Appendix D: Instruction Set Reference

This section provides encodings and approximate cycle times for all instructions that you would normally execute in *real* mode on an Intel processor. Missing are the special instructions on the 80286 and later processors that manipulate page tables, segment descriptors, and other instructions that only an operating system should use. The cycle times are approximate. To determine exact execution times, you will need to run an experiment. The cycle times are given for comparison purposes only.

Key to special bits in encodings:

x:	Don't care. Can be zero or one.
s:	Sign extension bit for immediate operands. If zero, immediate operand is 16 or 32 bits depending on destination operand size. If s bit is one, then the immediate operand is eight bits and the CPU sign extends to 16 or 32 bits, as appropriate.
rrr:	Same as reg field in [mod-reg-r/m] byte.

Other Notes:

[disp]	This field can be zero, one, two, or four bytes long as required by the instruction.
[imm]	This field is one byte long if the operand is an eight bit operand or if the *s* bit in the instruction opcode is one. It is two or four bytes long if the *s* bit contains zero and the destination operand is 16 or 32 bits, respectively.
[mod-reg-r/m]:	Instructions that have a mod-reg-r/m byte may have a scaled index byte (sib) and a zero, one, two, or four byte displacement. See Appendix E for details concerning the encoding of this portion of the instruction.
reg,reg	Many instructions allow two operands using a [mod-reg-r/m] byte. A single *direction* bit in the opcode determines whether the instruction treats the *reg* operand as the destination or the mod-r/m operand as the destination (e.g., mov reg,mem vs. mov mem,reg). Such instructions also allow two register operands. It turns out there are two encodings for each such reg-reg instruction. That is, you can encode an instruction like mov ax, bx with ax encoded in the reg field and bx encoded in the mod-r/m field, or you can encode it with bx encoded in the reg field and ax encoded in the mod-r/m field. Such instructions always have an *x* bit in the opcode. If the *x* bit is zero, the destination is the register specified by the mod-r/m field. If the *x* bit is one, the destination is the register specified by the reg field. Other types of instructions support multiple encodings for similar reasons.

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
aaa	0011 0111	8	8	3	4	3	3
aad	1101 0101 0000 1010	60	60	14	19	14	10
aam	1101 0100 0000 1010	83	83	16	17	15	18
aas	0011 1111	8	8	3	4	3	3
adc reg8, reg8	0001 00x0 [11-reg-r/m]	3	‘3	2	2	1	1
adc reg16, reg16	0001 00x1 [11-reg-r/m]	3	3	2	2	1	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
adc reg32, reg32	0110 0110 0001 00x1 [11-reg-r/m]	3	3	2	2	1	1
adc reg8, mem8	0001 0010 [mod-reg-r/m]	9+EA	9+EA	7	6	2	2
adc reg16, mem16	0001 0011 [mod-reg-r/m]	13+EA	9+EA	7	6	2	2
adc reg32, mem32	0110 0110 0001 0011 [mod-reg-r/m]	-	-	-	6	2	2
adc mem8, reg8	0001 0000 [mod-reg-r/m]	16+EA	16+EA	7	7	3	3
adc mem16, reg16	0001 0001 [mod-reg-r/m]	24+EA	16+EA	7	7	3	3
adc mem32, reg32	0110 0110 0001 0001 [mod-reg-r/m]	-	-	-	7	3	3
adc reg8, imm8	1000 00x0 [11-010-r/m] [imm]	4	4	3	2	1	1
adc reg16, imm16	1000 00s0 [11-010-r/m] [imm]	4	4	3	2	1	1
adc reg32, imm32	0110 0110 1000 00s0 [11-010-r/m] [imm]	4	4	3	2	1	1
adc mem8, imm8	1000 00x0 [mod-010-r/m] [imm]	17+EA	17+EA	7	7	3	3
adc mem16, imm16	1000 00s1 [mod-010-r/m] [imm]	23+EA	17+EA	7	7	3	3
adc mem32, imm32	0110 0110 1000 00s1 [mod-010-r/m] [imm]	-	-	-	7	3	3
adc al, imm	0001 0100 [imm]	4	4	3	2	1	1
adc ax, imm	0001 0101 [imm]	4	4	3	2	1	1
adc eax, imm	0110 0110 0001 0101 [imm]	-	-	-	2	1	1
add reg8, reg8	0000 00x0 [11-reg-r/m]	3	‘3	2	2	1	1
add reg16, reg16	0000 00x1 [11-reg-r/m]	3	3	2	2	1	1
add reg32, reg32	0110 0110 0000 00x1 [11-reg-r/m]	3	3	2	2	1	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
add reg8, mem8	0000 0010 [mod-reg-r/m]	9+EA	9+EA	7	6	2	2
add reg16, mem16	0000 0011 [mod-reg-r/m]	13+EA	9+EA	7	6	2	2
add reg32, mem32	0110 0110 0000 0011 [mod-reg-r/m]	-	-	-	6	2	2
add mem8, reg8	0000 0000 [mod-reg-r/m]	16+EA	16+EA	7	7	3	3
add mem16, reg16	0000 0001 [mod-reg-r/m]	24+EA	16+EA	7	7	3	3
add mem32, reg32	0110 0110 0000 0001 [mod-reg-r/m]	-	-	-	7	3	3
add reg8, imm8	1000 00x0 [11-000-r/m] [imm]	4	4	3	2	1	1
add reg16, imm16	1000 00s0 [11-000-r/m] [imm]	4	4	3	2	1	1
add reg32, imm32	0110 0110 1000 00s0 [11-000-r/m] [imm]	4	4	3	2	1	1
add mem8, imm8	1000 00x0 [mod-000-r/m] [imm]	17+EA	17+EA	7	7	3	3
add mem16, imm16	1000 00s1 [mod-000-r/m] [imm]	23+EA	17+EA	7	7	3	3
add mem32, imm32	0110 0110 1000 00s1 [mod-000-r/m] [imm]	-	-	-	7	3	3
add al, imm	0000 0100 [imm]	4	4	3	2	1	1
add ax, imm	0000 0101 [imm]	4	4	3	2	1	1
add eax, imm	0110 0110 0000 0101 [imm]	-	-	-	2	1	1
and reg8, reg8	0010 00x0 [11-reg-r/m]	3	‘3	2	2	1	1
and reg16, reg16	0010 00x1 [11-reg-r/m]	3	3	2	2	1	1
and reg32, reg32	0110 0110 0010 00x1 [11-reg-/rm]	3	3	2	2	1	1
and reg8, mem8	0010 0010 [mod-reg-r/m]	9+EA	9+EA	7	6	2	2

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
and reg16, mem16	0010 0011 [mod-reg-r/m]	13+EA	9+EA	7	6	2	2
and reg32, mem32	0110 0110 0010 0011 [mod-reg-r/m]	-	-	-	6	2	2
and mem8, reg8	0010 0000 [mod-reg-r/m]	16+EA	16+EA	7	7	3	3
and mem16, reg16	0010 0001 [mod-reg-r/m]	24+EA	16+EA	7	7	3	3
and mem32, reg32	0110 0110 0010 0001 [mod-reg-r/m]	-	-	-	7	3	3
and reg8, imm8	1000 00x0 [11-100-r/m] [imm]	4	4	3	2	1	1
and reg16, imm16	1000 00s1 [11-100-r/m] [imm]	4	4	3	2	1	1
and reg32, imm32	0110 0110 1000 00s1 [11-100-r/m] [imm]	4	4	3	2	1	1
and mem8, imm8	1000 00x0 [mod-100-r/m] [imm]	17+EA	17+EA	7	7	3	3
and mem16, imm16	1000 00s1 [mod-100-r/m] [imm]	23+EA	17+EA	7	7	3	3
and mem32, imm32	0110 0110 1000 00s1 [mod-100-r/m] [imm]	-	-	-	7	3	3
and al, imm	0010 0100 [imm]	4	4	3	2	1	1
and ax, imm	0010 0101 [imm]	4	4	3	2	1	1
and eax, imm	0110 0110 0010 0101 [imm]	-	-	-	2	1	1
bound reg16, mem32	0110 0010 [mod-reg-r/m]			13 (values within range)	10	7	8
bound reg32, mem64	0110 0110 0110 0010 [mod-reg-r/m]				10 (values within range)	7	8
bsf reg16, reg16	0000 1111 1011 1100 [11-reg-r/m]				10+3*n n= first set bit.	6-42	6-34

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
bsf reg32, reg32	0110 0110 0000 1111 1011 1100 [11-reg-r/m]				10+3*n n= first set bit.	6-42	6-42
bsf reg16, mem16	0000 1111 1011 1100 [mod-reg-r/m]				10+3*n n= first set bit.	7-43	6-35
bsf reg32, mem32	0110 0110 0000 1111 1011 1100 [mod-reg-r/m]				10+3*n n= first set bit.	7-43	6-43
bsr reg16, reg16	0000 1111 1011 1101 [11-reg-r/m]				10+3*n n= first set bit.	7-100	7-39
bsr reg32, reg32	0110 0110 0000 1111 1011 1101 [11-reg-r/m]				10+3*n n= first set bit.	8-100	7-71
bsr reg16, mem16	0000 1111 1011 1101 [mod-reg-r/m]				10+3*n n= first set bit.	7-101	7-40
bsr reg32, mem32	0110 0110 0000 1111 1011 1101 [mod-reg-r/m]				10+3*n n= first set bit.	8-101	7-72
bswap reg32	0000 1111 11001rrr					1	1
bt reg16, reg16	0000 1111 1010 0011 [11-reg-r/m]				3	3	4
bt reg32, reg32	0110 0110 0000 1111 1010 0011 [11-reg-r/m]				3	3	4
bt mem16, reg16	0000 1111 1010 0011 [mod-reg-r/m]				12	8	9
bt mem32, reg32	0110 0110 0000 1111 1010 0011 [mod-reg-r/m]				12	8	9
bt reg16, imm	0000 1111 1011 1010 [11-100-r/m] [imm8]				3	3	4
bt reg32, imm	0110 0110 0000 1111 1011 1010 [11-100-r/m] [imm8]				3	3	4
bt mem16, imm	0000 1111 1011 1010 [mod-100-r/m]				6	3	4

