr (r|e)di jest zwiększany/zmniejszany o 4 w zależności od flagi DF (0/1).

EAX - [es: (r|e)di]

(r|e)di := (r|e)di ± 4

scasd

Wpływa na flagi: OSZAPC

Instrukcja SCAS/SCASQ

scas qword ptr [(r|e)di]

scasq

Porównuje poczwórne słowo akumulatora RAX i pamięci es:(r|e)di. Rejestr (r|e)di jest zwiększany/zmniejszany o 8 w zależności od flagi DF (0/1).

RAX-[es:(r|e)di]

(r|e)di := (r|e)di ±8

scasq

Wpływa na flagi: -

Instrukcja LODS/LODSB

lods byte ptr [(r|e)si]

lodsb

Czyta bajt do akumulatora AL z pamięci ds:(r|e)si. Rejestr (r|e)si jest zwiększany/zmniejszany o 1 w zależności od flagi DF (0/1).

AL := [ds:(r|e)si]

(r|e)si := (r|e)si ±1

lodsb

Wpływa na flagi: -

Instrukcja LODS/LODSW

lods word ptr [(r|e)si]

lodsw

Czyta słowo do akumulatora AX z pamięci ds:(r|e)si. Rejestr (r|e)si jest zwiększany/zmniejszany o 2 w zależności od flagi DF (0/1).

AX = [ds:(r|e)si]

(r|e)si := (r|e)si ±2

lodsw

Wpływa na flagi: -

Instrukcja LODS/LODSD

lods dword ptr [(r|e)si]

lodsd

Czyta podwójne słowo do akumulatora EAX z pamięci ds:(r|e)si. Rejestr (r|e)si jest zwiększany/zmniejszany o 4 w zależności od flagi DF (0/1).

EAX = [ds:(r|e)si]

(r|e)si := (r|e)si ± 4

lodsd

Wpływa na flagi: -

Instrukcja LODS/LODSQ

lods qword ptr [(r|e)si]

lodsq

Czyta poczwórne słowo do akumulatora RAX z pamięci ds:(r|e)si. Rejestr (r|e)si jest zwiększany/zmniejszany o 8 w zależności od flagi DF (0/1).

RAX = [ds:(r|e)si]

(r|e)si := (r|e)si ± 8

lodsq

Wpływa na flagi: -

Instrukcja STOS/STOSB

stos byte ptr [(r|e)di]
stosb

Zapisuje bajt z akumulatora AL do pamięci es:(r|e)di. Rejestr (r|e)di jest zwiększany/zmniejszany o 1 w zależności od flagi DF (0/1).

[es:(r|e)di] := AL
(r|e)di:=(r|e)di ± 1

stosb

Wpływa na flagi: -

Instrukcja STOS/STOSW

stos word ptr [(r|e)di]
stosw

Zapisuje słowo z akumulatora AX do pamięci es:(r|e)di. Rejestr (r|e)di jest zwiększany/zmniejszany o 2 w zależności od flagi DF (0/1).

[es:(r|e)di] := AX
(r|e)di:=(r|e)di ± 2

stosw

Wpływa na flagi: -

Instrukcja STOS/STOSD

stos dword ptr [(r|e)di]
stosd

Zapisuje podwójne słowo z akumulatora EAX do pamięci es:(r|e)di. Rejestr (r|e)di jest zwiększany/zmniejszany o 4 w zależności od flagi DF (0/1).

[es:(r|e)di] := EAX
(r|e)di := (r|e)di ± 4

stosd

Wpływa na flagi: -

Instrukcja STOS/STOSQ

stos qword ptr [(r|e)di]
stosq

Zapisuje poczwórne słowo z akumulatora RAX do pamięci es:(r|e)di. Rejestr (r|e)di jest zwiększany/zmniejszany o 8 w zależności od flagi DF (0/1).

