LOGICKÉ INŠTRUKCIE

CMP FLAGS ← r/m8-32 ↔ r/m8-32
(porovnanie dst - src)

TEST FLAGS ← r/m8-32 ↔ r/m8-32
(bitový súčin AND)

NOT r/m8-32

10001011 → 01110100

NEG r/m8-32 (NOT + 1)

PTEST xmm, xmm

Z=1 ŽIADNY BIT ROVNAKÝ
Z=0 NEJAKÝ BIT ROVNAKÝ
C=1 xmm = xmm
C=0 xmm != xmm

RESULT:
FF = TRUE
00 = FALSE

CMPPD xmm,xmm/m128,imm8 | CMPPS xmm,xmm/m128,imm8 | CMPSD xmm,xmm/m128,imm8 | CMPSS xmm,xmm/m128,imm8

CMPEQPD xmm, xmm/m128 — 2x F.P. QWORD 8B 64b — xmm/m128 → xmm → xmm 000000 / FFFFFF — 2xQWORD 8B 64b

CMPEQPS xmm, xmm/m128 — 4x F.P. DWORD 4B 32b — xmm/m128 → xmm → xmm 000000 / FFFFFF — 4xDWORD 4B 32b

CMPEQSD xmm, xmm/m64 — F.P. QWORD 8B 64b — xmm/m64 → xmm → xmm 000000 / FFFFFF — QWORD 8B 64b

CMPEQSS xmm, xmm/m32 — F.P. DWORD 4B 32b — xmm/m32 → xmm → xmm 000000 / FFFFFF — DWORD 4B 32b

0 CMPEQxx xmm = xmm/m32
1 CMPLTxx xmm < xmm/m32
2 CMPLExx xmm <= xmm/m32
3 CMPORDxx xmm ord. xmm/m32
4 CMPNEQxx xmm != xmm/m32
5 CMPNLTxx xmm !< xmm/m32
6 CMPNLExx xmm !<= xmm/m32
7 CMPUNORDxx xmm unord. xmm/m32

PCMPGTQ PCMPEQQ xmm,xmm/m128 — 2x SIGN.QWORD 8B 64b — xmm/m128 → xmm → xmm 000000 / FFFFFF — 2x QWORD 8B 64b

PCMPGTD PCMPEQD xmm/mmx,xmm/mmx/m — 4/2x SIG.DWORD 4B 32b — xmm mmx → xmm mmx → xmm mmx 00/FF — 4x 2x DWORD 4B 32b

PCMPGTW PCMPEQW xmm/mmx,xmm/mmx/m — 8/4x SIGN.WORD 2B 16b — xmm mmx → xmm mmx → xmm mmx 00/FF — 8x 4x WORD 2B 16b

PCMPGTB PCMPEQB xmm/mmx,xmm/mmx/m — 16x 8x SIGN. BYTE 8b — xmm mmx → xmm mmx → xmm mmx 00/FF — 16x 8x BYTE 8b

PCMPEQx: FF = rovnaké, 00 = nerovnaké

PCMPGTx: FF = xmm,mmx > xmm,mmx,m; 00 = xmm,mmx <= xmm,mmx,m

COMISD UCOMISD xmm, xmm/m64 — F.P. QWORD 8B 64b — xmm/m64 → xmm

COMISS UCOMISS xmm, xmm/m32 — F.P. DWORD 4B 32b — xmm/m32 → xmm

FLAGS	ZF	PF	CF
UNORDERED	1	1	1
xmm > xmm/m	0	0	0
xmm < xmm/m	0	0	1
xmm == xmm/m	1	0	0

CMPXCHG r/m8-32, r8-32

r/m8-32 == EAX AX AL — ZF=1 — r/m8-32 ← r8-32

r/m8-32 != EAX AX AL — ZF=0 — EAX AX AL ← r/m8-32

CMPXCHG8B [r]/m64

[r]/m64 == EDX:EAX — ZF=1 — [r]/m64 ← ECX:EBX

[r]/m64 != EDX:EAX — ZF=0 — [r]/m64 ← EDX:EAX

MAXPD MINPD xmm, xmm/m128 — 2xSIGN QWORD 8B 64b — xmm/m128 (1.0000000 | 2.0000000) → xmm (3.0000000 | 1.0000000) → xmm (3.0000000 | 2.0000000) — 2xSIGN QWORD 8B 64b

MAXPS MINPS xmm, xmm/m128 — 4xSIGN DWORD 4B 32b — xmm/m128 (-2.00 | 0.00 | 3.00 | 2.00) → xmm (1.00 | -1.00 | 2.00 | -2.00) → xmm (1.00 | 0.00 | 3.00 | 2.00) — 4xSIGN DWORD 4B 32b