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
bt mem32, imm	0110 0110 0000 1111 1011 1010 [mod-100-r/m]				6	3	4
btc reg16, reg16	0000 1111 1011 1011 [11-reg-r/m]				6	6	7
btc reg32, reg32	0110 0110 0000 1111 1011 1011 [11-reg-r/m]				6	6	7
btc mem16, reg16	0000 1111 1011 1011 [mod-reg-r/m]				13	13	13
btc mem32, reg32	0110 0110 0000 1111 1011 1011 [mod-reg-r/m]				13	13	13
btc reg16, imm	0000 1111 1011 1010 [11-111-r/m] [imm8]				6	6	7
btc reg32, imm	0110 0110 0000 1111 1011 1010 [11-111-r/m] [imm8]				6	6	7
btc mem16, imm	0000 1111 1011 1010 [mod-111-r/m] [imm8]				8	8	8
btc mem32, imm	0110 0110 0000 1111 1011 1010 [mod-111-r/m] [imm8]				8	8	8
btr reg16, reg16	0000 1111 1011 0011 [11-reg-r/m]				6	6	7
btr reg32, reg32	0110 0110 0000 1111 1011 0011 [11-reg-r/m]				6	6	7
btr mem16, reg16	0000 1111 1011 0011 [mod-reg-r/m]				13	13	13
btr mem32, reg32	0110 0110 0000 1111 1011 0011 [mod-reg-r/m]				13	13	13

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
btr reg16, imm	0000 1111 1011 1010 [11-110-r/m] [imm8]				6	6	7
btr reg32, imm	0110 0110 0000 1111 1011 1010 [11-110-r/m] [imm8]				6	6	7
btr mem16, imm	0000 1111 1011 1010 [mod-110-r/m] [imm8]				8	8	8
btr mem32, imm	0110 0110 0000 1111 1011 1010 [mod-110-r/m] [imm8]				8	8	8
bts reg16, reg16	0000 1111 1010 1011 [11-reg-r/m]				6	6	7
bts reg32, reg32	0110 0110 0000 1111 1010 1011 [11-reg-r/m]				6	6	7
bts mem16, reg16	0000 1111 1010 1011 [mod-reg-r/m]				13	13	13
bts mem32, reg32	0110 0110 0000 1111 1010 1011 [mod-reg-r/m]				13	13	13
bts reg16, imm	0000 1111 1011 1010 [11-101-r/m] [imm8]				6	6	7
bts reg32, imm	0110 0110 0000 1111 1011 1010 [11-101-r/m] [imm8]				6	6	7
bts mem16, imm	0000 1111 1011 1010 [mod-101-r/m] [imm8]				8	8	8
bts mem32, imm	0110 0110 0000 1111 1011 1010 [mod-101-r/m] [imm8]				8	8	8
call near	1110 1000 [disp16]	23	19	7-10	7-10	3	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
call far	1001 1010 [offset] [segment]	36	28	13-16	17-20	18	4
call reg16	1111 1111 [11-010-r/m]	20	16	7-10	7-10	5	2
call mem16	1111 1111 [mod-010-r/m]	29+EA	21+EA	11-14	10-13	5	2
call mem32	1111 1111 [mod-011-r/m]	53+EA	37+EA	16-19	22-25	17	5
cbw	1001 1000	2	2	2	3	3	3
cdq	0110 0110 1001 1001				2	2	2
clc	1111 1000	2	2	2	2	2	2
cld	1111 1100	2	2	2	2	2	2
cli	1111 1010	2	2	3		5	7
cmc	1111 0101	2	2	2	2	2	2
cmp reg8, reg8	0011 10x0 [11-reg-r/m]	3	‘3	2	2	1	1
cmp reg16, reg16	0011 10x1 [11-reg-r/m]	3	3	2	2	1	1
cmp reg32, reg32	0110 0110 0011 10x1 [11-reg-/rm]	3	3	2	2	1	1
cmp reg8, mem8	0011 1010 [mod-reg-r/m]	9+EA	9+EA	7	6	2	2
cmp reg16, mem16	0011 1011 [mod-reg-r/m]	13+EA	9+EA	7	6	2	2
cmp reg32, mem32	0110 0110 0011 1011 [mod-reg-r/m]	-	-	-	6	2	2
cmp mem8, reg8	0011 1000 [mod-reg-r/m]	9+EA	9+EA	7	6	2	2
cmp mem16, reg16	0011 1001 [mod-reg-r/m]	13+EA	9+EA	7	6	2	2
cmp mem32, reg32	0110 0110 0011 1001 [mod-reg-r/m]	-	-	-	6	2	2
cmp reg8, imm8	1000 00x0 [11-111-r/m] [imm]	4	4	3	2	1	1
cmp reg16, imm16	1000 00s0 [11-111-r/m] [imm]	4	4	3	2	1	1
cmp reg32, imm32	0110 0110 1000 00s0 [11-111-r/m] [imm]	4	4	3	2	1	1
cmp mem8, imm8	1000 00x0 [mod-111-r/m] [imm]	10+EA	10+EA	6	5	2	2

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
cmp mem16, imm16	1000 00s1 [mod-111-r/m] [imm]	14+EA	10+EA	6	5	2	2
cmp mem32, imm32	0110 0110 1000 00s1 [mod-111-r/m] [imm]	-	-	-	5	2	2
cmp al, imm	0011 1100 [imm]	4	4	3	2	1	1
cmp ax, imm	0011 1101 [imm]	4	4	3	2	1	1
cmp eax, imm	0110 0110 0011 1101 [imm]	-	-	-	2	1	1
cmpsb	1010 0110	30	22	8	10	8	5
cmpsw	1010 0111	30	22	8	10	8	5
cmpsd	0110 0110 1010 0111	-	-	-	10	8	5
repe cmpsb	1111 0011 1010 0110	9+17*cx cx = # of repetitions	9+17*cx	5+9*cx	5+9*cx	7+7*cx 5 if cx=0	9+4*cx 7 if cx=0
repne cmpsb	1111 0010 1010 0110	9+17*cx	9+17*cx	5+9*cx	5+9*cx	7+7*cx 5 if cx=0	9+4*cx 7 if cx=0
repe cmpsw	1111 0011 1010 0111	9+25*cx	9+17*cx	5+9*cx	5+9*cx	7+7*cx 5 if cx=0	9+4*cx 7 if cx=0
repne cmpsw	1111 0010 1010 0111	9+25*cx	9+17*cx	5+9*cx	5+9*cx	7+7*cx 5 if cx=0	9+4*cx 7 if cx=0
repe cmpsd	0110 0110 1111 0011 1010 0111	-	-	-	5+9*cx	7+7*cx 5 if cx=0	9+4*cx 7 if cx=0
repne cmpsd	0110 0110 1111 0010 1010 0111	-	-	-	5+9*cx	7+7*cx 5 if cx=0	9+4*cx 7 if cx=0
cmpxchg reg8, reg8	0000 1111 1011 0000 [11-reg-r/m] Note: r/m is first register operand.	-	-	-	-	6	6
cmpxchg reg16, reg16	0000 1111 1011 0001 [11-reg-r/m]	-	-	-	-	6	6
cmpxchg reg32, reg32	0110 0110 0000 1111 1011 0001 [11-reg-r/m]	-	-	-	-	6	6
cmpxchg mem8, reg8	0000 1111 1011 0000 [mod-reg-r/m]	-	-	-	-	7 if equal, 10 if not equal	6

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
cmpxchg mem16, reg16	0000 1111 1011 0001 [mod-reg-r/m]	-	-	-	-	7 if equal, 10 if not equal	6
cmpxchg mem32, reg32	0110 0110 0000 1111 1011 0001 [mod-reg-r/m]	-	-	-	-	7 if equal, 10 if not equal	6
cmpxchg8b mem64	0000 1111 1100 0111 [mod-001-r/m]	-	-	-	-	-	10
cpuid	0000 1111 1010 0010	-	-	-	-	-	14
cwd	1001 1001	5	5	2	2	3	2
cwde	0110 0110 1001 1000				3	3	3
daa	0010 0111	4	4	3	4	2	3
das	0010 1111	4	4	3	4	2	3
dec reg8	1111 1110 [11-001-r/m]	3	3	2	2	1	1
dec reg16	0100 1rrr	3	3	2	2	1	1
dec reg16 (alternate encoding)	1111 1111 [11-001-r/m]	3	3	2	2	1	1
dec reg32	0110 0110 0100 1rrr	3	3	2	2	1	1
dec reg32 (alternate encoding)	0110 0110 1111 1111 [11-001-r/m]	3	3	2	2	1	1
dec mem8	1111 1110 [mod-001-r/m]	15+EA	15+EA	7	6	3	3
dec mem16	1111 1111 [mod-001-r/m]	23+EA	15+EA	7	6	3	3
dec mem32	0110 0110 1111 1111 [mod-001-r/m]	-	-	-	6	3	3
div reg8	1111 0110 [11-110-r/m]	80-90	80-90	14	14	16	17
div reg16	1111 0111 [11-110-r/m]	144-162	144-162	22	22	24	25
div reg32	0110 0110 1111 0111 [11-110-r/m]	-	-	-	38	40	41
div mem8	1111 0110 [mod-110-r/m]	(86-96) + EA	(86-96) + EA	17	17	16	17
div mem16	1111 0111 [mod-110-r/m]	(158-176) + EA	(150-168) + EA	25	25	24	25
div mem32	0110 0110 1111 0111 [mod-110-r/m]	-	-	-	41	40	41

