Systematic encoding of x86-32/64 instruction set 24-bit instructions plus immediates 4-bit register fields, 8-bit op-code James C. Brakefield 2020

M2R memory-operand op register => register op-code, three reg fields, operand size, displacement size

R2M memory-operand op register => memory op-code, three reg fields, operand size, displacement size

R2R register op register => register op-code, three reg fields (two source, one destination)

I2R immediate op register => register op-code, reg field, immediate

I2M memory-operand op immediate => memory op-code, two reg fields, opnd size, displacmt size, imm size

Op Code	M2R	R2R	R2M	I2R	I2M	x86	186	286	386	486	Pent	A64	Description	Comment	CCR chng	# src opnd	dst = src2	# byte	dbl reg
General Purpose Integer															CZSOAP				
min totals	13	10	8	12	10								17						
CALL	Y	Y		Y		Y	Y	Y	Y	Y	Y	Y	Near Procedure Call	Pushes return adr to stack	------	1			
JMP	Y			Y		Y	Y	Y	Y	Y	Y	Y	Near Jump		------	1			
Jcc						Y	Y	Y	Y	Y	Y	Y	Jump on Condition	24 & 48 bit versions?	------	1			
RET						Y	Y	Y	Y	Y	Y	Y	Near Return from Called Procedure		------			1,3	
ADC	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Add with Carry		CZSOAP	2	Y		
ADD	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Signed or Unsigned Add		CZSOAP	2	Y		
SBB	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Subtract with Borrow		CZSOAP	2	Y		
SUB	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Signed or Unsigned Subtract		CZSOAP	2	Y		
CMP	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Compare		CZSOAP	2			
AND	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Logical AND		CZSOAP	2	Y		
ANDN	Y	Y										Y	Logical And-Not		0ZS0UP	2	N		
OR	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Logical OR		0ZS0UP	2	Y		
XOR	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Logical Exclusive OR		0ZS0UP	2	Y		
TEST	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Test Bits		0ZS0UP	2			
MOV	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Move	also MOV for control & debug regs	------	1			
IN						Y	Y	Y	Y	Y	Y	Y	Input from Port			1		1-2	
OUT						Y	Y	Y	Y	Y	Y	Y	Output to Port			1		1-2	
BOUND	Y						Y	Y	Y	Y	Y		Check Array Bound	2nd operand is pair of integers (double size)		2			
BSWAP		Y								Y	Y	Y	Byte Swap			1	Y		
CBW						Y	Y	Y	Y	Y	Y	Y	Convert to Sign-Extended			1		1	
CDQE									Y	Y	Y	Y	Convert to Sign-Extended			1		1	
CDQE									Y	Y	Y	Y	Convert to Sign-Extended			1		1	
CLC						Y	Y	Y	Y	Y	Y	Y	Clear Carry Flag		C			1	
CLFLUSH	Y					Y	Y	Y	Y	Y	Y	Y	Cache Line Flush			1			
CLI						Y	Y	Y	Y	Y			Clear Interrrupt Flag					1	
CMC						Y	Y	Y	Y	Y	Y	Y	Complement Carry Flag		C			1	
CMOVcc	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Conditional Move	equals two inst		1			
CMPXCHG	Y	Y								Y	Y	Y	Compare and Exchange	two writes	Z	2	Y		
CMPXCHG16B											Y	Y	Compare and Exchange Sixteen Bytes	two writes	Z	2	Y		Y
CMPXCHG8B											Y	Y	Compare and Exchange Eight Bytes	two writes	Z	2	Y		
CPUID						Y	Y	Y	Y	Y	Y	Y	Processor Identification					2	
CQO						Y	Y	Y	Y	Y	Y	Y	Convert to Sign-Extended			1		1	
CRC32	Y	Y				Y	Y	Y	Y	Y	Y	Y	CRC32 Cyclical Redundancy Check			2	Y		
CWD						Y	Y	Y	Y	Y	Y	Y	Convert to Sign-Extended			1		1	
CWDE									Y	Y			Convert to Sign-Extended	see CDQ		1		1	
DEC				Y	Y	Y	Y	Y	Y	Y	Y	Y	Decrement by 1	redundant		1	Y		
DIV	Y	Y				Y	Y	Y	Y	Y	Y	Y	Unsigned Divide		Y	2	Y		Y
ENTER							Y	Y	Y	Y	Y	Y	Create Procedure Stack Frame	equals three inst				4	
HLT						Y	Y	Y	Y	Y			Halt					1	
IDIV	Y	Y				Y	Y	Y	Y	Y	Y	Y	Signed Divide		Y	2	Y		Y
IMUL	Y	Y				Y	Y	Y	Y	Y	Y	Y	Signed Multiply		Y	2	Y		Y
INC				Y	Y	Y	Y	Y	Y	Y	Y	Y	Increment by 1	redundant		1	Y		
INT						Y	Y	Y	Y	Y	Y	Y	Interrupt to Vector	8-bit vector #				2	
INTO						Y	Y	Y	Y	Y	Y	Y	Interrupt to Overflow Vector					1	
INVD										Y			Invalidate Cache					2	
INVLPG										Y			Invalidate TLB Entry						
IRET						Y	Y	Y	Y	Y			Interrupt Return					1	
JCXZ				Y		Y	Y	Y	Y	Y	Y	Y	Jump if rCX Zero	relative branch		1			
JECXZ				Y		Y	Y	Y	Y	Y	Y	Y	Jump if rCX Zero	relative branch		1			
JRCXZ				Y		Y	Y	Y	Y	Y	Y	Y	Jump if rCX Zero	relative branch		1			
LAHF						Y	Y	Y	Y	Y	Y	Y	Load Status Flags into AH Register			1			
LEA	Y					Y	Y	Y	Y	Y	Y	Y	Load Effective Address			1			
LEAVE							Y	Y	Y	Y	Y	Y	Delete Procedure Stack Frame	equals two inst				1	
LOCK						Y	Y	Y	Y	Y	Y	Y	Assert Hardware Lock					1	

