



CPU Instruction Set Summary

Table 6-20. 6x86 CPU Instruction Set Clock Count Summary

INSTRUCTION	OPCODE	FLAGS				REAL MODE CLOCK COUNT		PROTECTED MODE CLOCK COUNT		NOTES			
		OF	DF	IF	TF	SF	ZF	AF	PF	CF	Reg/Cache Hit	Cache Hit	Real Mode
AAA ASCII Adjust AL after Add	37	u	-	-	-	u	x	x	u	x	7		
AAD ASCII Adjust AX before Divide	D5 0A	u	-	-	-	x	x	u	x	u	7		
AAM ASCII Adjust AX after Multiply	D4 0A	u	-	-	-	x	x	u	x	u	13-21		
AAS ASCII Adjust AL after Subtract	3F	u	-	-	-	u	x	x	u	x	7		
ADC Add with Carry Register to Register	1 [00dw] [11 reg r/m]	x	-	-	-	x	x	x	x	x	1	b	h
Register to Memory	1 [000w] [mod reg r/m]										1		
Memory to Register	1 [001w] [mod reg r/m]										1		
Immediate to Register/Memory	8 [00sw] [mod 010 r/m]###										1		
Immediate to Accumulator	1 [010w] ###										1		
ADD Integer Add Register to Register	0 [00dw] [11 reg r/m]	x	-	-	-	x	x	x	x	x	1	b	h
Register to Memory	0 [000w] [mod reg r/m]										1		
Memory to Register	0 [001w] [mod reg r/m]										1		
Immediate to Register/Memory	8 [00sw] [mod 000 r/m]###										1		
Immediate to Accumulator	0 [010w] ###										1		
AND Boolean AND Register to Register	2 [00dw] [11 reg r/m]	0	-	-	-	x	x	u	x	0	1	b	h
Register to Memory	2 [000w] [mod reg r/m]										1		
Memory to Register	2 [001w] [mod reg r/m]										1		
Immediate to Register/Memory	8 [00sw] [mod 100 r/m]###										1		
Immediate to Accumulator	2 [010w] ###										1		
ARPL Adjust Requested Privilege Level From Register/Memory	63 [mod reg r/m]	-	-	-	-	-	-	-	-	-	9	a	h
BOUND Check Array Boundaries If Out of Range (Int 5) If In Range	62 [mod reg r/m]	-	-	-	-	-	-	-	-	-	20 11	b, e	g, h, j, k, r
BSF Scan Bit Forward Register, Register/Memory	0F BC [mod reg r/m]	-	-	-	-	-	-	-	-	-	3	b	h
BSR Scan Bit Reverse Register, Register/Memory	0F BD [mod reg r/m]	-	-	-	-	-	-	-	-	-	3	b	h
BSWAP Byte Swap Register, Register/Memory	0F C1 reg]	-	-	-	-	-	-	-	-	-	4		
BT Test Bit Register/Memory, Immediate	0F BA [mod 100 r/m]#	-	-	-	-	-	-	-	-	-	2	b	h
Register/Memory, Register	0F A3 [mod reg r/m]										5/6		
BTC Test Bit and Complement Register/Memory, Immediate	0F BA [mod 111 r/m]#	-	-	-	-	-	-	-	-	-	3	b	h
Register/Memory, Register	0F BB [mod reg r/m]										5/6		

+ = 8-bit signed displacement
 +++ = full signed displacement (16, 32 bits)
 # = immediate 8-bit data
 ## = immediate 16-bit data
 ### = full immediate 32-bit data (8, 16, 32 bits)
 x = modified
 - = unchanged
 u = undefined

Table 6-20. 6x86 CPU Instruction Set Clock Count Summary (Continued)

INSTRUCTION	OPCODE	FLAGS				REAL	PROTECTED	NOTES							
		OF	DF	IF	TF	SF	ZF	AF	PF	CF	REAL MODE	Cache Hit	PROTECTED MODE		
BTR <i>Test Bit and Reset Register/Memory, Immediate Register/Memory, Register</i>	0F BA [mod 110 r/m]# 0F B3 [mod reg r/m]	-	-	-	-	-	-	-	-	-	3 5/6	Reg/ Cache Hit	b	h	
BTS <i>Test Bit and Set Register/Memory Register (short form)</i>	0F BA [mod 101 r/m] 0F AB [mod reg r/m]	-	-	-	-	-	-	-	-	-	3 5/6	3 5/6	b	h	
CALL <i>Subroutine Call</i> Direct Within Segment Register/Memory Indirect Within Segment Direct Intersegment Call Gate to Same Privilege Call Gate to Different Privilege No Parameters Call Gate to Different Privilege m Par's 16-bit Task to 16-bit TSS 16-bit Task to 32-bit TSS 16-bit Task to Y86 Task 32-bit Task to 16-bit TSS 32-bit Task to 32-bit TSS 32-bit Task to Y86 Task Indirect Intersegment Call Gate to Same Privilege Call Gate to Different Privilege No Parameters Call Gate to Different Privilege Level m Par's 16-bit Task to 16-bit TSS 16-bit Task to 32-bit TSS 16-bit Task to Y86 Task 32-bit Task to 16-bit TSS 32-bit Task to 32-bit TSS 32-bit Task to Y86 Task	E8 +++ FF [mod 010 r/m] 9A [unsigned full offset, selector]	-	-	-	-	-	-	-	-	-	1 1/3 4 15 26 35+2m 110 118 96 112 120 98 8 20 31 40+2m 114 122 100 116 124 102	1 1/3 3	5	b	h,j,k,r
CBW <i>Convert Byte to Word</i>	98	-	-	-	-	-	-	-	-	-	3	3			
CDQ <i>Convert Doubleword to Quadword</i>	99	-	-	-	-	-	-	-	-	-	2	2			
CLC <i>Clear Carry Flag</i>	F8	-	-	-	-	-	-	-	-	-	1	1			
CLD <i>Clear Direction Flag</i>	FC	-	-	-	-	-	-	-	-	-	7	7			
CLI <i>Clear Interrupt Flag</i>	FA	-	-	-	-	-	-	-	-	-	7	7			m
CLTS <i>Clear Task Switched Flag</i>	0F 06	-	-	-	-	-	-	-	-	-	10	10	c	1	
CMC <i>Complement the Carry Flag</i>	F5	-	-	-	-	-	-	-	-	-	2	2			