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
enter local, 0	1100 1000 [locals-imm16] 0000 0000			11	10	14	11
enter local, 1	1100 1000 [locals-imm16] 0000 0001			15	12	17	15
enter local, lex	1100 1000 [locals:imm16] [lex:imm8]			12 + 4 * (lex-1)	15 + 4 * (lex-1)	17 + 3*lex	15 + 2*lex
hlt	1111 0100	2+[d]	2+	2+	5+	4+	12+
idiv reg8	1111 0110 [11-111-r/m]	101-112	101-112	17	19	19	22
idiv reg16	1111 0111 [11-111-r/m]	165-184	165-184	25	27	27	30
idiv reg32	0110 0110 1111 0111 [11-111-r/m]	-	-	-	43	43	46
idiv mem8	1111 0110 [mod-111-r/m]	(107-118) + EA	(107-118) + EA	20	22	20	30
idiv mem16	1111 0111 [mod-111-r/m] [disp]	(175-194) + EA	(171-190) + EA	28	30	28	30
idiv mem32	0110 0110 1111 0111 [mod-111-r/m]	-	-	-	46	44	46
imul reg8	1111 0110 [11-101-r/m]	80-98	80-98	13	9-14	13-18	11
imul reg16	1111 0111 [11-101-r/m]	128-154	128-154	21	9-22	13-26	11
imul reg32	0110 0110 1111 0111 [11-101-r/m]	-	-	-	9-38	13-42	11
imul mem8	1111 0110 [mod-101-r/m]	(86-104) + EA	(107-118) + EA	16	12-17	13-18	11
imul mem16	1111 0111 [mod-101-r/m]	(134-164) + EA	(134-160) + EA	24	15-25	13-26	11
imul mem32	0110 0110 1111 0111 [mod-101-r/m]	-	-	-	12-41	13-42	11
imul reg16, reg16, imm8 imul reg16, imm8 (Second form assumes reg and r/m are the same, instruction sign extends eight bit immediate operand to 16 bits)	0110 1011 [11-reg-r/m] [imm8] (1st reg operand is specified by reg field, 2nd reg operand is specified by r/m field)	-	-	21	13-26	13-26	10
imul reg16, reg16, imm imul reg16, imm	0110 1001 [11-reg-r/m] [imm16]	-	-	21	9-22	13-26	10

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
imul reg32, reg32, imm8 imul reg32, imm8	0110 0110 0110 1011 [11-reg-r/m] [imm8]	-	-		13-42	13-42	10
imul reg32, reg32, imm imul reg32, imm	0110 0110 0110 1001 [11-reg-r/m] [imm32]	-	-	-	9-38	13-42	10
imul reg16,mem16,imm8	0110 1011 [11-reg-r/m] [imm8]	-	-	24	14-27	13-26	10
imul reg16,mem16,imm	0110 1001 [11-reg-r/m] [imm16]	-	-	24	12-25	13-26	10
imul reg32, mem32, imm8	0110 0110 0110 1011 [11-reg-r/m] [imm8]	-	-	-	14-43	13-42	10
imul reg32, mem32, imm	0110 0110 0110 1001 [11-reg-r/m] [imm32]	-	-	-	12-41	13-42	10
imul reg16, reg16	0000 1111 1010 1111 [11-reg-r/m] (reg is dest operand)	-	-	-	12-25	13-26	10
imul reg32, reg32	0110 0110 0000 1111 1010 1111 [11-reg-r/m] (reg is dest operand)	-	-	-	12-41	12-42	10
imul reg16, mem16	0000 1111 1010 1111 [mod-reg-r/m]	-	-	-	15-28	13-26	10
imul reg32, mem32	0110 0110 0000 1111 1010 1111 [mod-reg-r/m]	-	-	-	14-44	13-42	10
in al, port	1110 0100 [port8]	10	10	5	12	14	7
in ax, port	1110 0101 [port8]	14	10	5	12	14	7
in eax, port	0110 0110 1110 0101 [port8]	-	-	-	12	14	7
in al, dx	1110 1100	8	8	5	13	14	7
in ax, dx	1110 1101	12	8	5	13	14	7
in eax, dx	0110 0110 1110 1101	12	8	5	13	14	7

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
inc reg8	1111 1110 [11-000-r/m]	3	2	2	2	1	1
inc reg16	0100 0rrr	3	3	2	2	1	1
inc reg16 (alternate encoding)	1111 1111 [11-000-r/m]	3	3	2	2	1	1
inc reg32	0110 0110 0100 0rrr	-	-	-	2	1	1
inc reg32 (alternate encoding)	0110 0110 1111 1111 [11-000-r/m]	-	-	-	2	1	1
inc mem8	1111 1110 [mod-000-r/m]	15+EA	15+EA	7	6	3	3
inc mem16	1111 1110 [mod-000-r/m] [disp]	23+EA	15+EA	7	6	3	3
inc mem32	0110 0110 1111 1110 [mod-000-r/m]	-	-	-	6	3	3
insb	1010 1010	-	-	5	15	17	9
insw	1010 1011	-	-	5	15	17	9
insd	0110 0110 1010 1011	-	-	-	15	17	9
rep insb	1111 0010 1010 1010	-	-	5 + 4*cx	14 + 6*cx	16+8*cx	11 + 3*cx
rep insw	1111 0010 1010 1011	-	-	5 + 4*cx	14 + 6*cx	16+8*cx	11 + 3*cx
rep insd	0110 0110 1111 0010 1010 1011	-	-	-	14 + 6*cx	16+8*cx	11 + 3*cx
int nn	1100 1101 [imm8]	71	51	23-26	37	30	16
int 03	1100 1100	72	52	23-26	33	26	13
into	1100 1110	73 (if ovr) 4 (no ovr)	53 4	24-27 3	35 3	28 3	13 3
iret	1100 1111	44	32	17-20	22	15	8
iretd	0110 0110 1100 1111				22	15	10
ja short	0111 0111 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
ja near	0000 1111 1000 0111 [disp16]	-	-	-	7-10 3	3 1	1
jae short	0111 0011 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jae near	0000 1111 1000 0011 [disp16]	-	-	-	7-10 3	3 1	1
jb short	0111 0010 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
jb near	0000 1111 1000 0010 [disp16]	- 	- 	- 	7-10 3	3 1	1
jbe short	0111 0110 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jbe near	0000 1111 1000 0110 [disp16]	- 	- 	- 	7-10 3	3 1	1
jc short	0111 0010 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jc near	0000 1111 1000 0010 [disp16]	- 	- 	- 	7-10 3	3 1	1
je short	0111 0100 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
je near	0000 1111 1000 0100 [disp16]	- 	- 	- 	7-10 3	3 1	1
jg short	0111 1111 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jg near	0000 1111 1000 1111 [disp16]	- 	- 	- 	7-10 3	3 1	1
jge short	0111 1101 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jge near	0000 1111 1000 1101 [disp16]	- 	- 	- 	7-10 3	3 1	1
jl short	0111 1100 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jl near	0000 1111 1000 1100 [disp16]	- 	- 	- 	7-10 3	3 1	1
jle short	0111 1110 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jle near	0000 1111 1000 1110 [disp16]	- 	- 	- 	7-10 3	3 1	1
jna short	0111 0110 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jna near	0000 1111 1000 0110 [disp16]	- 	- 	- 	7-10 3	3 1	1
jnae short	0111 0010 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jnae near	0000 1111 1000 0010 [disp16]	- 	- 	- 	7-10 3	3 1	1
jnb short	0111 0011 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
jnb near	0000 1111 1000 0011 [disp16]	- 	- 	- 	7-10 3	3 1	1
jnbe short	0111 0111 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jnbe near	0000 1111 1000 0111 [disp16]	- 	- 	- 	7-10 3	3 1	1
jnc short	0111 0011 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jnc near	0000 1111 1000 0011 [disp16]	- 	- 	- 	7-10 3	3 1	1
jne short	0111 0101 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jne near	0000 1111 1000 0101 [disp16]	- 	- 	- 	7-10 3	3 1	1
jng short	0111 1110 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jng near	0000 1111 1000 1110 [disp16]	- 	- 	- 	7-10 3	3 1	1
jnge short	0111 1100 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jnge near	0000 1111 1000 1100 [disp16]	- 	- 	- 	7-10 3	3 1	1
jnl short	0111 1101 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jnl near	0000 1111 1000 1101 [disp16]	- 	- 	- 	7-10 3	3 1	1
jnle short	0111 1111 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jnle near	0000 1111 1000 1111 [disp16]	- 	- 	- 	7-10 3	3 1	1
jno short	0111 0001 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jno near	0000 1111 1000 0001 [disp16]	- 	- 	- 	7-10 3	3 1	1
jnp short	0111 1011 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1
jnp near	0000 1111 1000 1011 [disp16]	- 	- 	- 	7-10 3	3 1	1
jns short	0111 1001 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
jns near	0000 1111 1000 1001 [disp16]	- - -	- - -	- - -	7-10 3 7-10	3 1 3	1 1 1
jnz short	0111 0101 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1 1
jnz near	0000 1111 1000 0101 [disp16]	- - -	- - -	- - -	7-10 3 7-10	3 1 3	1 1 1
jo short	0111 0000 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1 1
jo near	0000 1111 1000 0000 [disp16]	- - -	- - -	- - -	7-10 3 7-10	3 1 3	1 1 1
jp short	0111 1010 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1 1
jp near	0000 1111 1000 1010 [disp16]	- - -	- - -	- - -	7-10 3 7-10	3 1 3	1 1 1
jpe short	0111 1010 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1 1
jpe near	0000 1111 1000 1010 [disp16]	- - -	- - -	- - -	7-10 3 7-10	3 1 3	1 1 1
jpo short	0111 1011 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1 1
jpo near	0000 1111 1000 1011 [disp16]	- - -	- - -	- - -	7-10 3 7-10	3 1 3	1 1 1
js short	0111 1000 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1 1
js near	0000 1111 1000 1000 [disp16]	- - -	- - -	- - -	7-10 3 7-10	3 1 3	1 1 1
jz short	0111 0100 [disp8]	16 4 (not taken)	16 4	7-10 3	7-10 3	3 1	1 1
jz near	0000 1111 1000 0100 [disp16]	- - -	- - -	- - -	7-10 3 7-10	3 1 3	1 1 1
jcxz short	1110 0011 [disp8]	18 6 (not taken)	18 6	8-11 4	9-12 5	8 5	6 5
jecxz short	0110 0110 1110 0011 [disp8]				9-12 5	8 5	6 5
jmp short	1110 1011 [disp8]	15	15	7-10	7-10	3	1
jmp near	1110 1001 [disp16]	15	15	7-10	7-10	3	1
jmp reg16	1111 1111 [11-100-r/m]	11	11	7-10	7-10	5	2

