Spisak instrukcija za µP iAPX86

Sintaksa	Opis	Flags		Br. ciklusa
1. Data transfer		ODITSZAPC		
MOV dst, src	Move	- - - - - - - - -	MOV mem, acc	10
			MOV acc, mem	10
			MOV reg, reg	2
			MOV reg, mem	8+EA
			MOV mem, reg	9+EA
			MOV reg, imd	4
			MOV mem, imd	10+EA
			MOV acc, mem	2
PUSH src	Push word onto stack	- - - - - - - - -	PUSH reg	11
			PUSH sreg(#CS)	10
			PUSH mem	16+EA
POP dst	Pop word off stack	- - - - - - - - -	POP reg	8
			POP sreg(#CS)	8
			POP mem	17+EA
XCHG dst, src	Exchange	- - - - - - - - -	XCHG acc, reg16	3
			XCHG mem, reg	17+EA
			XCHG reg, reg	4
XLAT src-table	Translate	- - - - - - - - -	XLAT src-table	11
IN acc, port	Input byte or word	- - - - - - - - -	IN acc, imd8	10
		- - - - - - - - -	IN acc, DX	8
OUT port, acc	Output byte or word		OUT imd8, acc	10
			OUT DX, acc	8
LDS dst, src	Load pointer using DS	- - - - - - - - -	LDS reg16, mem32	16+EA
LEA dst, src	Load effective address	- - - - - - - - -	LEA reg16, mem16	2+EA
LES dst, src	Load pointer using ES	- - - - - - - - -	LES reg16, mem32	16+EA
LAHF	Load AH from flags		LAHF	4
SAHF	Store AH into flags	- - - -RRRRR	SAHF	4
POPF	Pop flags off stack	RRRRRRRRR	POPF	8
PUSHF	Push flags onto stack	- - - - - - - - -	PUSHF	10
2. Arithmetic instructions				
ADD dst, src	Addition	X- - -XXXXX	ADD reg, reg	3
			ADD reg, mem	9+EA
			ADD mem, reg	16+EA
			ADD reg, imd	4
			ADD mem, imd	17+EA
			ADD acc, imd	4
ADC dst, src	Add with carry	X- - -XXXXX	ADC reg, reg	3
			ADC reg, mem	9+EA
			ADC mem, reg	16+EA
			ADC reg, imd	4
			ADC mem, imd	17+EA
			ADC acc, imd	4
INC dst	Increment by 1	X- - -XXXX-	INC reg16	2
			INC reg8	3
			INC mem	15+EA
DAA	Decimal adjust for addition	X- - -XXXXX	DAA	4
AAA	ASCII adjust for addition	U- - -UUXUX	AAA	4
SUB dst, src	Subtraction	X- - -XXXXX	SUB reg, reg	3
			SUB reg, mem	9+EA
			SUB mem, reg	16+EA
			SUB reg, imd	4
			SUB mem, imd	17+EA
			SUB acc, imd	4
SBB dst, src	Subtract with borrow	X- - -XXXXX	SBB reg, reg	3
			SBB reg, mem	9+EA
			SBB mem, reg	16+EA
			SBB reg, imd	4
			SBB mem, imd	17+EA
			SBB acc, imd	4
DEC dst	Decrement by one	X- - -XXXX-	DEC reg16	2
			DEC reg8	3
			DEC mem	15+EA