LOOP				Y		Y	Y	Y	Y	Y			Loop	relative branch				2	
LOOPE, LOOPZ				Y		Y	Y	Y	Y	Y	Y	Y	Loop	relative branch				2	
LOOPNE, LOOPNZ				Y		Y	Y	Y	Y	Y	Y	Y	Loop	relative branch				2	
MOVBE	Y		Y									Y	Move Big Endian						
MOVSX	Y								Y	Y	Y	Y	Move with Sign-Extension						
MOVSXD	Y										Y	Y	Move with Sign-Extend Doubleword						
MOVZX	Y								Y	Y	Y	Y	Move with Zero-Extension						
MUL	Y	Y				Y	Y	Y	Y	Y	Y	Y	Unsigned Multiply		Y	2	Y		Y
NEG	Y	Y				Y	Y	Y	Y	Y	Y	Y	Two's Complement Negation		Y	1			
NOP	Y					Y	Y	Y	Y	Y	Y	Y	No Operation						
NOT	Y	Y				Y	Y	Y	Y	Y	Y	Y	One's Complement Negation		Y	1			
PAUSE						Y	Y	Y	Y	Y	Y	Y	Pause	architecturally a NOP				2	
POP	Y	Y				Y	Y	Y	Y	Y	Y	Y	Pop Stack			0	1		
POPA						Y	Y	Y	Y	Y	Y		POP All GPRs					1	
POPAD						Y	Y	Y	Y	Y	Y		POP All GPRs					1	
POPF						Y	Y	Y	Y	Y	Y	Y	POP to rFLAGS		Y			1	
POPFD						Y	Y	Y	Y	Y	Y	Y	POP to rFLAGS		Y			1	
POPFQ						Y	Y	Y	Y	Y	Y	Y	POP to rFLAGS		Y			1	
PUSH	Y	Y		Y		Y	Y	Y	Y	Y	Y	Y	Push to Stack			1			
PUSHA		Y				Y	Y	Y	Y	Y	Y		Push to Stack			1			
PUSHAD		Y				Y	Y	Y	Y	Y	Y		Push to Stack			1			
RCL		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Rotate Through Carry Left	variable amount shift	Y	2	Y		
RCR		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Rotate Through Carry Right	variable amount shift	Y	2	Y		
ROL		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Rotate Left	variable amount shift	Y	2	Y		
ROR		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Rotate Right	variable amount shift	Y	2	Y		
SAHF						Y	Y	Y	Y	Y	Y	Y	Store AH into Flags		Y			1	
SAL, SHL		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Shift Left	variable amount shift	Y	2	Y		
SAR		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Shift Arithmetic Right	variable amount shift	Y	2	Y		
SETcc		Y	Y						Y	Y	Y	Y	Set Byte on Condition						
SHLD		Y	Y	Y	Y				Y	Y	Y	Y	Shift Left Double	variable amount shift	Y	3	Y		
SHR		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Shift Right	variable amount shift	Y	2	Y		
SHRD		Y	Y	Y	Y				Y	Y	Y	Y	Shift Right Double	variable amount shift	Y	3	Y		
SMSW								Y	Y	Y			Store Machine Status Word						
STC						Y	Y	Y	Y	Y	Y	Y	Set Carry Flag		C			1	
STR								Y	Y	Y			Store Task Register						
WAIT						Y	Y	Y	Y	Y			Wait Until Not Busy					2	
WBINVD										Y			Write-Back and Invalidate Cache					2	
XADD		Y	Y							Y	Y	Y	Exchange and Add	two writes	Y	2	Y		
XCHG		Y	Y			Y	Y	Y	Y	Y	Y	Y	Exchange	two writes		2	Y		
XLAT						Y	Y	Y	Y	Y	Y	Y	Translate Table Index	byte only				1	
XLATB						Y	Y	Y	Y	Y	Y	Y	Translate Table Index	byte only				1	
totals	32	36	22	31	22														