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
jmp mem16	1111 1111 [mod-100-r/m]	18+EA	18+EA	11-14	10-13	5	2
jmp far	1110 1010 [offset16] [segment16]	15	15	11-14	12-15	17	3
jmp mem32	1111 1111 [mod-101-r/m]	24+EA	24+EA	15-18	43-46	13	2
lahf	1001 1111	4	4	2	2	3	2
lds reg, mem32	1100 0101 [mod-reg-r/m]	24+EA	16+EA	7	7	6	4
lea reg, mem	1000 1101 [mod-101-r/m]	2+EA	2+EA	3	2	1	1
leave	1100 1001	-	-	5	4	5	3
les reg, mem32	1100 0100 [mod-reg-r/m]	24+EA	16+EA	7	7	6	4
lfs reg, mem32	0000 1111 1011 0100 [mod-reg-r/m]	-	-	-	7	6	4
lgs reg, mem32	0000 1111 1011 0101 [mod-reg-r/m]	-	-	-	7	6	4
lodsb	1010 1100	12	12	5	5	5	2
lodsw	1010 1101	16	12	5	5	5	2
loadsd	0110 0110 1010 1101	-	-	-	5	5	2
loop short	1110 0010 [disp8]	17 5 (not taken)	17 5	8-11 4	11-14	7 6	5
loope short loopz short	1110 0001 [disp8]	18 6 (not taken)	18 6	8-11 4	11-14	9 6	7
loopne short loopnz short	1110 0000 [disp8]	19 5(not taken)	19 5	8-11 4	11-14	9 6	7
lss reg, mem32	0000 1111 1011 0010 [mod-reg-r/m]	-	-	-	7	6	4
mov reg8, reg8	1000 1000 [11-reg-r/m] (r/m specifies destination reg)	2	2	2	2	1	1
mov reg8, reg8 (alternate encoding)	1000 1010 [11-reg-r/m] (reg specifies destination reg)	2	2	2	2	1	1
mov reg16, reg16	1000 1001 [11-reg-r/m] (r/m specifies destination reg)	2	2	2	2	1	1
mov reg16, reg16 (alternate encoding)	1000 1011 [11-reg-r/m] (reg specifies destination reg)	2	2	2	2	1	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
mov reg32, reg32	0110 0110 1000 1001 [11-reg-r/m] (r/m specifies destination reg)	-	-	-	2	1	1
mov reg32, reg32 (alternate encoding)	0110 0110 1000 1011 [11-reg-r/m] (reg specifies destination reg)	-	-	-	2	1	1
mov mem, reg8	1000 1000 [mod-reg-r/m]	9+EA	9+EA	3	2	1	1
mov reg8, mem	1000 1010 [mod-reg-r/m]	8+EA	8+EA	5	4	1	1
mov mem, reg16	1000 1001 [mod-reg-r/m]	13+EA	9+EA	3	2	1	1
mov reg16, mem	1000 1011 [mod-reg-r/m]	12+EA	8+EA	5	4	1	1
mov mem, reg32	0110 0110 1000 1001 [mod-reg-r/m]	-	-	-	2	1	1
mov reg16, mem	0110 0110 1000 1011 [mod-reg-r/m]	-	-	-	4	1	1
mov reg8, imm	1011 0rrr [imm8]	4	4	2	2	1	1
mov reg8, imm (alternate encoding)	1100 0110 [11-000-r/m] [imm8]	10	10	2	2	1	1
mov reg16, imm	1011 1rrr [imm16]	4	4	2	2	1	1
mov reg16, imm (alternate encoding)	1100 0111 [11-000-r/m] [imm16]	10	10	2	2	1	1
mov reg32, imm	0110 0110 1011 1rrr [imm32]	-	-	-	2	1	1
mov reg32, imm (alternate encoding)	0110 0110 1100 0111 [11-000-r/m] [imm32]	-	-	-	2	1	1
mov mem8, imm	1100 0110 [mod-000-r/m] [imm8]	10+EA	10+EA	3	2	1	1
mov mem16, imm	1100 0111 [mod-000-r/m] [imm16]	14+EA	10+EA	3	2	1	1
mov mem32, imm	1100 0111 [mod-000-r/m] [imm32]	-	-	-	2	1	1
mov al, disp	1010 0000 [disp]	10	10	5	4	1	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
mov ax, disp	1010 0001 [disp]	14	10	5	4	1	1
mov eax, disp	0110 0110 1010 0001 [disp]	-	-	-	4	1	1
mov disp, al	1010 0010 [disp]	10	10	3	2	1	1
mov disp, ax	1010 0011 [disp]	14	10	3	2	1	1
mov disp, eax	0110 0110 1010 0011 [disp]	-	-	-	2	1	1
mov segreg, reg16	1000 1110 [11-sreg-r/m]	2	2	2	2	3	2-3
mov segreg, mem	1000 1110 [mod-reg-r/m]	12+EA	8+EA	5	5	3	2-3
mov reg16, segreg	1000 1100 [11-sreg-r/m]	2	2	2	2	3	1
mov mem, segreg	1000 1100 [mod-reg-r/m]	13+EA	9+EA	3	2	3	1
movsb	1010 0100	18	18	5	8	7	4
movsw	1010 0101	26	18	5	8	7	4
movsd	0110 0110 1010 0101	-	-	-	8	7	4
rep movsb	1111 0010 1010 0100	9 + 17 * cx	9 + 17*cx	5 + 4*cx	8 + 4*cx	12 + 3*cx 5 if cx=0 13 if cx=1	4 + 3*cx
rep movsw	1111 0010 1010 0101	9 + 25 * cx	9 + 17*cx	5 + 4*cx	8 + 4*cx	12 + 3*cx 5 if cx=0 13 if cx=1	4 + 3*cx
rep movsd	0110 0110 1111 0010 1010 0101	-	-	-	8 + 4*cx	12 + 3*cx 5 if cx=0 13 if cx=1	4 + 3*cx
movsx reg16, reg8	0000 1111 1011 1110 [11-reg-r/m] (dest is reg operand)				3	3	3
movsx reg32, reg8	0110 0110 0000 1111 1011 1110 [11-reg-r/m]				3	3	3
movsx reg32, reg16	0110 0110 0000 1111 1011 1111 [11-reg-r/m]				3	3	3
movsx reg16, mem8	0000 1111 1011 1110 [mod-reg-r/m]				6	3	3