[es:(r|e)di]=RAX
(r|e)di:=(r|e)di ±8

stosq

Wpływa na flagi: -

Prefiks REP
REPNZ/REPNE
REPZ/REPE

Powoduje powtórzenie (R|E)CX razy następującej po nim instrukcji łańcuchowej, jeśli spełniony jest warunek (repnz powtarza dopóty ZF = 0,jeśli ZF = 1 powtarzanie jest przerywane itd.). Jeżeli (R|E)CX = 0, to instrukcja nie zostanie wykonana.

Wpływa na flagi: -

Prefiks REP
REPNZ/REPNE
REPZ/REPE

rep movsb

rep lodsd

rep stosq

rep*ww* cmpsw

rep*ww* scasb

Przykład

```
mov ecx, 100
mov esi, bufor1
mov edi, bufor2
rep movsb
```

Kopiuje zawartość bufora1 do bufora2.

```
mov rax, 0
mov rcx, 100
mov rdi, bufor
rep ds:stosq
```

Zeruje zawartość bufora (800B).

```
mov al, 77
mov ecx, 100
mov edi, bufor
repnz ds:scasb
```

Szuka wartości 77 w buforze. ZF = 1 oznacza znalezienie żądanej wartości.

```
mov al, 0
mov rcx, 100
mov rdi, bufor
repz ds:scasb
```

Szuka wartości <>0 w buforze. ZF = 0 oznacza znalezienie żądanej wartości.

Operacje na rejestrach segmentowych

- LDS Załadowanie pełnego wskaźnika z użyciem DS
- LES Załadowanie pełnego wskaźnika z użyciem ES
- LFS Załadowanie pełnego wskaźnika z użyciem FS
- LGS Załadowanie pełnego wskaźnika z użyciem GS
- LSS Załadowanie pełnego wskaźnika z użyciem SS

Wpływa na flagi: -

Instrukcja LDS

lds cel, źródło

Wczytanie pełnego adresu źródła do pary rejestrów ds:cel(32).

ds:cel := wskaźnik do źródła

```
lds esi, tablica
```


Wpływa na flagi: -

Instrukcja LES

les cel, źródło

Wczytanie pełnego adresu źródła do pary rejestrów es:cel(32).

es:cel := wskaźnik do źródła

```
les edi, tablica2
```


Wpływa na flagi: -

Instrukcja LFS

lfs cel, źródło

Wczytanie pełnego adresu źródła do pary rejestrów fs:cel.

fs:cel := wskaźnik do źródła

```
lfs eax, tablica
```


Wpływa na flagi: -

Instrukcja LGS

lgs cel, źródło

Wczytanie pełnego adresu źródła do pary rejestrów gs:cel.

gs:cel := wskaźnik do źródła

```
lgs eax, tablica
```

Wpływa na flagi: -

Instrukcja LSS

lss cel, źródło

Wczytanie pełnego adresu źródła do pary rejestrów ss:cel.

ss:cel := wskaźnik do źródła

lss esp, nowy_stos

Inne operacje

- LOCK — Powoduje niepodzielne wykonanie następnej instrukcji
- LEA — Ładowanie adresu efektywnego
- NOP — Nie wykonuje żadnego działania
- UD2 — Instrukcja niezdefiniowana
- XLAT/XLATB — Tłumaczenie w oparciu o tablicę translacji
- MOVBE — Przesłanie po zamianie kolejności bajtów
- CPUID — Identyfikacja procesora

Wpływa na flagi: -

Prefiks LOCK

lock

Powoduje wystawienie sygnału LOCK procesora i wykonanie w sposób niepodzielny instrukcji:

add, adc, and, brc, btr, bts, cmpxchg, cmpxch8b, cmpxch16b, dec, inc, neg, not, or, sbb, sub, xor, xadd i xchg,

<u>jeśli argument celu jest w pamięci</u>.