MAXSD MINSD xmm, xmm/m64 — SIGN QWORD 8B 64b — xmm/m64 (-4.00) → xmm (2.00) → xmm (2.00) — SIGN QWORD 8B 64b

MAXSS MINSS xmm, xmm/m32 — SIGN DWORD 4B 32b — xmm/m32 (1.00) → xmm (2.00) → xmm (2.00) — SIGN DWORD 4B 32b

(DOSADÍ VYŠŠIU ALEBO NIŽŠIU HODNOTU)

PMAXSW PMINSW mmx/xmm,r/m64/128 — 8x4xSIGN WORD 2B 16b — r/m128 r/m64 → xmm mmx → xmm mmx → xmm mmx — 8x4xSIGN WORD 2B 16b

PMAXUB PMINUB mmx/xmm,r/m64/128 — 16x 8x BYTE 8b — r/m128 r/m64 → xmm mmx → xmm mmx → xmm mmx — 16x 8x BYTE 8b

LOCK CMPXCHG*
PREFIX MULTIPROCESOR LOCK

CMPXCHG16B [r]/m128

[r]/m128 == RDX:RAX — ZF=1 — [r]/m128 ← RCX:RBX

[r]/m128 != RDX:RAX — ZF=0 — [r]/m128 ← RDX:RAX

RIADIACE INŠTRUKCIE

CMC	invertuje CF
CLC	CF = 0
STC	CF = 1
CLD	DF = 0 smer hore +
STD	DF = 1 smer dole -
CLI	IF = 0 zakaz prerusenia
STI	IF = 1 povolenie prerusen
CLTS	reg.CR0(MSW-286) 3-bit.TS=0

OR r/m8-32,r/m16-32/imm8-16 — 11001100
ORPD xmm,r/m128 — 10001011
ORPS — OR
POR mmx/xmm,r/m64-128 — 11001111

XOR r/m8-32,r/m16-32/imm8-16 — 11110000
XORPD xmm,r/m128 — 00111100
XORPS — XOR
PXOR mmx/xmm,r/m64-128 — 11001100

AND r/m8-32,r/m/imm8-32

ANDPD — 11001100
ANDPS xmm,r/m128 — 10001011
AND
PAND mmx,r/m64 — 10001000

ANDNPD — 01001100
ANDNPS xmm,r/m128 — NOT — 10110011
10001011
PANDN mmx,r/m64 — AND — 10000011

BLOKOVÉ INŠTRUKCIE

				DF=0	DF=1
CMPSB FLAGS←	BYTE DS:[ESI]	↔	BYTE ES:[EDI]	ESI,EDI	+1	-1
CMPSW FLAGS←	WORD 2B DS:[ESI]	↔	WORD 2B ES:[EDI]	ESI,EDI	+2	-2
CMPSD FLAGS←	DWORD 4B DS:[ESI]	↔	DWORD 4B ES:[EDI]	ESI,EDI	+4	-4
CMPSQ FLAGS←	QWORD 8B [RSI]	↔	QWORD 8B [RDI]	RSI,RDI	+8	-8
MOVSB	BYTE ES:[EDI]	←	BYTE DS:[ESI]	EDI,ESI	+1	-1
MOVSW	WORD 2B ES:[EDI]	←	WORD 2B DS:[ESI]	EDI,ESI	+2	-2
MOVSD	DWORD 4B ES:[EDI]	←	DWORD 4B DS:[ESI]	EDI,ESI	+4	-4
MOVSQ	QWORD 8B [RDI]	←	QWORD 8B [RSI]	RDI,RSI	+8	-8
SCASB FLAGS←	BYTE ES:[EDI]	—	BYTE AL	EDI	+1	-1
SCASW FLAGS←	WORD 2B ES:[EDI]	—	WORD 2B AX	EDI	+2	-2
SCASD FLAGS←	DWORD 4B ES:[EDI]	—	DWORD 4B EAX	EDI	+4	-4
SCASQ FLAGS←	QWORD 8B [RDI]	—	QWORD 8B RAX	RDI	+8	-8
LODSB	BYTE AL	←	BYTE DS:[ESI]	ESI	+1	-1
LODSW	WORD 2B AX	←	WORD 2B DS:[ESI]	ESI	+2	-2
LODSD	DWORD 4B EAX	←	DWORD 4B DS:[ESI]	ESI	+4	-4
LODSQ	QWORD 8B RAX	←	QWORD 8B [RSI]	RSI	+8	-8
STOSB	BYTE ES:[EDI]	←	BYTE AL	EDI	+1	-1
STOSW	WORD 2B ES:[EDI]	←	WORD 2B AX	EDI	+2	-2
STOSD	DWORD 4B ES:[EDI]	←	DWORD 4B EAX	EDI	+4	-4
STOSQ	QWORD 8B [RDI]	←	QWORD 8B RAX	RDI	+8	-8