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
movsx reg32, mem8	0110 0110 0000 1111 1011 1110 [mod-reg-r/m]				6	3	3
movsx reg32, mem16	0110 0110 0000 1111 1011 1111 [mod-reg-r/m]				6	3	3
movzx reg16, reg8	0000 1111 1011 0110 [11-reg-r/m] (dest is reg operand)				3	3	3
movzx reg32, reg8	0110 0110 0000 1111 1011 0110 [11-reg-r/m]				3	3	3
movzx reg32, reg16	0110 0110 0000 1111 1011 0111 [11-reg-r/m]				3	3	3
movzx reg16, mem8	0000 1111 1011 0110 [mod-reg-r/m]				6	3	3
movzx reg32, mem8	0110 0110 0000 1111 1011 0110 [mod-reg-r/m]				6	3	3
movzx reg32, mem16	0110 0110 0000 1111 1011 0111 [mod-reg-r/m]				6	3	3
mul reg8	1111 0110 [11-100-r/m]	70-77	70-77	13	9-14	13-18	11
mul reg16	1111 0111 [11-100-r/m]	118-133	118-133	21	9-22	13-26	11
mul reg32	0110 0110 1111 0111 [11-100-r/m]	-	-	-	9-38	13-42	10
mul mem8	1111 0110 [mod-100-r/m]	(76-83) + EA	(76-83) + EA	16	12-17	13-18	11
mul mem16	1111 0111 [mod-100-r/m]	(124-139) + EA	(124-139) + EA	24	12-25	13-26	11
mul mem32	0110 0110 1111 0111 [mod-100-r/m]	-	-	-	12-41	13-42	10
neg reg8	1111 0110 [11-011-r/m]	3	3	2	2	1	1
neg reg16	1111 0111 [11-011-r/m]	3	3	2	2	1	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
neg reg32	0110 0110 1111 0111 [11-011-r/m]	3	3	2	2	1	1
neg mem8	1111 0110 [mod-011-r/m]	16+EA	16+EA	7	6	3	3
neg mem16	1111 0111 [mod-011-r/m]	24+EA	16+EA	7	6	3	3
neg mem32	0110 0110 1111 0111 [mod-011-r/m]	-	-	-	6	3	3
nop (same as xchg ax, ax)	1001 0000	3	3	3	3	1	1
not reg8	1111 0110 [11-010-r/m]	3	3	2	2	1	1
not reg16	1111 0111 [11-010-r/m]	3	3	2	2	1	1
not reg32	0110 0110 1111 0111 [11-010-r/m]	3	3	2	2	1	1
not mem8	1111 0110 [mod-010-r/m]	16+EA	16+EA	7	6	3	3
not mem16	1111 0111 [mod-010-r/m]	24+EA	16+EA	7	6	3	3
not mem32	0110 0110 1111 0111 [mod-010-r/m]	-	-	-	6	3	3
or reg8, reg8	0000 10x0 [11-reg-r/m]	3	‘3	2	2	1	1
or reg16, reg16	0000 10x1 [11-reg-r/m]	3	3	2	2	1	1
or reg32, reg32	0110 0110 0000 10x1 [11-reg-r/m]	3	3	2	2	1	1
or reg8, mem8	0000 1010 [mod-reg-r/m]	9+EA	9+EA	7	6	2	2
or reg16, mem16	0000 1011 [mod-reg-r/m]	13+EA	9+EA	7	6	2	2
or reg32, mem32	0110 0110 0000 1011 [mod-reg-r/m]	-	-	-	6	2	2
or mem8, reg8	0000 1000 [mod-reg-r/m]	16+EA	16+EA	7	7	3	3
or mem16, reg16	0000 1001 [mod-reg-r/m]	24+EA	16+EA	7	7	3	3
or mem32, reg32	0110 0110 0000 1001 [mod-reg-r/m]	-	-	-	7	3	3
or reg8, imm8	1000 00x0 [11-001-r/m] [imm]	4	4	3	2	1	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
or reg16, imm16	1000 00s0 [11-001-r/m] [imm]	4	4	3	2	1	1
or reg32, imm32	0110 0110 1000 00s0 [11-001-r/m] [imm]	4	4	3	2	1	1
or mem8, imm8	1000 00x0 [mod-001-r/m] [imm]	17+EA	17+EA	7	7	3	3
or mem16, imm16	1000 00s1 [mod-001-r/m] [imm]	25+EA	17+EA	7	7	3	3
or mem32, imm32	0110 0110 1000 00s1 [mod-001-r/m] [imm]	-	-	-	7	3	3
or al, imm	0000 1100 [imm]	4	4	3	2	1	1
or ax, imm	0000 10101 [imm]	4	4	3	2	1	1
or eax, imm	0110 0110 0000 1101 [imm]	-	-	-	2	1	1
out port, al	1110 0110 [port8]	14	10	3	10	16	12
out port, ax	1110 0111 [port8]	14	10	3	10	16	12
out port, eax	0110 0110 1110 0111 [port8]	-	-	-	10	16	12
out dx, al	1110 1110	8	8	3	11	16	12
out dx, ax	1110 1111	12	8	3	11	16	12
out dx, eax	0110 0110 1110 1111	-	-	-	11	16	12
outsb	1010 1010	-	-	5	14	17	13
outsw	1010 1011	-	-	5	14	17	13
outsd	0110 0110 1010 1011	-	-	-	14	17	13
rep outsb	1111 0010 1010 1010	-	-	5 + 4*cx	12 + 5*cx	17+5*cx	13 + 4*cx
rep outsw	1111 0010 1010 1011	-	-	5 + 4*cx	12 + 5*cx	17+5*cx	13 + 4*cx
rep outsd	0110 0110 1111 0010 1010 1011	-	-	-	12 + 5*cx	17+5*cx	13 + 4*cx
pop reg16	0101 1rrr	12	8	5	4	1	1
pop reg16 (alternate encoding)	1000 1111 [11-000-r/m]	12	8	5	4	1	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
pop reg32	0110 0110 0101 1rrr	-	-	-	4	1	1
pop reg32 (alternate encoding)	0110 0110 1000 1111 [11-000-r/m]	-	-	-	5	4	3
pop mem16	1000 1111 [mod-000-r/m]	25+EA	17+EA	5	5	6	3
pop mem32	1000 1111 [mod-000-r/m]	-	-	-	5	6	3
pop es	0000 0111	12	8	5	7	3	3
pop ss	0001 0111	12	8	5	7	3	3
pop ds	0001 1111	12	8	5	7	3	3
pop fs	0000 1111 1010 0001	-	-	-	7	3	3
pop gs	0000 1111 1010 1001	-	-	-	7	3	3
popa	0110 0001	-	-	19	24	9	5
popad	0110 0110 0110 0001	-	-	-	24	9	5
popf	1001 1101	12	8	5	5	9	6
popfd	0110 0110 1001 1101	-	-	-	5	9	6
push reg16	0101 0rrr	15	11	3	2	1	1
push reg16 (alternate encoding)	1111 1111 [11-110-r/m]	15	11	3	2	1	1
push reg32	0110 0110 0101 0rrr	-	-	-	2	1	1
push reg32 (alternate encoding)	0110 0110 1111 1111 [11-110-r/m]	-	-	-	2	1	1
push mem16	1111 1111 [mod-110-r/m]	24+EA	16+EA	5	5	4	2
push mem32	1111 1111 [mod-110-r/m]	-	-	-	5	4	2
push cs	0000 1110	14	10	3	2	3	1
push ds	0001 1110	14	10	3	2	3	1
push es	0000 0110	14	10	3	2	3	1
push ss	0001 0110	14	10	3	2	3	1
push fs	0000 1111 1010 0000	-	-	-	2	3	1
push gs	0000 1111 1010 1000	-	-	-	2	3	1
push imm8->16	0110 1000 [imm8] (sign extends value to 16 bits)	-	-	3	2	1	1
push imm16	0110 1010 [imm16]	-	-	3	2	1	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
push imm32	0110 0110 0110 1010 [imm32]	-	-	-	2	1	1
pusha	0110 0000	-	-	17	18	11	5
pushad	0110 0110 0110 0000	-	-	-	18	11	5
pushf	1001 1100	14	10	3	4	4	4
pushfd	0110 0110 1001 1100	-	-	-	4	4	4
rcl reg8, 1	1101 0000 [11-010-r/m]	2	2	2	9	3	1
rcl reg16, 1	1101 0001 [11-010-r/m]	2	2	2	9	3	1
rcl reg32, 1	0110 0110 1101 0001 [11-010-r/m]	-	-	-	9	3	1
rcl mem8, 1	1101 0000 [mod-010-r/m]	15+EA	15+EA	7	10	4	3
rcl mem16, 1	1101 0001 [mod-010-r/m]	23+EA	15+EA	7	10	4	3
rcl mem32, 1	0110 0110 1101 0001 [mod-010-r/m]	-	-	-	10	4	3
rcl reg8, cl	1101 0010 [11-010-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	9	8-30	7-24
rcl reg16, cl	1101 0011 [11-010-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	9	8-30	7-24
rcl reg32, cl	0110 0110 1101 0011 [11-010-r/m]	-	-	-	9	8-30	7-24
rcl mem8, cl	1101 0010 [mod-010-r/m]	20+EA+4*cl	20+EA+4*cl	8 + cl	10	9-31	9-26
rcl mem16, cl	1101 0011 [mod-010-r/m]	28+EA+4*cl	20+EA+4*cl	8 + cl	10	9-31	9-26
rcl mem32, cl	0110 0110 1101 0011 [mod-010-r/m]	-	-	-	10	9-31	9-26
rcl reg8, imm8	1100 0000 [11-010-r/m] [imm8]	-	-	5+imm8	9	8-30	8-25
rcl reg16, imm8	1100 0001 [11-010-r/m] [imm8]	-	-	5+imm8	9	8-30	8-25
rcl reg32, imm8	0110 0110 1100 0001 [11-010-r/m] [imm8]	-	-	-	9	8-30	8-25
rcl mem8, imm8	1100 0000 [mod-010-r/m] [imm8]	-	-	8+imm8	10	9-31	10-27

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
rcl mem16, imm8	1100 0001 [mod-010-r/m] [imm8]	-	-	8+imm8	10	9-31	10-27
rcl mem32, imm8	0110 0110 1100 0001 [mod-010-r/m] [imm8]	-	-	-	10	9-31	10-27
rcr reg8, 1	1101 0000 [11-011-r/m]	2	2	2	9	3	1
rcr reg16, 1	1101 0001 [11-011-r/m]	2	2	2	9	3	1
rcr reg32, 1	0110 0110 1101 0001 [11-011-r/m]	-	-	-	9	3	1
rcr mem8, 1	1101 0000 [mod-011-r/m]	15+EA	15+EA	7	10	4	3
rcr mem16, 1	1101 0001 [mod-011-r/m]	23+EA	15+EA	7	10	4	3
rcr mem32, 1	0110 0110 1101 0001 [mod-011-r/m]	-	-	-	10	4	3
rcr reg8, cl	1101 0010 [11-011-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	9	8-30	7-24
rcr reg16, cl	1101 0011 [11-011-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	9	8-30	7-24
rcr reg32, cl	0110 0110 1101 0011 [11-011-r/m]	-	-	-	9	8-30	7-24
rcr mem8, cl	1101 0010 [mod-011-r/m]	20+EA+4*cl	20+EA+4*cl	8 + cl	10	9-31	9-26
rcr mem16, cl	1101 0011 [mod-011-r/m]	28+EA+4*cl	20+EA+4*cl	8 + cl	10	9-31	9-26
rcr mem32, cl	0110 0110 1101 0011 [mod-011-r/m]	-	-	-	10	9-31	9-26
rcr reg8, imm8	1100 0000 [11-011-r/m] [imm8]	-	-	5+imm8	9	8-30	8-25
rcr reg16, imm8	1100 0001 [11-011-r/m] [imm8]	-	-	5+imm8	9	8-30	8-25
rcr reg32, imm8	0110 0110 1100 0001 [11-011-r/m] [imm8]	-	-	-	9	8-30	8-25
rcr mem8, imm8	1100 0000 [mod-011-r/m] [imm8]	-	-	8+imm8	10	9-31	10-27
rcr mem16, imm8	1100 0001 [mod-011-r/m] [imm8]	-	-	8+imm8	10	9-31	10-27