= immediate 8-bit data
= immediate 16-bit data
= full immediate 32-bit data (8, 16, 32 bits)
+ = 8-bit signed displacement
+++ = full signed displacement (16, 32 bits)
x = modified
- = unchanged
u = undefined



CPU Instruction Set Summary

Table 6-20. 6x86 CPU Instruction Set Clock Count Summary (Continued)

INSTRUCTION	OPCODE	FLAGS			REAL MODE CLOCK COUNT		PROTECTED MODE CLOCK COUNT		NOTES						
		OF	DF	IF	TF	SF	ZF	AF	PF	CF	Reg/Cache Hit	Cache Hit	Real Mode	Protected Mode	
CMP Compare Integers Register to Register Register to Memory Memory to Register Immediate to Register/Memory Immediate to Accumulator	3 [10dw] [11 reg r/m]	x	-	-	-	x	x	x	x	x	x	1		b	h
	3 [101w] [mod reg r/m]											1			
	3 [100w] [mod reg r/m]											1			
	8 [00sw] [mod 111 r/m] ### 3 [110w] ###												1		
CMPS Compare String	A [011w]	x	-	-	-	x	x	x	x	x	x	5		b	h
CMPXCHG Compare and Exchange Register1, Register2 Memory, Register	0F B [000w] [11 reg2 reg1]	x	-	-	-	x	x	x	x	x	x	11			
	0F B [000w] [mod reg r/m]											11			
CPUID CPU Identification	0F A2	-	-	-	-	-	-	-	-	-	-	12			
CWD Convert Word to Doubleword	99	-	-	-	-	-	-	-	-	-	-	2			
CWDE Convert Word to Doubleword Extended	98	-	-	-	-	-	-	-	-	-	-	2			
DAA Decimal Adjust AL after Add	27	-	-	-	-	x	x	x	x	x	x	9			
DAS Decimal Adjust AL after Subtract	2F	-	-	-	-	x	x	x	x	x	x	9			
DEC Decrement by 1 Register/Memory Register (short form)	F [111w] [mod 001 r/m]	x	-	-	-	x	x	x	x	x	-	1		b	h
	4 [1 reg]											1			
DIV Unsigned Divide Accumulator by Register/Memory Divisor: Byte Word Doubleword	F [011w] [mod 110 r/m]	-	-	-	-	x	x	u	u	-	-	13-17 13-25 13-41		b,e	e,h
	C8 ###											10 13 10+L*3		b	h
												5			
ENTER Enter New Stack Frame Level = 0 Level = 1 Level (L) > 1	F4	-	-	-	-	-	-	-	-	-	-	5			1
	F [011w] [mod 111 r/m]					x	x	u	u	-	-	16-20 16-28 17-45		b,e	e,h
IDIV Integer (Signed) Divide Accumulator by Register/Memory Divisor: Byte Word Doubleword															

= immediate 8-bit data
 ## = immediate 16-bit data
 ### = full immediate 32-bit data (8, 16, 32 bits)
 + = 8-bit signed displacement
 +++ = full signed displacement (16, 32 bits)
 x = modified
 - = unchanged
 u = undefined

Table 6-20. 6x86 CPU Instruction Set Clock Count Summary (Continued)

INSTRUCTION	OPCODE	FLAGS			REAL MODE CLOCK COUNT	PROTECTED MODE CLOCK COUNT	NOTES							
		OF	DF	IF	TF	SF	ZF	AF	PF	CF	Reg/Cache Hit	Reg/Cache Hit	Real Mode	Protected Mode
IMUL <i>Integer (Signed) Multiply</i> Accumulator by Register/Memory Multiplier: Byte Word Doubleword Register with Register/Memory Multiplier: Word Doubleword Register/Memory with Immediate to Register2 Multiplier: Word Doubleword	F [011w] [mod 101 r/m]	x	-	-	-	x	x	u	u	x	4 4 10	4 4 10	b	h
	0F AF [mod reg r/m]										4 10	4 10		
	6 [10s1] [mod reg r/m] ###										5 11	5 11		
IN <i>Input from I/O Port</i> Fixed Port Variable Port	E [010w] [#] E [110w]	-	-	-	-	-	-	-	-	-	14 14	14/28 14/28		m
	F [111w] [mod 000 r/m] 4 [0 reg]	x	-	-	-	x	x	x	x	-	1 1	1 1	b	h
INS <i>Input String from I/O Port</i>	6 [110w]	-	-	-	-	-	-	-	-	-	14	14/28	b	h,m
INT <i>Software Interrupt</i> INT i Protected Mode: Interrupt or Trap to Same Privilege Interrupt or Trap to Different Privilege 16-bit Task to 16-bit TSS by Task Gate 16-bit Task to 32-bit TSS by Task Gate 16-bit Task to V86 by Task Gate 16-bit Task to 16-bit TSS by Task Gate 32-bit Task to 32-bit TSS by Task Gate 32-bit Task to V86 by Task Gate V86 to 16-bit TSS by Task Gate V86 to 32-bit TSS by Task Gate V86 to Privilege 0 by Trap Gate/Int Gate	CD #	-	-	x	0	-	-	-	-	-	9		b,e	g,j,k,r
											21 32 114 122 100 116 124 102 124 102 102 46			

