**Original 8068/8088 instructions**

AAA = ASCII mengatur AL sesudah penambahan

AAD = ASCII mengatur AX sebelum pembagian

AAM = ASCII mengatur AX sesudah perkalian

AAS = ASCII mengatur AL sesudah pengurangan

ADC = menambahkan dengan muatan

ADD = menambahkan

AND = logic AND

CALL = memanggil prosedur

CBW = mengubah byte ke word

CLC = mengosongkan carry flag

CLD = mengosongkan direction flag

CLI = mengosongkan interrupt flag

CMC = menambahkan carry flag

CMP = membandingkan operan

CMPSB = membandingkan bytes dalam memori

CMPSW = membandingkan words

CWD = mengubah word menjadi doubleword

DAA = desimal mengatur AL sesudah penambahan

DAS = desimal mengatur AL sesudah pengurangan

DEC = mengurangi satu

DIV = pembagian tidak bertanda

ESC = digunakan dengan unit floating-point

HLT = memasuki keadaan berhenti

IDIV = pembagian bertanda

IMUL = penggandaaan bertanda

IN = input dari port

INC = data processing (menambah satu)

INT = memanggil untuk berhenti

INTO = memanggil untuk berhenti jika berlebihan

IRET = kembali dari penghentian

Jcc = loncat jika dalam suatu kondisi

JCXZ = loncat jika CX nol

JMP = terjadi proses control (loncat ke alamat lain)

LAHF = memuat FLAGS ke dalam register AH

LDS = memuat penanda menggunakan DS

LEA = memuat alamat efektif

LES = memuat ES dengan penanda

LOCK = memaksa sinyal BUS LOCK

LODSB = memuat string byte

LODSW = memuat string word

LOOP/LOOPx = kontrol pengulangan

MOV = memindahkan

MOVSB = memindahkan byte dari string ke string

MOVSW = memindahkan word dari string ke string

MUL = menggandakan tanpa tanda

NEG = dua negasi seimbang

NOP = tidak ada operasi

NOT = meniadakan operand, logika NOT

OR = logika OR

OUT = output ke port

POP = memunculkan data dari tumpukan (stack)

POPF = memunculkan register FLAGS dari tumpukan (stack)

PUSH = memasukkan data ke dalam tumpukan (stack)

PUSHF = memasukkan FLAGS ke dalam tumpukan (stack)

RCL = memutar ke kiri (dengan muatan)

RCR = memutar ke kanan (dengan muatan)

REPxx = mengulangi MOVS/STOS/CMPS/LODS/SCAS

RET = kembali dari prosedur

RETF = kembali dari prosedur yang dekat

RETN = kembali dari prosedur yang jauh

ROL = memutar ke kiri

ROR = memutar ke kanan

SAHF = menyimpan AH ke dalam FLAGS

SAL = mengubah aritmatika di kiri (dengan shift kiri)

SAR = mengubah aritmatika di kanan (dengan shift kanan)

SBB = pengurangan dengan meminjam

SCASB = membandingkan byte string

SCASW = membandingkan word string

SHL = mengubah di kiri (shift kiri tidak bertanda)

SHR = mengubah di kanan (shift kanan tidak bertanda)

STC = tanda mengatur muatan

STD = tanda mengatur arah

STI = tanda mengatur penghentian

STOSB = menyimpan byte ke dalam string

STOSW = menyimpan word ke dalam string

SUB = substraksi atau pengurangan

TEST = membandingkan logika (AND)

WAIT = menunggu hingga tidak sibuk

XCHG = bertukar data

XLAT = daftar pencarian terjemahan

XOR = OR eksklusif

**Added with 80186/80188**

BOUND = memeriksa indeks array terhadap bounds

ENTER = memasuki stack frame

INS = masukan dari port ke string

LEAVE = meninggalkan stack frame

OUTS = keluaran dari string ke port

POPA = memunculkan semua daftar tujuan umum dari stack

PUSHA = mendorong semua daftar tujuan umum ke dalam stack

PUSH immediate = mendorong nilai byte/word langsung ke dalam stack

IMUL immediate = menandai penggandaan dari nilai byte/word langsung

SHL/SHR/SAL/SAR/ROL/ROR/RCL/RCR immediate = memutar/mengganti bits dengan nilai langsung lebih dari satu

**Added with 80286**

ARPL = mengatur pemilih lingkungan RPL

CLTS = membersihkan tanda penggantian tugas dalam register CR0

LAR = memuat hak akses byte

LGDT = memuat daftar deskriptor global

LIDT = memuat daftar deskriptor interupsi

LLDT = memuat daftar deskriptor lokal

LMSW = memuat word status mesin

LOADALL = memuat semua register CPU, termasuk internal seperti GDT

LSL = memuat batasan segmen

LTR = memuat daftar tugas

SGDT = menyimpan daftar deskriptor global

SIDT = menyimpan daftar deskriptor interupsi

SLDT = menyimpan daftar deskriptor lokal

SMSW = menyimpan word status mesin

STR = menyimpan daftar tugas

VERR = memeriksa sebuah segmen untuk di baca

VERW = memeriksa sebuah segmen untuk di tulis

**Added with 80386**

BSF = memindai bit ke depan

BSR = memindai bit berlawanan

BT = tes bit

BTC = tes dan melengkapi bit

BTR = tes dan mengatur ulang bit

BTS = tes dan mengatur bit

CDQ = mengubah double-word ke quad-word

CMPSD = membandingkan string double-word

CWDE = mengubah word menjadi double-word

INSD = masukkan dari port ke string double-word

IRETx = interupsi kembali; akhiran D berarti mengembalikan 32-bit, akhiran F berarti tidak boleh menghasilkan kode akhiran (contoh instruksi LEAVE)