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
rcr mem32, imm8	0110 0110 1100 0001 [mod-011-r/m] [imm8]	-	-	-	10	9-31	10-27
ret retn	1100 0011	20	16	11-14	10-13	5	2
ret imm16 retn imm16	1100 0010 [imm16]	24	20	11-14	10-13	5	3
ret retf	1100 1011	34	26	15-18	18-21	13	4
ret imm16 retf imm16	1100 1010 [imm16]	33	25	15-18	18-21	14	4
rol reg8, 1	1101 0000 [11-000-r/m]	2	2	2	3	3	1
rol reg16, 1	1101 0001 [11-000-r/m]	2	2	2	3	3	1
rol reg32, 1	0110 0110 1101 0001 [11-000-r/m]	-	-	-	3	3	1
rol mem8, 1	1101 0000 [mod-000-r/m]	15+EA	15+EA	7	7	4	3
rol mem16, 1	1101 0001 [mod-000-r/m]	23+EA	15+EA	7	7	4	3
rol mem32, 1	0110 0110 1101 0001 [mod-000-r/m]	-	-	-	7	4	3
rol reg8, cl	1101 0010 [11-000-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	3	3	4
rol reg16, cl	1101 0011 [11-000-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	3	3	4
rol reg32, cl	0110 0110 1101 0011 [11-000-r/m]	-	-	-	3	3	4
rol mem8, cl	1101 0010 [mod-000-r/m]	20+EA+4*cl	20+EA+4*cl	8 + cl	7	4	4
rol mem16, cl	1101 0011 [mod-000-r/m]	28+EA+4*cl	20+EA+4*cl	8 + cl	7	4	4
rol mem32, cl	0110 0110 1101 0011 [mod-000-r/m]	-	-	-	7	4	4
rol reg8, imm8	1100 0000 [11-000-r/m] [imm8]	-	-	5+imm8	3	2	1
rol reg16, imm8	1100 0001 [11-000-r/m] [imm8]	-	-	5+imm8	3	2	1
rol reg32, imm8	0110 0110 1100 0001 [11-000-r/m] [imm8]	-	-	-	3	2	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
rol mem8, imm8	1100 0000 [mod-000-r/m] [imm8]	-	-	8+imm8	7	4	3
rol mem16, imm8	1100 0001 [mod-000-r/m] [imm8]	-	-	8+imm8	7	4	3
rol mem32, imm8	0110 0110 1100 0001 [mod-000-r/m] [imm8]	-	-	-	7	4	3
ror reg8, 1	1101 0000 [11-001-r/m]	2	2	2	3	3	1
ror reg16, 1	1101 0001 [11-001-r/m]	2	2	2	3	3	1
ror reg32, 1	0110 0110 1101 0001 [11-001-r/m]	-	-	-	3	3	1
ror mem8, 1	1101 0000 [mod-001-r/m]	15+EA	15+EA	7	7	4	3
ror mem16, 1	1101 0001 [mod-001-r/m]	23+EA	15+EA	7	7	4	3
ror mem32, 1	0110 0110 1101 0001 [mod-001-r/m]	-	-	-	7	4	3
ror reg8, cl	1101 0010 [11-001-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	3	3	4
ror reg16, cl	1101 0011 [11-001-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	3	3	4
ror reg32, cl	0110 0110 1101 0011 [11-001-r/m]	-	-	-	3	3	4
ror mem8, cl	1101 0010 [mod-001-r/m]	20+EA+4*cl	20+EA+4*cl	8 + cl	7	4	4
ror mem16, cl	1101 0011 [mod-001-r/m]	28+EA+4*cl	20+EA+4*cl	8 + cl	7	4	4
ror mem32, cl	0110 0110 1101 0011 [mod-001-r/m]	-	-	-	7	4	4
ror reg8, imm8	1100 0000 [11-001-r/m] [imm8]	-	-	5+imm8	3	2	1
ror reg16, imm8	1100 0001 [11-001-r/m] [imm8]	-	-	5+imm8	3	2	1
ror reg32, imm8	0110 0110 1100 0001 [11-001-r/m] [imm8]	-	-	-	3	2	1
ror mem8, imm8	1100 0000 [mod-001-r/m] [imm8]	-	-	8+imm8	7	4	3

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
ror mem16, imm8	1100 0001 [mod-001-r/m] [imm8]	-	-	8+imm8	7	4	3
ror mem32, imm8	0110 0110 1100 0001 [mod-001-r/m] [imm8]	-	-	-	7	4	3
sahf	1001 1110	4	4	2	3	2	2
sal reg8, 1 (Same instruction as shl)	1101 0000 [11-100-r/m]	2	2	2	3	3	1
sal reg16, 1	1101 0001 [11-100-r/m]	2	2	2	3	3	1
sal reg32, 1	0110 0110 1101 0001 [11-100-r/m]	-	-	-	3	3	1
sal mem8, 1	1101 0000 [mod-100-r/m]	15+EA	15+EA	7	7	4	3
sal mem16, 1	1101 0001 [mod-100-r/m]	23+EA	15+EA	7	7	4	3
sal mem32, 1	0110 0110 1101 0001 [mod-100-r/m]	-	-	-	7	4	3
sal reg8, cl	1101 0010 [11-100-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	3	3	4
sal reg16, cl	1101 0011 [11-100-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	3	3	4
sal reg32, cl	0110 0110 1101 0011 [11-100-r/m]	-	-	-	3	3	4
sal mem8, cl	1101 0010 [mod-100-r/m]	20+EA+4*cl	20+EA+4*cl	8 + cl	7	4	4
sal mem16, cl	1101 0011 [mod-100-r/m]	28+EA+4*cl	20+EA+4*cl	8 + cl	7	4	4
sal mem32, cl	0110 0110 1101 0011 [mod-100-r/m]	-	-	-	7	4	4
sal reg8, imm8	1100 0000 [11-100-r/m] [imm8]	-	-	5+imm8	3	2	1
sal reg16, imm8	1100 0001 [11-100-r/m] [imm8]	-	-	5+imm8	3	2	1
sal reg32, imm8	0110 0110 1100 0001 [11-100-r/m] [imm8]	-	-	-	3	2	1
sal mem8, imm8	1100 0000 [mod-100-r/m] [imm8]	-	-	8+imm8	7	4	3

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
sal mem16, imm8	1100 0001 [mod-100-r/m] [imm8]	-	-	8+imm8	7	4	3
sal mem32, imm8	0110 0110 1100 0001 [mod-100-r/m] [imm8]	-	-	-	7	4	3
sar reg8, 1	1101 0000 [11-111-r/m]	2	2	2	3	3	1
sar reg16, 1	1101 0001 [11-111-r/m]	2	2	2	3	3	1
sar reg32, 1	0110 0110 1101 0001 [11-111-r/m]	-	-	-	3	3	1
sar mem8, 1	1101 0000 [mod-111-r/m]	15+EA	15+EA	7	7	4	3
sar mem16, 1	1101 0001 [mod-111-r/m]	23+EA	15+EA	7	7	4	3
sar mem32, 1	0110 0110 1101 0001 [mod-111-r/m]	-	-	-	7	4	3
sar reg8, cl	1101 0010 [11-111-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	3	3	4
sar reg16, cl	1101 0011 [11-111-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	3	3	4
sar reg32, cl	0110 0110 1101 0011 [11-111-r/m]	-	-	-	3	3	4
sar mem8, cl	1101 0010 [mod-111-r/m]	20+EA+4*cl	20+EA+4*cl	8 + cl	7	4	4
sar mem16, cl	1101 0011 [mod-111-r/m]	28+EA+4*cl	20+EA+4*cl	8 + cl	7	4	4
sar mem32, cl	0110 0110 1101 0011 [mod-111-r/m]	-	-	-	7	4	4
sar reg8, imm8	1100 0000 [11-111-r/m] [imm8]	-	-	5+imm8	3	2	1
sar reg16, imm8	1100 0001 [11-111-r/m] [imm8]	-	-	5+imm8	3	2	1
sar reg32, imm8	0110 0110 1100 0001 [11-111-r/m] [imm8]	-	-	-	3	2	1
sar mem8, imm8	1100 0000 [mod-111-r/m] [imm8]	-	-	8+imm8	7	4	3