Continued on the next page...
 # = immediate 8-bit data
 ## = immediate 16-bit data
 ### = full immediate 32-bit data (8, 16, 32 bits)
 + = 8-bit signed displacement
 +++ = full signed displacement (16, 32 bits)
 x = modified
 - = unchanged
 u = undefined



CPU Instruction Set Summary

Table 6-20. 6x86 CPU Instruction Set Clock Count Summary (Continued)

INSTRUCTION	OPCODE	FLAGS	REAL MODE CLOCK COUNT		PROTECTED MODE CLOCK COUNT		NOTES	
			Reg/Cache Hit	Cache Hit	Reg/Cache Hit	Cache Hit	Real Mode	Protected Mode
INT Software Interrupt (Continued)								
INT 3	CC							
INTO	CE							
If OF==0								
If OF==1 (INT 4)								
INVD Invalidate Cache	0F 08							
INVLPG Invalidate TLB Entry	0F 01 [mod 111 r/m]							
IRET Interrupt Return	CF							
Real Mode								
Protected Mode:								
Within Task to Same Privilege								
Within Task to Different Privilege								
16-bit Task to 16-bit Task								
16-bit Task to 32-bit TSS								
16-bit Task to V86 Task								
32-bit Task to 16-bit TSS								
32-bit Task to 32-bit TSS								
32-bit Task to V86 Task								
JB/JNAE/JC Jump on Below/Not Above or Equal/Carry	72 + 0F 82 +++							
8-bit Displacement								
Full Displacement								
JBE/JNA Jump on Below or Equal/Not Above	76 + 0F 86 +++							
8-bit Displacement								
Full Displacement								
JCXZ/JECXZ Jump on CX/ECX Zero	E3 +							
JE/JZ Jump on Equal/Zero	74 + 0F 84 +++							
8-bit Displacement								
Full Displacement								
JL/JNGE Jump on Less/Not Greater or Equal	7C + 0F 8C +++							
8-bit Displacement								
Full Displacement								
JLE/JNG Jump on Less or Equal/Not Greater	7E + 0F 8E +++							
8-bit Displacement								
Full Displacement								

= immediate 8-bit data
= immediate 16-bit data
= full immediate 32-bit data (8, 16, 32 bits)
+ = 8-bit signed displacement
+++ = full signed displacement (16, 32 bits)
x = modified
- = unchanged
u = undefined

Table 6-20. 6x86 CPU Instruction Set Clock Count Summary (Continued)

INSTRUCTION	OPCODE	FLAGS			REAL MODE CLOCK COUNT		PROTECTED MODE CLOCK COUNT		NOTES					
		OF	DF	IF	TF	SF	ZF	AF	PF	CF	Reg/Cache Hit	Cache Hit	Real Mode	Protected Mode
JMP Unconditional Jump 8-bit Displacement Full Displacement Register/Memory Indirect Within Segment Direct Intersegment	EB +	-	-	-	-	-	-	-	-	-	1	1	b	h,j,k,r
	E9 +++	-	-	-	-	-	-	-	-	-	1	1		
	FF [mod 100 r/m]	-	-	-	-	-	-	-	-	-	1/3	1/3		
	EA [unsigned full offset, selector]	-	-	-	-	-	-	-	-	-	1	4		
	FF [mod 101 r/m]	-	-	-	-	-	-	-	-	-	5	14 110 118 96 112 120 98 7 17 113 121 99 115 123 101		
Indirect Intersegment Call Gate Same Privilege Level	FF [mod 101 r/m]	-	-	-	-	-	-	-	-	-	5			
	FF [mod 101 r/m]	-	-	-	-	-	-	-	-	-	5			
JNB/JAE/JNC Jump on Not Below/Above or Equal/Not Carry 8-bit Displacement Full Displacement	73 +	-	-	-	-	-	-	-	-	-	1	1		r
	0F 83 +++	-	-	-	-	-	-	-	-	-	1	1		
JNBE/JA Jump on Not Below or Equal/Above 8-bit Displacement Full Displacement	77 +	-	-	-	-	-	-	-	-	-	1	1		r
	0F 87 +++	-	-	-	-	-	-	-	-	-	1	1		
JNE/JNZ Jump on Not Equal/Not Zero 8-bit Displacement Full Displacement	75 +	-	-	-	-	-	-	-	-	-	1	1		r
	0F 85 +++	-	-	-	-	-	-	-	-	-	1	1		
JNL/JGE Jump on Not Less/Greater or Equal 8-bit Displacement Full Displacement	7D +	-	-	-	-	-	-	-	-	-	1	1		r
	0F 8D +++	-	-	-	-	-	-	-	-	-	1	1		
JNLE/JG Jump on Not Less or Equal/Greater 8-bit Displacement Full Displacement	7F +	-	-	-	-	-	-	-	-	-	1	1		r
	0F 8F +++	-	-	-	-	-	-	-	-	-	1	1		

= immediate 8-bit data
= immediate 16-bit data
= full immediate 32-bit data (8, 16, 32 bits)
+ = 8-bit signed displacement
+++ = full signed displacement (16, 32 bits)
x = modified
- = unchanged
u = undefined



CPU Instruction Set Summary

Table 6-20. 6x86 CPU Instruction Set Clock Count Summary (Continued)

