PRENOSNÉ INŠTRUKCIE

MOV reg/m8-64b ← r/m/imm8-64b

VMOVUPS YWORD32B/DQWORD16B YWORD32B/DQWORD16B
VMOVUPD ymm/m xmm/m ← ymm/m xmm/m
VMOVDQU

VLDDQU ymm xmm ← m256b m128b

VMOVAPS YWORD32B/DQWORD16B YWORD32B/DQWORD16B
VMOVAPD ymm/m xmm/m ← ymm/m xmm/m
VMOVDQA mem ALIGNED 32/16B (adresa v pamäti musí byť násobok 32/16B)

VMOVDDUP 4x2xQWORD 8B 64b ymm xmm ← 2x1xQWORD 8B 64b ymm/m xmm/m

VMOVLHPS QWORD 8B 64b xmm ← QWORD 8B 64b xmm
└3operands:

VMOVHLPS QWORD 8B 64b xmm ← QWORD 8B 64b xmm
└3operands:

VMOVLPS QWORD 8B 64b xmm m64b ← QWORD 8B 64b xmm m64b
VMOVLPD
└3operands:

VMOVHPS QWORD 8B 64b xmm m64b ← QWORD 8B 64b xmm m64b
VMOVHPD
└3operands:

MOVD ZERO DWORD 4B 32b xmm mmx/m32 ← DWORD 4B 32b xmm mmx/m32
MOVSS

MOVQ ZERO QWORD 8B 64b xmm mmx/m64 ← QWORD 8B 64b xmm mmx/m64
MOVSD

MOVQ2DQ ZERO QWORD 8B xmm ← QWORD 8B mmx64b

MOVDQ2Q QWORD 8B mmx64b ← QWORD 8B xmm

VMOVSHDUP 8x4xDWORD 4B 32b ymm xmm ← 255-224 191-160 127-96 63-32 ymm/m xmm/m

VMOVSLDUP 8x4xDWORD 4B 32b ymm xmm ← 223-192 159-128 95-64 31-0 ymm/m xmm/m
mem ALIGNED 16B(adresa v pamäti musí byť násobok 32/16B)

VMOVMSKPD 00 00 4b/2b reg32/64 ← ONLY 4x 2x 63.BITs QWORD ymm xmm
VMOVMSKPS 8b/4b ONLY 8x 4x 31.BITs WORD

VPMOVMSKB 00 32b 16b 8b reg32/64 ← 32x 16x 8x ONLY 7.BITs BYTE ymm xmm mmx

MOVNTI QWORD/DWORD 8/4B m32/64 ← reg32/64

MOVNTQ QWORD 8B 64b m64b ← mmx64b

VMOVNTPS YWORD32B/DQWORD16B m256 m128 ← ymm xmm
VMOVNTPD
VMOVNTDQ mem ALIGNED 32/16B

VMOVNTDQA YWORD32B/DQWORD16B ymm xmm ← mem ALIGNED 32/16B m256 m128

BIT	FLAG	r16-64,r/m16-64 1	0
6	ZF	CMOVZ CMOVE	CMOVNZ CMOVNE
0	CF	CMOVC CMOVB CMOVNAE	CMOVNC CMOVNB CMOVAE
7	SF	CMOVS	CMOVNS
2	PF	CMOVP CMOVPE	CMOVNP CMOVPO
11	OF	CMOVO	CMOVNO
CF or ZF =		CMOVBE	CMOVNA
CF and ZF =		CMOVA	CMOVNBE
SF=OF ZF=		CMOVG	CMOVNLE
SF<>OF ZF=		CMOVLE	CMOVNG
SF=OF		CMOVGE CMOVNL	
SF<>OF		CMOVL CMOVNGE	

SHUFPS xmm,xmm/m128,mask8b — 4xDWORD 4B 32b
SHUFPD xmm,xmm/m128,mask2b — 2xQWORD 8B 64b

PSHUFB xmm/mmx,r/m64-128 — 16x8xBYTE 8b, 80h=00, 0-7/15

PSHUFW mmx,r/m64,mask8b — 4xWORD 2B 16b
PSHUFD xmm,xmm/m128,mask8b — 4xDWORD 4B 32b

VSHUFPD 4x 2x QWORD 8B 64b
VSHUFPS 8x 4x DWORD 4B 32b
MASK4/2b, MASK8b, i8/4/2b

PSHUFHW xmm,xmm/m128,mask8b — 4xWORD 2B 16b
PSHUFLW xmm,xmm/m128,mask8b — 4xWORD 2B

VPSHUFB 32x 16x BYTE ymm xmm ← 15 015 0 ymm xmm, 32x 16xBYTE 0-15 (4b) 80h=00 ymm/m xmm/m

MOVZX
MOVSXD SIGN (znamienkové rozšírenie) UN/SIGN DWORD/WORD reg.16/32/64b ← UN/SIGN WORD / BYTE reg/m8/16/32b