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
sar mem16, imm8	1100 0001 [mod-111-r/m] [imm8]	-	-	8+imm8	7	4	3
sar mem32, imm8	0110 0110 1100 0001 [mod-111-r/m] [imm8]	-	-	-	7	4	3
sbb reg8, reg8	0001 10x0 [11-reg-r/m]	3	‘3	2	2	1	1
sbb reg16, reg16	0001 10x1 [11-reg-r/m]	3	3	2	2	1	1
sbb reg32, reg32	0110 0110 0001 10x1 [11-reg-r/m]	3	3	2	2	1	1
sbb reg8, mem8	0001 1010 [mod-reg-r/m]	9+EA	9+EA	7	7	2	2
sbb reg16, mem16	0001 1011 [mod-reg-r/m]	13+EA	9+EA	7	7	2	2
sbb reg32, mem32	0110 0110 0001 1011 [mod-reg-r/m]	-	-	-	7	2	2
sbb mem8, reg8	0001 1000 [mod-reg-r/m]	16+EA	16+EA	7	6	3	3
sbb mem16, reg16	0001 1001 [mod-reg-r/m]	24+EA	16+EA	7	6	3	3
sbb mem32, reg32	0110 0110 0001 1001 [mod-reg-r/m]	-	-	-	6	3	3
sbb reg8, imm8	1000 00x0 [11-011-r/m] [imm]	4	4	3	2	1	1
sbb reg16, imm16	1000 00s1 [11-011-r/m] [imm]	4	4	3	2	1	1
sbb reg32, imm32	0110 0110 1000 00s1 [11-011-r/m] [imm]	4	4	3	2	1	1
sbb mem8, imm8	1000 00x0 [mod-011-r/m] [imm]	17+EA	17+EA	7	7	3	3
sbb mem16, imm16	1000 00s1 [mod-011-r/m] [imm]	25+EA	17+EA	7	7	3	3
sbb mem32, imm32	0110 0110 1000 00s1 [mod-011-r/m] [imm]	-	-	-	7	3	3
sbb al, imm	0001 1100 [imm]	4	4	3	2	1	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
sbb ax, imm	0001 1101 [imm]	4	4	3	2	1	1
sbb eax, imm	0110 0110 0001 1101 [imm]	-	-	-	2	1	1
scasb	1010 0100	15	15	7	8	6	4
scasw	1010 0101	19	15	7	8	6	4
scasd	0110 0110 1010 0101	-	-	-	8	6	4
rep scasb	1111 0010 1010 0100	9 + 15 * cx	9 + 15*cx	5 + 8*cx	5 + 8*cx	7 + 5*cx 5 if cx=0	9 + 4*cx 7 if cx=0
rep scasw	1111 0010 1010 0101	9 + 19 * cx	9 + 15*cx	5 + 8*cx	5 + 8*cx	7 + 5*cx 5 if cx=0	9 + 4*cx 7 if cx=0
rep scasd	0110 0110 1111 0010 1010 0101	-	-	-	5 + 8*cx	7 + 5*cx 5 if cx=0	9 + 4*cx 7 if cx=0
seta reg8	0000 1111 1001 0111 [11-000-r/m][e]	-	-	-	4	4 if set 3 if clear	1
seta mem8	0000 1111 1001 0011 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setae reg8	0000 1111 1001 0011 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setae mem8	0000 1111 1001 0011 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setb reg8	0000 1111 1001 0010 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setb mem8	0000 1111 1001 0010 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setbe reg8	0000 1111 1001 0110 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setbe mem8	0000 1111 1001 0110 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setc reg8	0000 1111 1001 0010 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setc mem8	0000 1111 1001 0010 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
sete reg8	0000 1111 1001 0100 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
sete mem8	0000 1111 1001 0100 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setg reg8	0000 1111 1001 1111 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setg mem8	0000 1111 1001 1111 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setge reg8	0000 1111 1001 1101 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setge mem8	0000 1111 1001 1101 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setl reg8	0000 1111 1001 1100 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setl mem8	0000 1111 1001 1100 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setle reg8	0000 1111 1001 1110 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setle mem8	0000 1111 1001 1110 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setna reg8	0000 1111 1001 0110 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setna mem8	0000 1111 1001 0110 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setnae reg8	0000 1111 1001 0010 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setnae mem8	0000 1111 1001 0010 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setnb reg8	0000 1111 1001 0011 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setnb mem8	0000 1111 1001 0011 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setnbe reg8	0000 1111 1001 0111 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setnbe mem8	0000 1111 1001 0111 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
setnc reg8	0000 1111 1001 0011 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setnc mem8	0000 1111 1001 0011 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setne reg8	0000 1111 1001 0101 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setne mem8	0000 1111 1001 0101 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setng reg8	0000 1111 1001 1110 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setng mem8	0000 1111 1001 1110 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setnge reg8	0000 1111 1001 1100 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setnge mem8	0000 1111 1001 1100 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setnl reg8	0000 1111 1001 1101 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setnl mem8	0000 1111 1001 1101 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setnle reg8	0000 1111 1001 1111 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setnle mem8	0000 1111 1001 1111 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setno reg8	0000 1111 1001 0001 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setno mem8	0000 1111 1001 0001 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setnp reg8	0000 1111 1001 1011 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setnp mem8	0000 1111 1001 1011 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setns reg8	0000 1111 1001 1001 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
setns mem8	0000 1111 1001 1001 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setnz reg8	0000 1111 1001 0101 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setnz mem8	0000 1111 1001 0101 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
seto reg8	0000 1111 1001 0000 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
seto mem8	0000 1111 1001 0000 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setp reg8	0000 1111 1001 1010 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setp mem8	0000 1111 1001 1010 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setpe reg8	0000 1111 1001 1010 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setpe mem8	0000 1111 1001 1010 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setpo reg8	0000 1111 1001 1011 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setpo mem8	0000 1111 1001 1011 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
sets reg8	0000 1111 1001 1000 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
sets mem8	0000 1111 1001 1000 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
setz reg8	0000 1111 1001 0100 [11-000-r/m]	-	-	-	4	4 if set 3 if clear	1
setz mem8	0000 1111 1001 0100 [mod-000-r/m]	-	-	-	5	3 if set 4 if clear	2
shl reg8, 1	1101 0000 [11-100-r/m]	2	2	2	3	3	1
shl reg16, 1	1101 0001 [11-100-r/m]	2	2	2	3	3	1
shl reg32, 1	0110 0110 1101 0001 [11-100-r/m]	-	-	-	3	3	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
shl mem8, 1	1101 0000 [mod-100-r/m]	15+EA	15+EA	7	7	4	3
shl mem16, 1	1101 0001 [mod-100-r/m]	23+EA	15+EA	7	7	4	3
shl mem32, 1	0110 0110 1101 0001 [mod-100-r/m]	-	-	-	7	4	3
shl reg8, cl	1101 0010 [11-100-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	3	3	4
shl reg16, cl	1101 0011 [11-100-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	3	3	4
shl reg32, cl	0110 0110 1101 0011 [11-100-r/m]	-	-	-	3	3	4
shl mem8, cl	1101 0010 [mod-100-r/m]	20+EA+4*cl	20+EA+4*cl	8 + cl	7	4	4
shl mem16, cl	1101 0011 [mod-100-r/m]	28+EA+4*cl	20+EA+4*cl	8 + cl	7	4	4
shl mem32, cl	0110 0110 1101 0011 [mod-100-r/m]	-	-	-	7	4	4
shl reg8, imm8	1100 0000 [11-100-r/m] [imm8]	-	-	5+imm8	3	2	1
shl reg16, imm8	1100 0001 [11-100-r/m] [imm8]	-	-	5+imm8	3	2	1
shl reg32, imm8	0110 0110 1100 0001 [11-100-r/m] [imm8]	-	-	-	3	2	1
shl mem8, imm8	1100 0000 [mod-100-r/m] [imm8]	-	-	8+imm8	7	4	3
shl mem16, imm8	1100 0001 [mod-100-r/m] [imm8]	-	-	8+imm8	7	4	3
shl mem32, imm8	0110 0110 1100 0001 [mod-100-r/m] [imm8]	-	-	-	7	4	3
shld reg16, reg16, imm8 r/m is 1st operand, reg is second operand.	0000 1111 1010 0100 [11-reg-r/m] [imm8]	-	-	-	3	2	4
shld reg32, reg32, imm8 r/m is 1st operand, reg is second operand.	0110 0110 0000 1111 1010 0100 [11-reg-r/m] [imm8]	-	-	-	3	2	4

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
shld mem16, reg16, imm8	0000 1111 1010 0100 [mod-reg-r/m] [imm8]	-	-	-	7	3	4
shld mem32, reg32, imm8	0110 0110 0000 1111 1010 0100 [mod-reg-r/m] [imm8]	-	-	-	7	3	4
shld reg16, reg16, cl r/m is 1st operand, reg is second operand.	0000 1111 1010 0101 [11-reg-r/m]	-	-	-	3	3	4
shld reg32, reg32, cl r/m is 1st operand, reg is second operand.	0110 0110 0000 1111 1010 0101 [11-reg-r/m]	-	-	-	3	3	4
shld mem16, reg16, cl	0000 1111 1010 0101 [mod-reg-r/m]	-	-	-	7	4	5
shld mem32, reg32, cl	0110 0110 0000 1111 1010 0101 [mod-reg-r/m]	-	-	-	7	4	5
shr reg8, 1	1101 0000 [11-101-r/m]	2	2	2	3	3	1
shr reg16, 1	1101 0001 [11-101-r/m]	2	2	2	3	3	1
shr reg32, 1	0110 0110 1101 0001 [11-101-r/m]	-	-	-	3	3	1
shr mem8, 1	1101 0000 [mod-101-r/m]	15+EA	15+EA	7	7	4	3
shr mem16, 1	1101 0001 [mod-101-r/m]	23+EA	15+EA	7	7	4	3
shr mem32, 1	0110 0110 1101 0001 [mod-101-r/m]	-	-	-	7	4	3
shr reg8, cl	1101 0010 [11-101-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	3	3	4
shr reg16, cl	1101 0011 [11-101-r/m]	8 + 4*cl	8 + 4*cl	5 + cl	3	3	4
shr reg32, cl	0110 0110 1101 0011 [11-101-r/m]	-	-	-	3	3	4
shr mem8, cl	1101 0010 [mod-101-r/m]	20+EA+4*cl	20+EA+4*cl	8 + cl	7	4	4
shr mem16, cl	1101 0011 [mod-101-r/m]	28+EA+4*cl	20+EA+4*cl	8 + cl	7	4	4
shr mem32, cl	0110 0110 1101 0011 [mod-101-r/m]	-	-	-	7	4	4