INSTRUCTION	OPCODE	FLAGS	REAL MODE CLOCK COUNT		PROTECTED MODE CLOCK COUNT		NOTES	
			Reg/Cache Hit	Cache Hit	Reg/Cache Hit	Cache Hit	Real Mode	Protected Mode
JNO <i>Jump on Not Overflow</i> 8-bit Displacement Full Displacement	71 +	- - - - - - - -	1	1			r	
	0F 81 +++	- - - - - - - -	1	1				
JNP/JPO <i>Jump on Not Parity/Parity Odd</i> 8-bit Displacement Full Displacement	7B +	- - - - - - - -	1	1			r	
	0F 8B +++	- - - - - - - -	1	1				
JNS <i>Jump on Not Sign</i> 8-bit Displacement Full Displacement	79 +	- - - - - - - -	1	1			r	
	0F 89 +++	- - - - - - - -	1	1				
JO <i>Jump on Overflow</i> 8-bit Displacement Full Displacement	70 +	- - - - - - - -	1	1			r	
	0F 80 +++	- - - - - - - -	1	1				
JP/JPE <i>Jump on Parity/Parity Even</i> 8-bit Displacement Full Displacement	7A +	- - - - - - - -	1	1			r	
	0F 8A +++	- - - - - - - -	1	1				
JS <i>Jump on Sign</i> 8-bit Displacement Full Displacement	78 +	- - - - - - - -	1	1			r	
	0F 88 +++	- - - - - - - -	1	1				
LAHF <i>Load AH with Flags</i>	9F	- - - - - - - -	2	2		2		
LAR <i>Load Access Rights</i> From Register/Memory	0F 02 [mod reg r/m]	- - - - - x - - - -				8	a	g,h,j,p
	C5 [mod reg r/m]	- - - - - - - -	2	2		4	b	h,i,j
LEA <i>Load Effective Address</i> No Index Register With Index Register	8D [mod reg r/m]	- - - - - - - -	1	1		1		
	C9	- - - - - - - -	1	1		1		
LEAVE <i>Leave Current Stack Frame</i>	C9	- - - - - - - -	4	4		4	b	h
LES <i>Load Pointer to ES</i>	C4 [mod reg r/m]	- - - - - - - -	2	2		4	b	h,i,j
LFS <i>Load Pointer to FS</i>	0F B4 [mod reg r/m]	- - - - - - - -	2	2		4	b	h,i,j
LGD <i>Load GDT Register</i>	0F 01 [mod 010 r/m]	- - - - - - - -	8	8		8	b,c	h,l
LGS <i>Load Pointer to GS</i>	0F B5 [mod reg r/m]	- - - - - - - -	2	2		4	b	h,i,j
LIDT <i>Load IDT Register</i>	0F 01 [mod 011 r/m]	- - - - - - - -	8	8		8	b,c	h,l
LLDT <i>Load LDT Register</i> From Register/Memory	0F 00 [mod 010 r/m]	- - - - - - - -	5	5		5	a	g,h,j,l
LMSW <i>Load Machine Status Word</i> From Register/Memory	0F 01 [mod 110 r/m]	- - - - - - - -	13	13		13	b,c	h,l
LODS <i>Load String</i>	A [110 w]	- - - - - - - -	3	3		3	b	h
LOOP <i>Offset Loop/No Loop</i>	E2 +	- - - - - - - -	1	1		1		r

= immediate 8-bit data
= immediate 16-bit data
= full immediate 32-bit data (8, 16, 32 bits)
+ = 8-bit signed displacement
+++ = full signed displacement (16, 32 bits)
x = modified
- = unchanged
u = undefined

Table 6-20. 6x86 CPU Instruction Set Clock Count Summary (Continued)

INSTRUCTION	OPCODE	FLAGS		REAL MODE CLOCK COUNT		PROTECTED MODE CLOCK COUNT		NOTES							
		OF	DF	IF	TF	SF	ZF	AF	PF	CF	Reg/Cache Hit	Cache Hit	Real Mode	Protected Mode	
LOOPNZ/LOOPNE <i>Offset</i>	E0 +	-	-	-	-	-	-	-	-	-	1	1		r	
LOOPZ/LOOPE <i>Offset</i>	E1 +	-	-	-	-	-	-	-	-	-	1	1		r	
LSS <i>Load Segment Limit</i> From Register/Memory	0F 03 [mod reg r/m]	-	-	-	-	x	-	-	-	-		8		a	g,h,j,p
LSS <i>Load Pointer to SS</i>	0F B2 [mod reg r/m]	-	-	-	-	-	-	-	-	-	2	4		a	h,i,j
LTR <i>Load Task Register</i> From Register/Memory	0F 00 [mod 011 r/m]	-	-	-	-	-	-	-	-	-		7		a	g,h,j,l
MOV <i>Move Data</i> Register to Register Register to Memory Register/Memory to Register Immediate to Register/Memory Immediate to Register (short form) Memory to Accumulator (short form) Accumulator to Memory (short form) Register/Memory to Segment Register Segment Register to Register/Memory	8 [10dw] [11 reg r/m] 8 [100w] [mod reg r/m] 8 [101w] [mod reg r/m] C [011w] [mod 000 r/m] ### B [w reg] ### A [000w] +++ A [001w] +++ 8E [mod sreg3 r/m] 8C [mod sreg3 r/m]	-	-	-	-	-	-	-	-	-	1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1/3 1		b	h,i,j
MOV <i>Move to/from Control/Debug/Test Regs</i> Register to CR0/CR2/CR3 CR0/CR2/CR3 to Register Register to DR0-DR3 DR0-DR3 to Register Register to DR6-DR7 DR6-DR7 to Register Register to TR3-5 TR3-5 to Register Register to TR6-TR7 TR6-TR7 to Register	0F 22 [11 eee reg] 0F 20 [11 eee reg] 0F 23 [11 eee reg] 0F 21 [11 eee reg] 0F 23 [11 eee reg] 0F 21 [11 eee reg] 0F 26 [11 eee reg] 0F 24 [11 eee reg] 0F 24 [11 eee reg]	-	-	-	-	-	-	-	-	-	20/5/5 6 16 14 16 16 14 10 5 10 6	20/5/5 6 16 14 16 16 14 10 5 10 6			1
MOVS <i>Move String</i>	A [010w]	-	-	-	-	-	-	-	-	-	4	4		b	h
MOVSB <i>Move with Sign Extension</i> Register from Register/Memory	0F B[111w] [mod reg r/m]	-	-	-	-	-	-	-	-	-	1	1		b	h
MOVZB <i>Move with Zero Extension</i> Register from Register/Memory	0F B[011w] [mod reg r/m]	-	-	-	-	-	-	-	-	-	1	1		b	h