JECXZ = meloncat jika ECX nol

LFS, LGS = memuat penanda jauh

LSS = memuat segmen stack

LODSD = memuat string double-word

LOOPW, LOOP*cc*W = perulangan, perulangan kondisional

LOOPD, LOOP*cc*D = perulangan dengan seimbang

MOV to/from CR/CD/TR = berpindah ke/dari register spesial

MOVSD = memindahkan string double-word

MOVSX = memindahkan dengan ekstensi bertanda

MOVZX = memindahkan dengan ekstensi nol

OUTSD = keluaran ke port dari string double-word

POPAD = memunculkan seluruh register double-word (32-bit) dari stack

POPFD = memunculkan data ke dalam register EFLAGS

PUSHAD = mendorong semua register double-word (32-bit) ke dalam stack

PUSHFD = mendorong register EFLAGS ke dalam stack

SCASD = memindai data string double-word

SETcc = mengatur byte ke satu dalam kondisi tertentu, jika tidak nol

SHLD = mengubah double-word kiri

SHRD = mengubah double-word kanan

STOSD = menyimpan string double-word

**Added with 80486**

BSWAP = menukar byte

CMPXCHG = atomic CoMPare dan eXCHanGe

INVD = menghapuskan caches internal

INVLPG = menghapuskan TLB entry

WBINVD = menulis ulang dan menghapuskan cache

XADD = menukar operan pertama dengan operan kedua, lalu memuat dua nilai operan tujuan

**Added with Pentium**

CPUID = identifikasi CPU

CMPXCHG8B = membandingkan dan menukar EDX:EAX dengan m64

RDMSR = membaca dari MSR (Model-specific register)

RDTSC = membaca dari TSC (Time Stamp Counter)

WRMSR = menulis di Model-Specific Register (MSR)

RSM = melanjutkan dari mode manajemen sistem

**Added with Pentium MMX**

RDPMC = membaca PMC (Performance Monitoring Counter)

**Added with AMD K6**

SYSCALL = fungsi sama dengan SYSENTER

SYSRET = fungsi sama dengan SYSEXIT

**Added with Pentium Pro**

CMOVcc = pemindahan kondisional

UD2 = instruksi tidak terdefinisi

**Added with Pentium II**

SYSENTER = sistem untuk memanggil ENTER

SYSEXIT = sistem untuk memanggil EXIT

**Added with SSE**

MASKMOVQ mm1, mm2 = pemindahan masked dari Quadword

MOVNTPS m128, xmm1 = memindahkan Aligned Four Packed Single-FP Non Temporal

MOVNTQ m64, mm = memindahkan Quadrow menggunakan tanda tidak sementara

NOP r/m16 = instruksi multi-byte tanpa operasi

NOP r/m32 = instruksi multi-byte tanpa operasi

PREFETCHT0 = mentransfer data dari alamat

PREFETCHT1 = mentransfer data dari alamat

PREFETCHT2 = mentransfer data dari alamat

PREFETCHNTA = mentransfer data dari alamat

SFENCE = menyimpan Fence

**Added with SSE2**

CLFUSH m8 = membersihkan baris cache

LFENCE = memuat Fence

MFENCE = memori Fence

MOVNTI m32, r32 = memindahkan Doubleword tidak sementara

PAUSE = memutar petunjuk perulangan

**Added with SSE3**

MONITOR EAX, ECX, EDX = pengaturan alamat monitor

MWAIT EAX, ECX = monitor menunggu

**Added with SSE4.2**

CRC32 r32, r/m8, CRC32 r32, r/m16, CRC32 r32, r/m32, CRC32 r64, r/m8, CRC32 r64, r/m64 = akumulasi CRC32

**Added with x86-64**

CDQE = tanda memperluas EAX ke dalam RAX

CQO = tanda memperluas RAX ke dalam RDX:RAX

CMPSQ = membandingkan string Quadword

CMPXCHG16B = membandingkan dan menukar 16 bytes

IRETQ = 64-bit kembali dari gangguan

JRCXZ = meloncat jika RCX nol

LODSQ = memuat String Quadword

MOVSXD = MOV dengan tanda memperluas 32-bit ke 64-bit

POPFQ = memunculkan register RFLAGS

PUSHFQ = mendorong register RFLAGS

RDTSCP = membaca Time Stamp Counter dan ID prosesor

SCASQ = memindai string Quadword

STOSQ = menyimpan string Quadword

SWAPGS = menukar GS base dengan KernelGSBase MSR

**Added with AMD-V**

CLGI = membersihkan GIF (Global Interrupt Flag)

INVLPGA = menghapuskan masukan TLB dalam ASID khusus

MOV(CRn) = berpindah ke atau dari kontrol register

MOV(DRn) = berpindah ke atau dari debug register

SKINIT = mengamankan Init dan Jump dengan pengesahan

STGI = mengatur GIF (Global Interrupt Flag)

VMLOAD = memuat keadaan dari VMCB

VMMCALL = memanggil VMM

VMRUN = menjalankan mesin virtual

VMSAVE = menyimpan keadaan ke dalam VMCB

**Added with Intel VT-x**

VMPTRLD = memuat petunjuk ke struktur kontrol Virtual-Machine

VMPTRST = menyimpan petunjuk ke struktur kontrol Virtual-Machine

VMCLEAR = membersihkan kontrol struktur Virtual-Machine

VMREAD = membaca bagian dari struktur kontrol Virtual-Machine

VMWRITE = menulis bagian dari struktur kontrol Virtual-Machine

VMCALL = memanggil ke monitor VM

VMLAUNCH = menjalankan Virtual Machine

VMRESUME = melanjutkan Virtual Machie

VMXOFF = meninggalkan operasi VMX

VMXON = memasuki operasi VMX