VPMOVZXBW UNS/SIGN16x8xWORD ymm/xmm ← UNS/SIGN 16x8xBYTE xmm/m64128
VPMOVSXBW

VPMOVZXBD UNS/SIGN8x4xDWORD ymm/xmm ← UNS/SIGN 8x4xBYTE xmm/m3264
VPMOVSXBD

VPMOVZXBQ UNS/SIGN4x2xQWORD ymm/xmm ← UNS/SIGN 4x2xBYTE xmm/m1632
VPMOVSXBQ

VPMOVZXWD UNS/SIGN8x4xDWORD ymm/xmm ← UNS/SIGN8x4xWORD 2B xmm/m64128
VPMOVSXWD

VPMOVZXWQ UNS/SIGN4x2xQWORD ymm/xmm ← UNS/SIGN4x2xWORD 2B xmm/m3264
VPMOVSXWQ

VPMOVZXDQ UNS/SIGN4x2xQWORD ymm/xmm ← UNS/SIGN4x2xDWORD 4B xmm/m64128
VPMOVSXDQ

BLENDPD 2x QWORD 8B 64b xmm ← 10b 01b xmm/m128 ,00b

BLENDPS 4x DWORD 4B 32b xmm ← 4x DWORD 4B 32b xmm/m128 ,i4b

PBLENDW 8x WORD 2B 16b xmm ← 1 WORD xmm/m128 ,i8b

VPBLENDW 16x WORD 2B 16b ymm xmm ← 0 WORD ymm xmm ← 1 WORD ymm xmm ,i8b

VPBLENDD 8x DWORD 2B 16b ymm xmm ← 0 DWORD ymm xmm ← 1 DWORD ymm xmm ,i8/4b

VBLENDVPD 4x 2x QWORD 8B 64b ymm xmm ← 0 ymm xmm ← 1 ymm/m xmm/m ← 4x 2xSIGN 63.BITs MASK ymm xmm
VBLENDVPS 8x 4x DWORD 4B 32b ← 0 ← 1 ← 8x 4xSIGN 31.BITs MASK

VPBLENDVB 32x 16x BYTE ymm xmm ← 0 ymm xmm ← 1 ymm/m xmm/m ← 32x 16xSIGN 7.BITs MASK ymm xmm

(SKOPÍRUJE ZDROJOVÝ KTORÝ JE V XMM0 SIGN=1)
BLENDVPD 2x QWORD xmm ← 2x QWORD xmm/[r/m] ← SIGN 63.BITs MASK XMM0
BLENDVPS 4x DWORD ← 4x DWORD ← SIGN 31.BITs MASK
PBLENDVB 16x BYTE xmm ← 16x BYTE xmm/[r/m] ← SIGN 7.BITs MASK XMM0

nairam
BEGIN: 07.01.2010

EXAMPLE: ESI=EBX+8*EAX+4
LEA ESI, [EBX + 8*EAX + 4]

LAHF SAHF 7 AH 0 ↔ [SP]; inc/dec SP
POP PUSH src reg.16-64/imm8-32/seg. ↔ [SP]; inc/dec SP
POP PUSHQ src reg.64/ SIGNED 63 32 31 imm32b 0 ↔ [SP]; inc/dec SP
POP PUSHA AX,BX,CX,DX,BP,SI,DI ↔ [SP]; inc/dec SP
POP PUSHAD EAX,EBX,ECX,EDX,EBP,ESI,EDI ↔ [SP]; inc/dec SP
POP PUSHFW 15 FLAGS 0 ↔ [SP]; inc/dec SP
POP PUSHF/FD 31 EFLAGS 0 ↔ [SP]; inc/dec SP
POP PUSHF/FQ 63 RFLAGS 0 ↔ [SP]; inc/dec SP

CBW SIGNED 15 AX 0 ← 7 AL 0
CWD 31 DX 16 : 15 AX 0 ← 15 AX 0
CWDE 31 EAX 0 ← 15 AX 0
CDQ 63 EDX 32 : 31 EAX 0 ← 31 EAX 0
CDQE 63 RAX 0 ← 31 EAX 0
CQO 127 RDX 64 : 63 RAX 0 ← 63 RAX 0

VPEXTRB 00 BYTE r32/64 m8 ← 15 7 0 xmm ,0-15
VPINSRB

VPEXTRW 00 WORD 2B r32/64 m16 ← 7 6 5 4 3 2 1 0 xmm mmx ,0-7
VPINSRW

VPEXTRD 00 DWORD 4B r32/64 m32 ← 3 2 1 0 xmm ,0-3
VPINSRD

VPEXTRQ QWORD 8B 64b r64/m64 ← 1 0 xmm ,0-1
VPINSRQ

XCHG r/m8-64 ⇄ r/m8-64
XLATB AL 8b ← DS:[EBX+AL]

MOVBE 8x 4x 2x BYTE 33 22 11 00 mem/reg.16-64b ← 8x 4x 2x BYTE 00 11 22 33 reg./mem16-64b

BSWAP reg32/64

MASKMOVQ mmx, mmx 16x 8x BYTE m[ES:EDI] ← 16x 8x BYTE xmm mmx, SIGN 7.BITs MASK xmm mmx
MASKMOVDQU xmm, xmm

VPMASKMOVQ
VMASKMOVPD 4x 2x QWORD 8B 64b ymm/m xmm/m ← mask in mem: 1=copy / 0=fill $0 ymm xmm, 4x 2xSIGN 63.BITs MASK ymm/m xmm/m
VMASKMOVPS ymm/xmm,ymm/xmm,m m,ymm/xmm,ymm/xmm 8x 4x DWORD 4B 32b 8x 4xSIGN 31.BITs MASK
VPMASKMOVD

INSERTPS 4xDWORD 4B xmm ← 4xDWORD 4B xmm/m128 ← ODKIAĽ KAM VYPLŇ NULAMI 7 6 5 4 3 2 1 0 $80 $40 $20 $10 $08 $04 $02 $01 2 1 2 1 1 1 1 1 $0 if bit=1

VINSERTF128 2x DQWORD 16B 128b ymm256 32B ← DQWORD 16B ymm256 32B ← DQWORD 16B xmm/m128 ,0
VINSERTI128 F.P. / INTEGER ymm256 32B ← ymm256 32B ← xmm/m128 ,1

EXTRACTPS 00 DWORD 4B 32b r64/m32b ← DWORD 4B 32b xmm128 ,0-3

VEXTRACTF128 xmm/m128 ← 1 0 ymm256 32B ,0-1
VEXTRACTI128 F.P. / INTEGER DQWORD 16B 128b DQWORD 16B 128b