= immediate 8-bit data
= immediate 16-bit data
= full immediate 32-bit data (8, 16, 32 bits)
+ = 8-bit signed displacement
+++ = full signed displacement (16, 32 bits)
x = modified
- = unchanged
u = undefined



CPU Instruction Set Summary

Table 6-20. 6x86 CPU Instruction Set Clock Count Summary (Continued)

INSTRUCTION	OPCODE	FLAGS		REAL MODE CLOCK COUNT		PROTECTED MODE CLOCK COUNT		NOTES	
		OF DF IF TF SF ZF AF PF CF	Other	Reg/Cache Hit	Other	Reg/Cache Hit	Other	Real Mode	Protected Mode
MUL Unsigned Multiply Accumulator with Register/Memory Multiplier: Byte Word Doubleword	F [011w] [mod 100 r/m]	x - - - x x x u u x		4 4 10		4 4 10		b	h
NEG Negate Integer	F [011w] [mod 011 r/m]	x - - - x x x x x		1		1		b	h
NOP No Operation	90	- - - - - - - - - -		1		1			
NOT Boolean Complement	F [011w] [mod 010 r/m]	- - - - - - - - - -		1		1		b	h
OIO Official Invalid OpCode	0F FF	- - x 0 - - - - - -		1		8 - 125			
OR Boolean OR Register to Register Register to Memory Memory to Register Immediate to Register/Memory Immediate to Accumulator	0 [10dw] [11 reg r/m] 0 [100w] [mod reg r/m] 0 [101w] [mod reg r/m] 8 [00sw] [mod 001 r/m] ### 0 [110w] ###	0 - - - x x u x 0		1 1 1 1 1		1 1 1 1 1		b	h
OUT Output to Port Fixed Port Variable Port	E [011w] # E [111w]	- - - - - - - - - -		14 14		14/28 14/28			m
OUTS Output String	6 [111w]	- - - - - - - - - -		14		14/28		b	h,m
POP Pop Value off Stack Register/Memory Register (short form) Segment Register (ES, SS, DS) Segment Register (FS, GS)	8F [mod 000 r/m] 5 [1 reg] [000 sreg2 111] 0F [10 sreg3 001]	- - - - - - - - - -		1 1 1 1		1 1 3 3		b	h,i,j
POPA Pop All General Registers	61	- - - - - - - - - -		6		6		b	h
POPF Pop Stack into FLAGS	9D	x x x x x x x x x x		9		9		b	h,n
PREFIX BYTES Assert Hardware LOCK Prefix Address Size Prefix Operand Size Prefix Segment Override Prefix CS DS ES FS GS SS	F0 67 66 2E 3E 26 64 65 36	- - - - - - - - - -							m

= immediate 8-bit data
= immediate 16-bit data
= full immediate 32-bit data (8, 16, 32 bits)
+ = 8-bit signed displacement
+++ = full signed displacement (16, 32 bits)
x = modified
- = unchanged
u = undefined

Table 6-20. 6x86 CPU Instruction Set Clock Count Summary (Continued)

INSTRUCTION	OPCODE	FLAGS				REAL MODE CLOCK COUNT		PROTECTED MODE CLOCK COUNT		NOTES				
		OF	DF	IF	TF	SF	ZF	AF	PF	CF	Reg/Cache Hit	Cache Hit	Real Mode	Protected Mode
PUSH <i>Push Value onto Stack</i> Register/Memory Register (short form) Segment Register (ES, CS, SS, DS) Segment Register (FS, GS) Immediate	FF [mod 110 r/m] 5 [0 reg] [000 sreg2 110] 0F [10 sreg3 000] 6 [10s0]###	-	-	-	-	-	-	-	-	-	1 1 1 1 1		b	h
PUSHA <i>Push All General Registers</i>	60	-	-	-	-	-	-	-	-	6	6	b	h	
PUSHF <i>Push FLAGS Register</i>	9C	-	-	-	-	-	-	-	-	2	2	b	h	
RCL <i>Rotate Through Carry Left</i> Register/Memory by 1 Register/Memory by CL Register/Memory by Immediate	D [000w] [mod 010 r/m] D [001w] [mod 010 r/m] C [000w] [mod 010 r/m] #	x	-	-	-	-	-	-	x	3 8 8		b	h	
RCR <i>Rotate Through Carry Right</i> Register/Memory by 1 Register/Memory by CL Register/Memory by Immediate	D [000w] [mod 011 r/m] D [001w] [mod 011 r/m] C [000w] [mod 011 r/m] #	x	-	-	-	-	-	-	x	4 9 9		b	h	
REP INS <i>Input String</i>	F3 6[110w]	-	-	-	-	-	-	-	-	12+5n 28+5n		b	h,m	
REP LODS <i>Load String</i>	F3 A[110w]	-	-	-	-	-	-	-	-	10+n		b	h	
REP MOVS <i>Move String</i>	F3 A[010w]	-	-	-	-	-	-	-	-	9+n		b	h	
REP OUTS <i>Output String</i>	F3 6[111w]	-	-	-	-	-	-	-	-	12+5n 28+5n		b	h,m	
REP STOS <i>Store String</i>	F3 A[101w]	-	-	-	-	-	-	-	-	10+n		b	h	
REPE CMPS <i>Compare String</i> (Find non-match)	F3 A[011w]	x	-	-	-	x	x	x	x	10+2n		b	h	
REPE SCAS <i>Scan String</i> (Find non-AL/AX/EAX)	F3 A[111w]	x	-	-	-	x	x	x	x	10+2n		b	h	
REPNE CMPS <i>Compare String</i> (Find match)	F2 A[011w]	x	-	-	-	x	x	x	x	10+2n		b	h	
REPNE SCAS <i>Scan String</i> (Find AL/AX/EAX)	F2 A[111w]	x	-	-	-	x	x	x	x	10+2n		b	h	
RET <i>Return from Subroutine</i> Within Segment Within Segment Adding Immediate to SP Intersegment Intersegment Adding Immediate to SP Protected Mode: Different Privilege Level Intersegment Intersegment Adding Immediate to SP	C3 C2 ## CB CA ##	-	-	-	-	-	-	-	-	3 4 4 4	3 4 7 7 23 23	b	g,h,j,k,r	