NEG dst	Negate	X- - -XXXXU	NEG reg	3
			NEG mem	16+EA
CMP dst, src	Compare destination to source	X- - -XXXXX	CMP reg, reg	3
			CMP reg, mem	9+EA
			CMP mem, reg	9+EA
			CMP reg, imd	4
			CMP mem, imd	10+EA
			CMP acc, imd	4
AAS	ASCII adjust for subtraction	U- - -UUXUX	AAS	4
DAS	Decimal adjust for subtraction	U - - -XXXXX	DAS	4
MUL src	Multiplication, unsigned	X - UUUUX	MUL reg8	70-77
			MUL reg16	118-133
			MUL mem8	(76-83)+EA
			MUL mem16	(124-139)+EA
IMUL src	Integer multiplication	X - UUUUX	IMUL reg8	80-98
			IMUL reg16	128-154
			IMUL mem8	(86-104)+EA
			IMUL mem16	(134-160)+EA
AAM	ASCII adjust for multiply	U - - -UUXUX	AAM	83
DIV src	Division unsigned	U- - -UUUUU	DIV reg8	80-90
			DIV reg16	144-162
			DIV mem8	(86-96)+EA
			DIV mem16	(158-168)+EA
IDIV src	Integer division	U- - -UUUUU	IDIV reg8	101-112
			IDIV reg16	165-184
			IDIV mem8	(107-118)+EA
			IDIV mem16	(171-190)+EA
AAD	ASCII adjust for division	U- - -UUXUX	AAD	80
CBW	Convert byte to word	- - - - - - - - -	CBW	2
CWD	Convert word to doubleword	- - - - - - - - -	CWD	5
3. Bit manipulation				
NOT dst	Logical not	- - - - - - - - -	NOT reg	3
			NOT mem	16+EA
AND dst, src	Logical and	0- - -XXUX0	AND reg, reg	3
			AND reg, imd	4
			AND reg, mem	9+EA
			AND mem, reg	16+EA
			AND mem, imd	17+EA
			AND acc, imd	4
OR dst, src	Logical or	0 - - -XXUX0	OR reg, mem	9+EA
			OR mem, reg	16+EA
			OR acc, imd	4
			OR reg, imd	4
			OR mem, imd	17+EA
XOR dst, src	Logical exclusive or	0- - 0XXUX0	XOR reg, reg	3
			XOR reg, mem	9+EA
			XOR mem, reg	16+EA
			XOR acc, imd	4
			XOR reg, imd	4
			XOR mem, imd	17+EA
TEST dst, src	Test	0 - - -XXUX0	TEST reg, reg	3
			TEST reg, mem	9+EA
			TEST acc, imd	4
			TEST reg, imd	4
			TEST mem, imd	17+EA
SAL dst, count / SHL dst, count	Shift arithmetic/logical left (synonims)	X- - - - - - -X	SAL reg, 1	2
			SHL reg, imd8	5+1/bit
			SAL reg, CL	8+4/bit
			SHL mem, 1	15+EA
			SHL mem, imd8	17+1/bit
			SAL mem, CL	20+EA+4/bit
SHR dst, count	Shift logical right	X- - - - - - -X	SHR reg, 1	2
			SHR reg, CL	8+4/bit
			SHR reg, imd8	5+1/bit
			SHR mem, 1	15+EA
			SHR mem, imd8	17+1/bit
			SHR mem, CL	20+EA+4/bit

SAR dst, count	Shift arithmetic right	X- - - - - - -X	SAR reg, 1	2
			SAR reg, imd8	5+1/bit
			SAR reg, CL	8+4/bit
			SAR mem, 1	15+EA
			SAR mem, imd8	17+1/bit
			SAR mem, CL	20+EA+4/bit
ROL dst, count	Rotate left	X- - - - - - -X	ROL reg, 1	2
			ROL reg, imd8	5+1/bit
			ROL reg, CL	8+4/bit
			ROL mem, 1	15+EA
			ROL mem, imd8	17+1/bit
			ROL mem, CL	20+EA+4/bit
ROR dst, count	Rotate right	X- - - - - - -X	ROR reg, 1	2
			ROR reg, imd8	5+1/bit
			ROR reg, CL	8+4/bit
			ROR mem, 1	15+EA
			ROR mem, imd8	17+1/bit
			ROR mem, CL	20+EA+4/bit
RCL dst, count	Rotate left trough carry	X- - - - - - -X	RCL reg, 1	2
			RCL reg, imd8	5+1/bit
			RCL reg, CL	8+4/bit
			RCL mem, 1	15+EA
			RCL mem, imd8	17+1/bit
			RCL mem, CL	20+EA+4/bit
RCR dst, count	Rotate right trough carry	X- - - - - - -X	RCR reg, 1	2
			RCR reg, imd8	5+1/bit
			RCR reg, CL	8+4/bit
			RCR mem, 1	15+EA
			RCR mem, imd8	17+1/bit
			RCR mem, CL	20+EA+4/bit
4. String manipulation				
REP	Repeat string operation	- - - - - - - - -		
REPE/REPZ	Repeat string operation while equal/ zero			
REPNE/REPNZ	Repeat string operation while not equal/ not zero			
MOVS dss, srs	Move string	- - - - - - - - -	MOVS dss, srs	18
		REP	MOVS dss, srs	2+17/rep
MOVSB/	Move string (byte / word)	- - - - - - - - -	MOVSB	18
MOVSW		REP	MOVDW	2+17/rep
CMPS dss, srs	Compare string	X- - -XXXXX	CMPS dss, srs	22
		REPE	CMPS dss, srs	9+22/rep
SCAS dss	Scan string	X- - -XXXXX	SCAS dss	15
		REPNE	SCAS dss	9+15/rep
LODS srs	Load string (byte or word)	- - - - - - - - -	LODS srs	12
		REP	LODS srs	9+13/rep
STOS dss	Store (byte or word) string	- - - - - - - - -	STOS dss	11
		REP	STOS dss	9+10/rep
5. Control transfer				
CALL target	Call a procedure	- - - - - - - - -	CALL near-proc	19
			CALL far-proc	28
			CALL memptr16	21+EA
			CALL regptr16	16
			CALL memptr32	37+EA
RET const	Return from procedure	- - - - - - - - -	RET intra-seg	8
			RET intra-seg, const	12
			RET inter-seg	18
			RET inter-seg, const	17
JMP target	Jump	- - - - - - - - -	JMP slb	15
			JMP near-lbl	15
			JMP far-lbl	15
			JMP memptr16	18+EA
			JMP regptr16	11
			JMP memptr32	24+EA
JA/JNBE slb	Jump if above	CF=0, ZF=0	JA slb	16 or 4
JAE/JNB slb	Jump if above or equal	CF=0	JAE slb	16 or 4
JB/JNAE slb	Jump if below	CF=1	JB slb	16 or 4
JBE/JNA slb	Jump if below or equal	CF=1, ZF=1	JBE slb	16 or 4
JC slb	Jump if carry	CF=1	JC slb	16 or 4
JE/JZ slb	Jump if equal/zero	ZF=1	JE slb	16 or 4