lock btr

Przykład

```
        mov     zu, 1   ;inicjalizacja
        ...
        ...
        xor     ax, ax  ;ax=0
@p:
lock    xchg    zu, ax  ;al<->zu
        test    ax, ax  ;czy 0
        jz      @p      ;akt. czekanie
        ...
        ...
        ...
        mov     zu, 1   ;oddaj 1 do zu
        ...
```

```
        mov     zu, 1
        ...
        ...
        ...
        ...
@p:
lock    btr     zu, 0   ;zajmij
        jnc     @p
        ...
        ...
        ...
        mov     zu, 1   ;uwolnij
        ...
```


Wpływa na flagi: -

Instrukcja LEA

lea cel, źródło

Wczytanie wyznaczonego adresu źródła do rejestru celu.

cel := adres źródła

```
lea eax, [edx+esi*4+12]     ; eax = edx + esi * 4 + 12
lea rax, [rdx+rsi*4+12]     ; rax = rdx + rsi * 4 + 12
```


Przykład

```
lea     rax, [rdx]          ;kopiuje rejestr
lea     rbx, [rcx + rdx]    ;suma rejestrów
lea     rdx, [rax +10]      ;suma rejestru i stałej
lea     rax, [rax +1]       ;inkrementacja
lea     rax, [rbx+8*rsi + 3]    ;suma trzech wartości
lea     rax, [rax + 4*rax]  ;mnożenie przez 5
```

Wpływa na flagi: -

Instrukcja NOP

nop

Nic nie robi.

nop

Wpływa na flagi: -

Instrukcja UD2

ud2

Generuje wyjątek *instrukcja niezdefiniowana*, nic nie robi, wprowadzona do testów.

ud2

Wpływa na flagi: -

Instrukcja XLAT/XLATB

xlat arg

xlatb

Tłumaczenie w oparciu o tablicę translacji.

AL := DS:[(R|E)BX+AL]

xlatb

Wpływa na flagi: -

Instrukcja MOVBE

movbe cel, źródło

Przesłanie po zamianie kolejności bajtów. Jeden z argumentów musi być rejestrem (16, 32, 64).

cel := zamień(źródło)

movbe eax, zmienna

przed	12	c4	7f	de
po	de	7f	c4	12

Rejestr flag

Rejestr flag w architekturze Intel x86			
bit	Skrót/wartość	opis	typ
0	CF	flaga przeniesienia (carry)	S
1	1	zarezerwowany	
2	PF	flaga parzystości (parity)	S
4	AF	flaga wyrównania (adjust)	S
6	ZF	flaga zera (zero)	S
7	SF	flaga znaku (sign)	S
8	TP	flaga umożliwiająca krokowe wykonanie (trap)	X
9	IF	flaga zezwolenia na przerwania (interrupt enable)	X
10	DF	flaga kierunku (direction)	C
11	OF	flaga przepełnienia (overflow)	S
12, 13	IOPL	poziom uprawnień we/wy (I/O privilege level, od 286)	X
14	NT	nested task flag (od 286)	X
16	RF	flaga wznowienia (resume, od 386)	X
17	VM	flaga trybu Virtual 8086 (od 386)	X
18	AC	alignment check (od 486SX)	X
19	VIF	Virtual interrupt flag (od Pentium)	X
20	VIP	Virtual interrupt pending (od Pentium)	X
21	ID	Identification (od Pentium)	X
3, 5, 15, 22-31	0	zarezerwowany	

S: Znacznik stanu
C: Znacznik kontrolny
X: Znacznik systemowy

Wpływa na flagi: -

Instrukcja CPUID

cpuid

Identyfikacja procesora jest możliwa, jeśli bit 21 flaga ID w rejestrze flag może być zmieniana. Na podstawie EAX (czasem też ECX) podaje w EAX,EBX,ECX i EDX różne informacje o procesorze.

cpuid

Test ID

```
        xor     eax, eax  ;0
        pusfd
        pop     edx
        bts     edx, 21   ;ustaw
        push    edx
        popfd
        pushfd
        pop     edx
        btr     edx, 21   ;skasuj
        rcl     eax, 1    ;1b
        push    edx
        popfd
        pushfd
        pop     edx
```

```
        bts     edx, 21   ;ustaw
        rcl     eax, 1    ;10b
        push    edx
        popfd
        pushfd
        pop     edx
        bt      edx, 21   ;sprawdź
        rcl     eax, 1    ;101b
        cmp     eax, 5
        je      cpuidOK
```