+ = 8-bit signed displacement
 +++ = full signed displacement (16, 32 bits)
 x = modified
 - = unchanged
 u = undefined



CPU Instruction Set Summary

Table 6-20. 6x86 CPU Instruction Set Clock Count Summary (Continued)

INSTRUCTION	OPCODE	FLAGS								REAL MODE CLOCK COUNT		PROTECTED MODE CLOCK COUNT		NOTES	
		OF	DF	IF	TF	SF	ZF	AF	PF	CF	Reg/Cache Hit	Cache Hit	Reg/Cache Hit	Protected Mode	Real Mode
ROL Rotate Left Register/Memory by 1 Register/Memory by CL Register/Memory by Immediate	D[000w] [mod 000 r/m]	x	-	-	-	-	-	-	-	x	1	1		b	h
	D[001w] [mod 000 r/m]	u	-	-	-	-	-	-	-	x	2	2			
	C[000w] [mod 000 r/m] #	u	-	-	-	-	-	-	-	x	1	1			
ROR Rotate Right Register/Memory by 1 Register/Memory by CL Register/Memory by Immediate	D[000w] [mod 001 r/m]	x	-	-	-	-	-	-	-	x	1	1		b	h
	D[001w] [mod 001 r/m]	u	-	-	-	-	-	-	-	x	2	2			
	C[000w] [mod 001 r/m] #	u	-	-	-	-	-	-	-	x	1	1			
RSDC Restore Segment Register and Descriptor	0F 79 [mod sreg3 r/m]	-	-	-	-	-	-	-	-	-	6	6		s	s
RSLDT Restore LDTR and Descriptor	0F 7B [mod 000 r/m]	-	-	-	-	-	-	-	-	-	6	6		s	s
RSM Resume from SMM Mode	0F AA	x	x	x	x	x	x	x	x	x	40	40		s	s
RSTS Restore TSR and Descriptor	0F 7D [mod 000 r/m]	-	-	-	-	-	-	-	-	-	6	6		s	s
SAHF Store AH in FLAGS	9E	-	-	-	-	x	x	x	x	x	1	1			
SAL Shift Left Arithmetic Register/Memory by 1 Register/Memory by CL Register/Memory by Immediate	D[000w] [mod 100 r/m]	x	-	-	-	x	x	u	x	x	1	1		b	h
	D[001w] [mod 100 r/m]	u	-	-	-	x	x	u	x	x	2	2			
	C[000w] [mod 100 r/m] #	u	-	-	-	x	x	u	x	x	1	1			
SAR Shift Right Arithmetic Register/Memory by 1 Register/Memory by CL Register/Memory by Immediate	D[000w] [mod 111 r/m]	x	-	-	-	x	x	u	x	x	1	1		b	h
	D[001w] [mod 111 r/m]	u	-	-	-	x	x	u	x	x	2	2			
	C[000w] [mod 111 r/m] #	u	-	-	-	x	x	u	x	x	1	1			
SBB Integer Subtract with Borrow Register to Register Register to Memory Memory to Register Immediate to Register/Memory Immediate to Accumulator (short form)	I[10dw] [11 reg r/m]	x	-	-	-	x	x	x	x	x	1	1		b	h
	I[100w] [mod reg r/m]	1	-	-	-	x	x	u	x	x	1	1			
	I[101w] [mod reg r/m]	1	-	-	-	x	x	u	x	x	1	1			
	8[00sw] [mod 011 r/m] ###	1	-	-	-	x	x	u	x	x	1	1			
	I[110w] ###	1	-	-	-	x	x	u	x	x	1	1			
SCAS Scan String	A [111w]	x	-	-	-	x	x	x	x	x	2	2		b	h
SETB/SETNAE/SETC Set Byte on Below/Not Above or Equal/Carry To Register/Memory	0F 92 [mod 000 r/m]	-	-	-	-	-	-	-	-	-	1	1			h
SETBE/SETNA Set Byte on Below or Equal/Not Above To Register/Memory	0F 96 [mod 000 r/m]	-	-	-	-	-	-	-	-	-	1	1			h
SETE/SETZ Set Byte on Equal/Zero To Register/Memory	0F 94 [mod 000 r/m]	-	-	-	-	-	-	-	-	-	1	1			h
SETL/SETNGE Set Byte on Less/Not Greater or Equal To Register/Memory	0F 9C [mod 000 r/m]	-	-	-	-	-	-	-	-	-	1	1			h

x = modified
- = unchanged
u = undefined

+ = 8-bit signed displacement
+++ = full signed displacement (16, 32 bits)

= immediate 8-bit data
= immediate 16-bit data
= full immediate 32-bit data (8, 16, 32 bits)

Table 6-20. 6x86 CPU Instruction Set Clock Count Summary (Continued)