JG/JNLE slb	Jump if greater	ZF=0, SF=0F	JNLE slb	16 or 4
JGE/JNL slb	Jump if greater or equal	SF=0F	JNL slb	16 or 4
JL/JNGE slb	Jump if less	SF≠0F	JL slb	16 or 4
JLE/JNG slb	Jump if less or equal	ZF=1, SF≠0	JNG slb	16 or 4
JNC slb	Jump if not carry	CF=0	JNC slb	16 or 4
JNE/JNZ slb	Jump if not equal/not zero	ZF=0	JNE slb	16 or 4
JNO slb	Jump if not overflow	OF=0	JNO slb	16 or 4
JNP/JPO slb	Jump if not parity / if odd	PF=0	JNP slb	16 or 4
JNS slb	Jump if not sign	SF=0	JNS slb	16 or 4
JO slb	Jump if overflow	OF=1	JO slb	16 or 4
JP/JPE slb	Jump if parity/even	PF=1	JP slb	16 or 4
JS slb	Jump if sign	SF=1	JS slb	16 or 4
LOOPslb	Loop	- - - - - - - - -	LOOP slb	17 or 5
LOOPE/LOOPZ slb	Loop if equal/zero		LOOPE slb	18 or 6
LOOPNE/ LOOPNZ slb	Lopp if not equal/not zero		LOOPNE slb	19 or 5
JCXZ slb	Jump if CX is zero		JCXZ slb	16 or 4
INT int-type	Interrupt	- - OO - - - - -	INT imd8(type-3)	52
			INT imd8(type#3)	51
INTO	Interrupt is overflow	- - OO - - - - -	INTO	53
IRET	Interrupt return	RRRRRRRRR	IRET	24
6. Processor control				
STC	Set carry flag	- - - - - - - - 1	STC	2
CLC	Clear carry flag	- - - - - - - - 0	CLC	2
CMC	Complement carry flag	- - - - - - - - X	CMC	2
STD	Set direction flag	- 1 - - - - - - -	STD	2
CLD	Clear direction flag	- 0 - - - - - - -	CLD	2
STI	Set interrupt flag	- - 1 - - - - - -	STI	2
CLI	Clear interrupt flag	- - 0 - - - - - -	CLI	2
ESC eoc, src	Escape	- - - - - - - - -	ESC imd, mem	8+EA
		- - - - - - - - -	ESC imd, reg	2
HLT	Halt	- - - - - - - - -	HLT	2
LOCK	Lock bus	- - - - - - - - -	LOCK	2
WAIT	Wait while TEST pin not asserted	- - - - - - - - -	WAIT	3+5n
NOP	No operation	- - - - - - - - -	NOP	3

Oznake za flag registar:	-	ne utiče na stanje flega
	0	postavlja fleg na 0
	1	postavlja fleg na 1
	X	briše ili postavlja fleg
	U	vrednost flega nije važeća
	R	obnovljena vrednost flega

Broj taktova za izračunavanje EA:	Displacement	6
	Base or Index	5
	Displacement + Base or Index	9
	Base + Index	7 or 7
	Displacement + Base + Index	11 or 12

Legenda:

acc	- akumulator	sreg	- segmentni registar	memptr16	- pointer u memoriji
mem	- memorija	src	- source	memeptr32	
mem16		dst	- destination	regptr16	- pointer u registru
mem32		srs	- source string		
imd	- neposredni operand	dss	- destination string		
imd8		slb	- short label		
reg	- registar	lbl	- label		
reg8		const	- constant		
reg16					