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
shr reg8, imm8	1100 0000 [11-101-r/m] [imm8]	-	-	5+imm8	3	2	1
shr reg16, imm8	1100 0001 [11-101-r/m] [imm8]	-	-	5+imm8	3	2	1
shr reg32, imm8	0110 0110 1100 0001 [11-101-r/m] [imm8]	-	-	-	3	2	1
shr mem8, imm8	1100 0000 [mod-101-r/m] [imm8]	-	-	8+imm8	7	4	3
shr mem16, imm8	1100 0001 [mod-101-r/m] [imm8]	-	-	8+imm8	7	4	3
shr mem32, imm8	0110 0110 1100 0001 [mod-101-r/m] [imm8]	-	-	-	7	4	3
shrd reg16, reg16, imm8 r/m is 1st operand, reg is second operand.	0000 1111 1010 1100 [11-reg-r/m] [imm8]	-	-	-	3	2	4
shrd reg32, reg32, imm8 r/m is 1st operand, reg is second operand.	0110 0110 0000 1111 1010 1100 [11-reg-r/m] [imm8]	-	-	-	3	2	4
shrd mem16, reg16, imm8	0000 1111 1010 1100 [mod-reg-r/m] [imm8]	-	-	-	7	3	4
shrd mem32, reg32, imm8	0110 0110 0000 1111 1010 1100 [mod-reg-r/m] [imm8]	-	-	-	7	3	4
shrd reg16, reg16, cl r/m is 1st operand, reg is second operand.	0000 1111 1010 1101 [11-reg-r/m]	-	-	-	3	3	4
shrd reg32, reg32, cl r/m is 1st operand, reg is second operand.	0110 0110 0000 1111 1010 1101 [11-reg-r/m]	-	-	-	3	3	4
shrd mem16, reg16, cl	0000 1111 1010 1101 [disp]	-	-	-	7	4	5
shld mem32, reg32, cl	0110 0110 0000 1111 1010 1101 [mod-reg-r/m]	-	-	-	7	4	5

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
stc	1111 1001	2	2	2	2	2	2
std	1111 1101	2	2	2	2	2	2
sti	1111 1011	2	2	2	3	5	7
stosb	1010 1010	11	11	3	4	5	3
stosw	1010 1011	15	11	3	4	5	3
stosd	0110 0110 1010 1011	-	-	-	4	5	3
rep stosb	1111 0010 1010 1010	9 + 10 * cx	9 + 10*cx	4 + 3*cx	5 + 5*cx	7 + 5*cx 5 if cx=0	9 + 3*cx 6 if cx=0
rep stosw	1111 0010 1010 1011	9 + 14 * cx	9 + 10*cx	4 + 3*cx	5 + 5*cx	7 + 5*cx 5 if cx=0	9 + 3*cx 6 if cx=0
rep stosd	0110 0110 1111 0010 1010 1011	-	-	-	5 + 5*cx	7 + 5*cx 5 if cx=0	9 + 3*cx 6 if cx=0
sub reg8, reg8	0010 10x0 [11-reg-r/m]	3	‘3	2	2	1	1
sub reg16, reg16	0010 10x1 [11-reg-r/m]	3	3	2	2	1	1
sub reg32, reg32	0110 0110 0010 10x1 [11-reg-r/m]	3	3	2	2	1	1
sub reg8, mem8	0010 1010 [mod-reg-r/m]	9+EA	9+EA	7	7	2	2
sub reg16, mem16	0010 1011 [mod-reg-r/m]	13+EA	9+EA	7	7	2	2
sub reg32, mem32	0110 0110 0010 1011 [mod-reg-r/m]	-	-	-	7	2	2
sub mem8, reg8	0010 1000 [mod-reg-r/m]	16+EA	16+EA	7	6	3	3
sub mem16, reg16	0010 1001 [mod-reg-r/m]	24+EA	16+EA	7	6	3	3
sub mem32, reg32	0110 0110 0010 1001 [mod-reg-r/m]	-	-	-	6	3	3
sub reg8, imm8	1000 00x0 [11-101-r/m] [imm]	4	4	3	2	1	1
sub reg16, imm16	1000 00s1 [11-101-r/m] [imm]	4	4	3	2	1	1
sub reg32, imm32	0110 0110 1000 00s1 [11-101-r/m] [imm]	4	4	3	2	1	1
sub mem8, imm8	1000 00x0 [mod-101-r/m] [imm]	17+EA	17+EA	7	7	3	3

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
sub mem16, imm16	1000 00s1 [mod-101-r/m] [imm]	25+EA	17+EA	7	7	3	3
sub mem32, imm32	0110 0110 1000 00s1 [mod-101-r/m] [imm]	-	-	-	7	3	3
sub al, imm	0010 1100 [imm]	4	4	3	2	1	1
sub ax, imm	0010 1101 [imm]	4	4	3	2	1	1
sub eax, imm	0110 0110 0010 1101 [imm]	-	-	-	2	1	1
test reg8, reg8	1000 0100 [11-reg-r/m]	3	‘3	2	2	1	1
test reg16, reg16	1000 0101 [11-reg-r/m]	3	3	2	2	1	1
test reg32, reg32	0110 0110 1000 0101 [11-reg-r/m]	3	3	2	2	1	1
test reg8, mem8	1000 0110 [mod-reg-r/m]	9+EA	9+EA	6	5	2	2
test reg16, mem16	1000 0111 [mod-reg-r/m]	13+EA	9+EA	6	5	2	2
test reg32, mem32	0110 0110 1000 0111 [mod-reg-r/m]	-	-	-	5	2	2
test reg8, imm8	1111 0110 [11-000-r/m] [imm]	4	4	3	2	1	1
test reg16, imm16	1111 0111 [11-000-r/m] [imm]	4	4	3	2	1	1
test reg32, imm32	0110 0110 1111 0111 [11-000-r/m] [imm]	4	4	3	2	1	1
test mem8, imm8	1111 0110 [mod-000-r/m] [imm]	9+EA	9+EA	6	5	2	2
test mem16, imm16	1111 0111 [mod-000-r/m] [imm]	13+EA	9+EA	6	5	2	2
test mem32, imm32	0110 0110 1111 0111 [mod-000-r/m] [imm]	-	-	-	5	2	2
test al, imm	1010 1000 [imm]	4	4	3	2	1	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
test ax, imm	1010 1001 [imm]	4	4	3	2	1	1
test eax, imm	0110 0110 1010 1001 [imm]	-	-	-	2	1	1
xadd reg8, reg8 r/m is first operand, reg is second operand.	0000 1111 1100 0000 [11-reg-r/m]	-	-	-	-	3	3
xadd reg16, reg16	0000 1111 1100 0001 [11-reg-r/m]	-	-	-	-	3	3
xadd reg32, reg32	0110 0110 0000 1111 1100 0001 [11-reg-r/m]	-	-	-	-	3	3
xadd mem8, reg8	0000 1111 1100 0000 [mod-reg-r/m]	-	-	-	-	4	4
xadd mem16, reg16	0000 1111 1100 0001 [mod-reg-r/m]	-	-	-	-	4	4
xadd mem32, reg32	0110 0110 0000 1111 1100 0001 [mod-reg-r/m]	-	-	-	-	4	4
xchg reg8, reg8	1000 0110 [11-reg-r/m]	4	4	3	3	3	3
xchg reg16, reg16	1000 0111 [11-reg-r/m]	4	4	3	3	3	3
xchg reg32, reg32	0110 0110 1000 0111 [11-reg-r/m]	-	-	-	3	3	3
xchg mem8, reg8[f]	1000 0110 [11-reg-r/m]	17 + EA	17 + EA	5	5	5	3
xchg mem16, reg16	1000 0111 [11-reg-r/m]	25 + EA	17 + EA	5	5	5	3
xchg mem32, reg32	0110 0110 1000 0111 [11-reg-r/m]	-	-	-	5	5	3
xchg ax, reg16	1001 0rrr	3	3	3	3	3 1 if reg=ax	2 1 if reg=ax
xchg ax, reg32	0110 0110 1001 0rrr	3	3	3	3	3	2
xlat	1101 0111	11	11	5	5	4	4
xor reg8, reg8	0011 00x0 [11-reg-r/m]	3	‘3	2	2	1	1
xor reg16, reg16	0011 00x1 [11-reg-r/m]	3	3	2	2	1	1

Table 97: 80x86 Instruction Set Reference[a]

Instruction	Encoding (bin)[b]	Execution Time in Cycles[c]					
		8088	8086	80286	80386	80486	Pentium
xor reg32, reg32	0110 0110 0011 00x1 [11-reg-r/m]	3	3	2	2	1	1
xor reg8, mem8	0011 0010 [mod-reg-r/m]	9+EA	9+EA	7	7	2	2
xor reg16, mem16	0011 0011 [mod-reg-r/m]	13+EA	9+EA	7	7	2	2
xor reg32, mem32	0110 0110 0011 0011 [mod-reg-r/m]	-	-	-	7	2	2
xor mem8, reg8	0011 0000 [mod-reg-r/m]	16+EA	16+EA	7	6	3	3
xor mem16, reg16	0011 0001 [mod-reg-r/m]	24+EA	16+EA	7	6	3	3
xor mem32, reg32	0110 0110 0011 0001 [mod-reg-r/m]	-	-	-	6	3	3
xor reg8, imm8	1000 00x0 [11-110-r/m] [imm]	4	4	3	2	1	1
xor reg16, imm16	1000 00s1 [11-110-r/m] [imm]	4	4	3	2	1	1
xor reg32, imm32	0110 0110 1000 00s1 [11-110-r/m] [imm]	4	4	3	2	1	1
xor mem8, imm8	1000 00x0 [mod-110-r/m] [imm]	17+EA	17+EA	7	7	3	3
xor mem16, imm16	1000 00s1 [mod-110-r/m] [imm]	25+EA	17+EA	7	7	3	3
xor mem32, imm32	0110 0110 1000 00s1 [mod-110-r/m] [imm]	-	-	-	7	3	3
xor al, imm	0011 0100 [imm]	4	4	3	2	1	1
xor ax, imm	0011 0101 [imm]	4	4	3	2	1	1
xor eax, imm	0110 0110 0011 0101 [imm]	-	-	-	2	1	1

a. Real mode, 16-bit segments.
b. Instructions with a 66h or 67h prefix are available only on 80386 and later processors.
c. Timings are all optimistic and do not include the cost of prefix bytes, hazards, fetching, misaligned operands, etc.
d. Cycle timings for HLT instruction are above and beyond the time spent waiting for an interrupt to occur.

e. On the 80386 and most versions of later processors, the processor ignores the *reg* field's value for the S*cc* instruction; the reg field, however, should contain zero.
f. Most assemblers accept "xchg reg,mem" and encode it as "xchg mem,reg" which does the same thing.