INSTRUCTION	OPCODE	FLAGS				REAL MODE CLOCK COUNT	PROTECTED MODE CLOCK COUNT	NOTES								
		OF	DF	IF	TF			SF	ZF	AF	PF	CF	Reg/Cache Hit	Real Mode	Protected Mode	
SETLE/SETNG Set Byte on Less or Equal/Not Greater To Register/Memory	0F 9E [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	h
SETNB/SETAE/SETNC Set Byte on Not Below/Above or Equal/Not Carry To Register/Memory	0F 93 [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	h
SETNBE/SETA Set Byte on Not Below or Equal/Above To Register/Memory	0F 97 [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	h
SETNE/SETNZ Set Byte on Not Equal/Not Zero To Register/Memory	0F 95 [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	h
SETNL/SETGE Set Byte on Not Less/Greater or Equal To Register/Memory	0F 9D [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	h
SETNLE/SETG Set Byte on Not Less or Equal/Greater To Register/Memory	0F 9F [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	h
SETNO Set Byte on Not Overflow To Register/Memory	0F 91 [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	h
SETNP/SETPO Set Byte on Not Parity/Parity Odd To Register/Memory	0F 9B [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	h
SETNS Set Byte on Not Sign To Register/Memory	0F 99 [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	h
SETO Set Byte on Overflow To Register/Memory	0F 90 [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	h
SETP/SETPE Set Byte on Parity/Parity Even To Register/Memory	0F 9A [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	h
SETS Set Byte on Sign To Register/Memory	0F 98 [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	h
SGDT Store GDT Register To Register/Memory	0F 01 [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	h
SHL Shift Left Logical Register/Memory by 1 Register/Memory by CL Register/Memory by Immediate	D [000w] [mod 100 r/m]	x	-	-	-	x	x	u	x	x	-	-	-	-	-	h
	D [001w] [mod 100 r/m]	u	-	-	-	x	x	u	x	x	-	-	-	-	-	h
	C [000w] [mod 100 r/m] #	u	-	-	-	x	x	u	x	x	-	-	-	-	-	h
SHLD Shift Left Double Register/Memory by Immediate Register/Memory by CL	0F A4 [mod reg r/m] #	u	-	-	-	x	x	u	x	x	-	-	-	-	-	h
	0F A5 [mod reg r/m]	u	-	-	-	x	x	u	x	x	-	-	-	-	-	h

x = modified
- = unchanged
u = undefined

+ = 8-bit signed displacement
+++ = full signed displacement (16, 32 bits)

= immediate 8-bit data
= immediate 16-bit data
= full immediate 32-bit data (8, 16, 32 bits)



CPU Instruction Set Summary

Table 6-20. 6x86 CPU Instruction Set Clock Count Summary (Continued)

INSTRUCTION	OPCODE	FLAGS								REAL MODE CLOCK COUNT		PROTECTED MODE CLOCK COUNT		NOTES	
		OF	DF	IF	TF	SF	ZF	AF	PF	CF	Reg/Cache Hit	Cache Hit	Real Mode	Protected Mode	
SHR Shift Right Logical Register/Memory by 1 Register/Memory by CL Register/Memory by Immediate	D [000w] [mod 101 r/m]	x	-	-	-	x	x	x	u	x	x	1	1	b	h
	D [001w] [mod 101 r/m]	u	-	-	-	x	x	u	u	x	x	2	2		
	C [000w] [mod 101 r/m] #	u	-	-	-	x	x	u	u	x	x	1	1		
SHRD Shift Right Double Register/Memory by Immediate Register/Memory by CL	0F AC [mod reg r/m] #	u	-	-	-	x	x	u	u	x	x	4	4	b	h
	0F AD [mod reg r/m]											5	5		
SIDT Store IDT Register To Register/Memory	0F 01 [mod 001 r/m]	-	-	-	-	-	-	-	-	-	-	4	4	b,c	h
SLDT Store LDT Register To Register/Memory	0F 00 [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	1	1	a	h
SMINT Software SMM Entry	0F 7E	-	-	-	-	-	-	-	-	-	-	55	55	s	s
SMSW Store Machine Status Word	0F 01 [mod 100 r/m]	-	-	-	-	-	-	-	-	-	-	6	6	b,c	h
STC Set Carry Flag	F9	-	-	-	-	-	-	-	-	-	-	1	1		
STD Set Direction Flag	FD	-	1	-	-	-	-	-	-	-	-	7	7		
STI Set Interrupt Flag	FB	-	-	1	-	-	-	-	-	-	-	7	7		m
STOS Store String To Register/Memory	A [101w]	-	-	-	-	-	-	-	-	-	-	2	2	b	h
STR Store Task Register To Register/Memory	0F 00 [mod 001 r/m]	-	-	-	-	-	-	-	-	-	-	4	4	a	h
SUB Integer Subtract Register to Register Register to Memory Memory to Register Immediate to Register/Memory (short form)	2 [10dw] [11 reg r/m]	x	-	-	-	x	x	x	x	x	x	1	1	b	h
	2 [100w] [mod reg r/m]											1	1		
	2 [101w] [mod reg r/m]											1	1		
	8 [00sw] [mod 101 r/m] ###											1	1		
	2 [110w] ###											1	1		
SVDC Save Segment Register and Descriptor	0F 78 [mod seg3 r/m]	-	-	-	-	-	-	-	-	-	-	12	12	s	s
	0F 7A [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	12	12	s	s
SVLDT Save LDTR and Descriptor															
	0F 7C [mod 000 r/m]	-	-	-	-	-	-	-	-	-	-	14	14	s	s
SVTS Save TSR and Descriptor															
TEST Test Bits Register/Memory and Register Immediate Data and Register/Memory Immediate Data and Accumulator	8 [010w] [mod reg r/m]	0	-	-	-	x	x	u	u	x	0	1	1	b	h
	F [011w] [mod 000 r/m] ###											1	1		
	A [100w] ###											1	1		
VERR Verify Read Access To Register/Memory	0F 00 [mod 100 r/m]	-	-	-	-	-	-	x	-	-	-	7	7	a	g,h,j,p
VERW Verify Write Access To Register/Memory	0F 00 [mod 101 r/m]	-	-	-	-	-	-	x	-	-	-	7	7	a	g,h,j,p
WAIT Wait Until FPU Not Busy	9B	-	-	-	-	-	-	-	-	-	-	5	5		

x = modified
- = unchanged
u = undefined

+ = 8-bit signed displacement
+++ = full signed displacement (16, 32 bits)