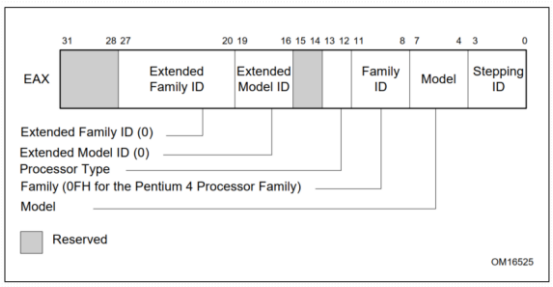

Przykład

```
mov eax, 0
cpuid
```

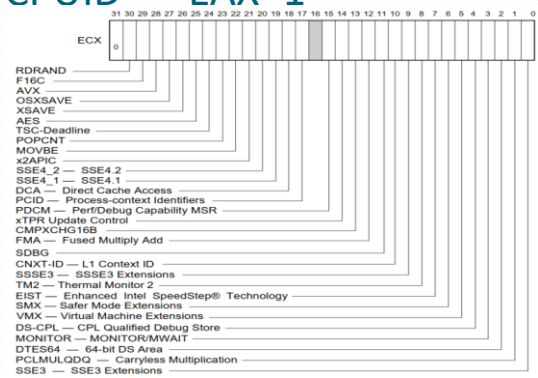
Zwraca wartość maksymalną dla cpuid
oraz identyfikator producenta:

eax = max
ebx = 'Genu'
ecx = 'ntel'
edx = 'ineI'

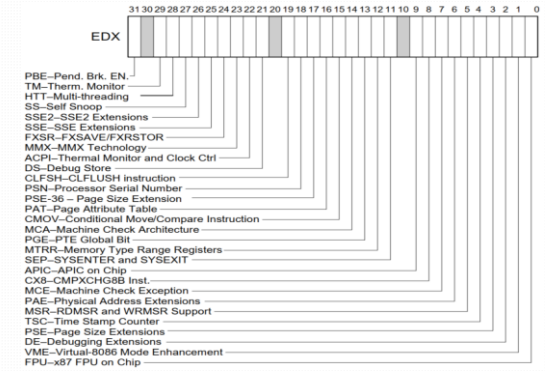
CPUID EAX=1



CPUID EAX=1



CPUID EAX=1



Przykład

Sprawdzenie, czy procesor posiada technologię MMX.

```
mov EAX, 1          ; żądanie informacji o właściwościach
CPUID               ; oFH, oA2H instrukcja CPUID
test EDX, oo8ooooooH ; sprawdzenie bitu technologii MMX (bit 23 w EDX)
jnz                 ; znaleziono technologię MMX
```

Wpływa na flagi: -

Instrukcja XGETBV

xgetbv

Czyta do edx:eax zawartość rozszerzonego rejestru kontrolnego o indeksie ecx.

xgetbv

Przykład - Sprawdzenie, czy procesor posiada technologię AVX.

```
supports_AVX proc
        mov     eax, 1
        cpuid
        and     ecx, 018000000H
        cmp     ecx, 018000000H  ; sprawdź flagi OSXSAVE i AVX
        jne     not_sup
        ; procesor wspiera inst. AVX i XGETBV jest włączone przez OS
        mov     ecx, 0           ; wybierz rejestr XCR0
        XGETBV                   ; wynik w EDX:EAX
        and     eax, 06H
        cmp     eax, 06H         ; czy OS włączył obsługę XMM i YMM
        jne     not_sup
        mov     eax, 1           ; wspiera AVX
        jmp     done
not_sup: mov    eax, 0           ; nie wspiera AVX
done:   ret
endp
```


XCR0