

Vybrané inštrukcie mikroprocesora Intel Pentium®

Použité skratky pre operandy inštrukcií

<i>operand</i>	ľubovoľný operand	<i>label</i>	návestie pre skok (kam skočiť)
<i>zdroj</i>	ľubovoľný operand	<i>m</i>	ľubovoľný pamäťový operand
<i>ciel'</i>	ľubovoľný operand	<i>m8</i>	8-bitový pamäťový operand
<i>r</i>	ľubovoľný všeobecný register	<i>m16</i>	16-bitový pamäťový operand
<i>r8</i>	8-bitový všeobecný register	<i>imm</i>	ľubovoľný priamy operand (číslo)
<i>r16</i>	16-bitový všeobecný register	<i>imm8</i>	8-bitový priamy operand (číslo)
<i>r32</i>	32-bitový všeobecný register	<i>imm16</i>	16-bitový priamy operand (číslo)
<i>r/m</i>	register alebo pamäťový operand	/	lomka znamená <i>alebo</i> (napr. <i>r16/r32</i>)
[...]	v hranatých zátvorkách je nepovinný operand		

Presuny údajov

MOV <i>ciel'</i> , <i>zdroj</i>	Prenesie zdrojový operand do cieľového, nemení register EFLAGS
XCHG <i>ciel'</i> , <i>zdroj</i>	Vymení obsah zdroja a cieľa. Aspoň jeden operand musí byť register
MOVSX <i>r16/r32</i> , <i>r8/r16/m8/m16</i>	Prvý operand musí byť dlhší než druhý. Ako MOV ale bity neobsiahnuté v prvom operande sa vyplnia znamienkovým bitom
MOVZX <i>r16/r32</i> , <i>r8/r16/m8/m16</i>	Prvý operand musí byť dlhší než druhý. Ako MOV ale bity neobsiahnuté v prvom operande sa vyplnia nulami.
LEA <i>r16/r32</i> , <i>m</i>	Do registra uloží offsetovú časť adresy zdroja.
LDS/LFS/LGS/LSS <i>r16/r32</i> , <i>m</i>	Lxx do <i>xx:r16/32</i> uloží celú adresu (segment:offset) uloženú v <i>m</i>
BSWAP <i>r32</i>	Vymení poradie bajtov v operande - prevod medzi little-endian a big-endian

Aritmetické inštrukcie

ADD <i>ciel'</i> , <i>zdroj</i>	Pripočíta <i>zdroj</i> k <i>ciel'u</i>
ADC <i>ciel'</i> , <i>zdroj</i>	Pripočíta <i>zdroj</i> a CF k <i>ciel'u</i>
INC <i>operand</i>	K <i>operandu</i> pripočíta 1, nezmení obsah CF
SUB <i>ciel'</i> , <i>zdroj</i>	Od <i>ciel'a</i> odčíta <i>zdroj</i>
SBB <i>ciel'</i> , <i>zdroj</i>	Od <i>ciel'a</i> odčíta CF a <i>zdroj</i>
DEC <i>operand</i>	Od <i>operandu</i> odčíta 1, nezmení obsah CF
IMUL <i>r</i> , <i>zdroj</i>	Register <i>r</i> <u>znamienkovo</u> vynásobí <i>zdrojom</i> ($r = r * zdroj$). OF a CF sa vynulujú ak sa výsledok vošiel do dĺžky <i>r</i> , inak sú oba nastavené na 1.
IMUL <i>r/m</i>	AL, AX alebo EAX (podľa dĺžky <i>r/m</i>) <u>znamienkovo</u> vynásobí <i>r/m</i> a výsledok uloží do AX, spojených DX:AX alebo spojených EDX:EAX.
MUL <i>r/m</i>	AL, AX alebo EAX (podľa dĺžky <i>r/m</i>) sa <u>neznamienkovo</u> vynásobí <i>r/m</i> a výsledok uloží do spojených AX, DX:AX alebo spojených EDX:EAX.
IDIV <i>r/m</i>	AX, DX:AX alebo EDX:EAX <u>znamienkovo</u> vydeli operandom a podiel uloží do AL, AX alebo EAX a zvyšok uloží do AH, DX alebo EDX
DIV <i>r/m</i>	AX, DX:AX alebo EDX:EAX <u>neznamienkovo</u> vydeli operandom a podiel uloží do AL, AX alebo EAX a zvyšok uloží do AH, DX alebo EDX
NEG <i>operand</i>	Zmeni číslo v <i>operande</i> na opačné (vynásobí ho -1)
CMP <i>ciel'</i> , <i>zdroj</i>	Podľa výsledku odčítania <i>ciel' - zdroj</i> nastaví bity v registri EFLAGS. Výsledok nikam neuloží

Logické inštrukcie

AND <i>ciel'</i> , <i>zdroj</i>	<i>ciel'</i> := <i>ciel' AND zdroj</i>
OR <i>ciel'</i> , <i>zdroj</i>	<i>ciel'</i> := <i>ciel' OR zdroj</i>
XOR <i>ciel'</i> , <i>zdroj</i>	<i>ciel'</i> := <i>ciel' XOR zdroj</i>
NOT <i>operand</i>	<i>operand</i> := NOT <i>operand</i>
TEST <i>ciel'</i> , <i>zdroj</i>	Podľa <i>ciel' AND zdroj</i> nastaví bity registra EFLAGS. Výsledok neuloží

Rotácie, posuny a práca s bitmi

Z druhého operandu sa berie do úvahy len spodných 5 bitov.