INSB blokove IN bajtu/word/dword
INSW ES:[DI],port[DX]
INSD if DF=0 then DI=DI+n else DI=DI-n

OUTSB blokove OUT bajtu/word/dword
OUTSW port[DX],ES:[DI]
OUTSD if DF=0 then DI=DI+n else DI=DI-n

PREFIXY OPAKOVANIA ECX-1

REP	INSx MOVSx OUTSx LODSx STOSx	ECX=0
REPE,REPZ	CMPSx SCASx	ECX!=0 or ZF=1
REPNE,REPNZ	CMPSx SCASx	ECX!=0 or ZF=0

dosadi hodnotu	1	0
SETA dst SETNBE dst	(CF==0)&&(ZF==0)	
SETAE dst SETNB dst SETNC dst	CF==0	
SETB dst SETC dst SETNAE dst	CF==1	
SETBE dst SETNA dst	(CF==1)||(ZF==1)	
SETE dst SETZ dst	ZF==1	
SETG dst SETNLE dst	(ZF==0)&&(SF==OF)	
SETGE dst SETNL dst	SF==OF.	
SETL dst SETNGE dst	(ZF==0)&&(SF<>OF)	
SETLE dst	SF<>OF	
SETNE dst SETNZ dst	ZF==0	
SETNG dst	(ZF==1)&&(SF<>OF)	
SETNO dst	OF==0	
SETNP dst SETPO dst	PF==0	
SETNS dst	SF==0	
SETO dst	OF==1	
SETP dst SETPE dst	PF==1	
SETS dst	SF==1	

SKOKOVÉ INŠTRUKCIE

BIT	FLAG	1	0
6	ZF	JZ JE	JNZ JNE
0	CF	JC JB JNAE	JNC JNB JAE
7	SF	JS	JNS
2	PF	JP JPE	JNP JPO
11	OF	JO	JNO
CF or ZF =		JBE JNA	JA JNBE
SF xor OF =		JL JNGE	JNL JGE
(SF xor OF) or ZF		JNG JLE	JG JNLE
CX = 0		JCXZ	
ECX = 0		JECXZ	

LOOP		(E)CX-- (E)CX!=0
LOOPZ	LOOPE	(E)CX-- (E)CX!=0 && ZF=1
LOOPNZ	LOOPNE	(E)CX-- (E)CX!=0 && ZF=0
LOOPD		ECX-- ECX!=0
LOOPDE	LOOPDZ	ECX-- ECX!=0 && CF=1
LOOPDNE	LOOPDNZ	ECX-- ECX!=0 && CF=0

JMP	SKOK NA ADRESU
CALL	CS,IP->(SP) JMP ADRESA
INT	F->(SP) CS,IP->(SP) JMP VECTOR
INTO	INT4 PRETEČENIE AK OF=1
IRET	NÁVRAT Z INT
RET	NÁVRAT IP<-(SP)
RETN	NÁVRAT (E)IP<-(SP)
RETF	NÁVRAT CS:EIP<-(SP)
RETQ	NÁVRAT RIP
RSM	NÁVRAT Z MÓDU

nairam
BEGIN: 16.01.2010

PREFIXY ZMENY SEGMENTU

SEGCS	SEGDS	SEGES
SEGFS	SEGGS	SEGSS

BITOVÉ INŠTRUKCIE

reg/m16/32/64,reg16/32/64/imm8

BT bit určený src z dst → CF
BTC bit určený src z dst → CF; bit v dst=neg.
BTR bit určený src z dst → CF; bit v dst=0
BTS bit určený src z dst → CF; bit v dst=1

TZCNT reg16/32/64,reg/m16/32/64
BSF — 7 6 5 4 3 2 1 0 — 0 0 0 1 1 1 1 0 b — reg.=1; ZF=1
BSR — 7 6 5 4 3 2 1 0 — reg.=4; ZF=1
LZCNT
(zapíše pozíciu nenulového. bitu)

POPCNT reg16/32/64, r/m16/32/64 — 7 6 5 4 3 2 1 0 — 0 0 0 1 1 1 1 0 b
(zapíše počet nenulových bitov) → reg16/32/64 = 4