= immediate 8-bit data
= immediate 16-bit data
= full immediate 32-bit data (8, 16, 32 bits)

Table 6-20. 6x86 CPU Instruction Set Clock Count Summary (Continued)

INSTRUCTION	OPCODE	FLAGS		REAL MODE CLOCK COUNT	PROTECTED MODE CLOCK COUNT	NOTES	
		OF DF IF TF SF ZF AF PF CF	- - - - - - - -	Reg/Cache Hit	Reg/Cache Hit	Real Mode	Protected Mode
WBINVD Write-Back and Invalidate Cache	0F 09	- - - - - - - -	- - - - - - - -	15	15	t	t
XADD Exchange and Add Register1, Register2 Memory, Register	0F C[000w] [11 reg2 reg1] 0F C[000w] [mod reg r/m]	x - - - - x x x x x	- - - - - - - -	2 2	2 2		
XCHG Exchange Register/Memory with Register Register with Accumulator	8[011w] [mod reg r/m] 9[0 reg]	- - - - - - - -	- - - - - - - -	2 2	2 2	b,f	f,h
XLAT Translate Byte	D7	- - - - - - - -	- - - - - - - -	4	4		h
XOR Boolean Exclusive OR Register to Register Register to Memory Memory to Register Immediate to Register/Memory Immediate to Accumulator (short form)	3 [00dw] [11 reg r/m] 3 [000w] [mod reg r/m] 3 [001w] [mod reg r/m] 8 [00sw] [mod 110 r/m] ### 3 [010w] ###	0 - - - - x x u x 0	- - - - - - - -	1 1 1 1 1	1 1 1 1 1	b	h

= immediate 8-bit data
= immediate 16-bit data
= full immediate 32-bit data (8, 16, 32 bits)
+ = 8-bit signed displacement
+++ = full signed displacement (16, 32 bits)
x = modified
- = unchanged
u = undefined

Instruction Notes for Instruction Set Summary

Notes a through c apply to Real Address Mode only:

- a. This is a Protected Mode instruction. Attempted execution in Real Mode will result in exception 6 (invalid op-code).
- b. Exception 13 fault (general protection) will occur in Real Mode if an operand reference is made that partially or fully extends beyond the maximum CS, DS, ES, FS, or GS segment limit (FFFFH). Exception 12 fault (stack segment limit violation or not present) will occur in Real Mode if an operand reference is made that partially or fully extends beyond the maximum SS limit.
- c. This instruction may be executed in Real Mode. In Real Mode, its purpose is primarily to initialize the CPU for Protected Mode.

Notes e through g apply to Real Address Mode and Protected Virtual Address Mode:

- e. An exception may occur, depending on the value of the operand.
- f. LOCK# is automatically asserted, regardless of the presence or absence of the LOCK prefix.
- g. LOCK# is asserted during descriptor table accesses.

Notes h through r apply to Protected Virtual Address Mode only:

- h. Exception 13 fault will occur if the memory operand in CS, DS, ES, FS, or GS cannot be used due to either a segment limit violation or an access rights violation. If a stack limit is violated, an exception 12 occurs.
- i. For segment load operations, the CPL, RPL, and DPL must agree with the privilege rules to avoid an exception 13 fault. The segment's descriptor must indicate "present" or exception 11 (CS, DS, ES, FS, GS not present). If the SS register is loaded and a stack segment not present is detected, an exception 12 occurs.



Continued on next page...

- j. All segment descriptor accesses in the GDT or LDT made by this instruction will automatically assert LOCK# to maintain descriptor integrity in multiprocessor systems.
- k. JMP, CALL, INT, RET, and IRET instructions referring to another code segment will cause an exception 13, if an applicable privilege rule is violated.
- l. An exception 13 fault occurs if CPL is greater than 0 (0 is the most privileged level).
- m. An exception 13 fault occurs if CPL is greater than IOPL.
- n. The IF bit of the flag register is not updated if CPL is greater than IOPL. The IOPL and VM fields of the flag register are updated only if CPL = 0.
- o. The PE bit of the MSW (CR0) cannot be reset by this instruction. Use MOV into CR0 if desiring to reset the PE bit.
- p. Any violation of privilege rules as apply to the selector operand does not cause a Protection exception, rather, the zero flag is cleared.
- q. If the coprocessor's memory operand violates a segment limit or segment access rights, an exception 13 fault will occur before the ESC instruction is executed. An exception 12 fault will occur if the stack limit is violated by the operand's starting address.
- r. The destination of a JMP, CALL, INT, RET, or IRET must be in the defined limit of a code segment or an exception 13 fault will occur.

Note s applies to Cyrix specific SMM instructions:

- s. All memory accesses to SMM space are non-cacheable. An invalid opcode exception 6 occurs unless SMI is enabled and ARR3 size > 0, and CPL = 0 and ISMAC is set or if in an SMI handler].

Note t applies to cache invalidation instructions with the cache operating in write-back mode:

- t. The total clock count is the clock count shown plus the number of clocks required to write all "modified" cache lines to external memory.