ROL <i>ciel'</i> , <i>imm8/CL</i>	Rotuje <i>ciel'</i> doľava o daný počet bitov, najvyšší bit kopíruje do CF
ROR <i>ciel'</i> , <i>imm8/CL</i>	Rotuje <i>ciel'</i> doprava o daný počet bitov, najnižší bit kopíruje do CF
RCL <i>ciel'</i> , <i>imm8/CL</i>	Rotuje <i>ciel'</i> spojený s CF o zadaný počet bitov doľava
RCR <i>ciel'</i> , <i>imm8/CL</i>	Rotuje <i>ciel'</i> spojený s CF o zadaný počet bitov doprava
SAL/SHL <i>ciel'</i> , <i>imm8/CL</i>	Posunie bity <i>ciel'a</i> o zadaný počet doľava, do spodných bitov prichádzajú nuly, znamienkový bit sa presunie do CF
SAR. <i>ciel'</i> , <i>imm8/CL</i>	Posunie aritmeticky (znamienkovo) bity <i>ciel'a</i> o zadaný počet doprava, znamienkový bit zostáva na mieste a kopíruje sa aj do nižšieho bitu. Najnižší bit sa presunie do CF
SHR. <i>ciel'</i> , <i>imm8/CL</i>	Posunie logicky (neznamienkovo) bity <i>ciel'a</i> o zadaný počet doprava, do znamienkového bitu príde 0, najnižší bit prejde do CF

Vetvenie programu, podmienky a cykly

JMP <i>label</i>	Nepodmienený skok
Jxxx <i>label</i>	Skočí ak je splnená podmienka xxx. Vid' nasledujúcu tabuľku.
SETxxx <i>r8/m8</i>	Do operandu uloží hodnotu 1 ak je podmienka xxx splnená, inak uloží 0.
CALL <i>operand</i>	Volanie podprogramu
RET [<i>imm16</i>]	Návrat z podprogramu volaného CALL a zvýšenie ESP o <i>imm16</i> ak je uvedený
LOOP <i>label</i>	od ECX odčíta 1 a ak je ECX ≠ 0 skočí na <i>label</i>

Skratka	Z anglického: Jump if	Po CMP skoč ak (je)	Po inej inštrukcii skoč ak výsledok (je)	Podmienka
JE/JZ	Equal / Zero	=	= 0	ZF = 1
JNE/JNZ	Not equal / Not zero	≠	≠ 0	ZF = 0
JP/JPE	Parity / Parity Even		má párny počet 1	PF = 1
JNP/JPO	Not parity / Parity odd		má nepárny počet 1	PF = 0
JS	Sign		má najvyšší bit 1; teda z. < 0	SF = 1
JNS	Not sign		má najvyšší bit 0; teda z. >= 0	SF = 0
JC	Carry		spôsobil prenos	CF = 1
JNC	Not Carry		nespôsobil prenos	CF = 0
JO	Overflow		pretiekol	OF = 1
JNO	Not overflow		nepretiekol	OF = 0
JB/JNAE	Below / Not above or equal	nz. <		CF = 1
JAE/JNB	Above or equal / Not below	nz. >=		CF = 0
JBE/JNA	Below or equal / Not above	nz. <=		(CF = 1) or (ZF = 1)
JA/JNBE	Above / Not below or equal	nz. >		(CF = 0) or (ZF = 0)
JL/JNGE	Less / Not greater or equal	z. <		SF ≠ OF
JGE/JNL	Greater or equal / Not less	z. >=		SF = OF
JLE/JNG	Less or equal / Not greater	z. <=		(ZF = 1) or (SF ≠ OF)
JG/JNLE	Greater / Not less or equal	z. >		(ZF = 0) or (SF = OF)
JCXZ	CX is zero			CX = 0
JECXZ	ECX is zero			ECX = 0

Narábanie so zásobníkom a s bitmi registra EFLAGS

PUSH <i>r32/m32/imm32</i>	Uloží operand do zásobníka
POP <i>r32/m32</i>	Vyberie operand zo zásobníka
PUSHFD	Do zásobníka uloží register EFLAGS
POPFD	Zo zásobníka vyberie register EFLAGS
STC	Nastaví CF na 1
CLC	Nastaví CF na 0
CMC	Neguje CF