| [pf](http://ref.x86asm.net/#column_pf) | [0F](http://ref.x86asm.net/#column_0F) | [po](http://ref.x86asm.net/#column_po) | [so](http://ref.x86asm.net/#column_so) | [o](http://ref.x86asm.net/#column_o) | [proc](http://ref.x86asm.net/#column_proc) | [st](http://ref.x86asm.net/#column_st) | [m](http://ref.x86asm.net/#column_m) | [rl](http://ref.x86asm.net/#column_rl) | [x](http://ref.x86asm.net/#column_x) | [mnemonic](http://ref.x86asm.net/#column_mnemonic) | [op1](http://ref.x86asm.net/#column_op) | [op2](http://ref.x86asm.net/#column_op) | [op3](http://ref.x86asm.net/#column_op) | [op4](http://ref.x86asm.net/#column_op) | [iext](http://ref.x86asm.net/#column_iext) | [tested f](http://ref.x86asm.net/#column_tested_f_modif_f_def_f_undef_f) | [modif f](http://ref.x86asm.net/#column_tested_f_modif_f_def_f_undef_f) | [def f](http://ref.x86asm.net/#column_tested_f_modif_f_def_f_undef_f) | [undef f](http://ref.x86asm.net/#column_tested_f_modif_f_def_f_undef_f) | [f values](http://ref.x86asm.net/#column_f_values) | [description, notes](http://ref.x86asm.net/#column_description_notes) |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 00 |  | r |  |  |  |  | L | ADD | **r/m8** | r8 |  |  |  |  | o..szapc | o..szapc |  |  | Add |
|  |  | 01 |  | r |  |  |  |  | L | ADD | **r/m16/32** | r16/32 |  |  |  |  | o..szapc | o..szapc |  |  | Add |
|  |  | 02 |  | r |  |  |  |  |  | ADD | **r8** | r/m8 |  |  |  |  | o..szapc | o..szapc |  |  | Add |
|  |  | 03 |  | r |  |  |  |  |  | ADD | **r16/32** | r/m16/32 |  |  |  |  | o..szapc | o..szapc |  |  | Add |
|  |  | 04 |  |  |  |  |  |  |  | ADD | **AL** | imm8 |  |  |  |  | o..szapc | o..szapc |  |  | Add |
|  |  | 05 |  |  |  |  |  |  |  | ADD | **eAX** | imm16/32 |  |  |  |  | o..szapc | o..szapc |  |  | Add |
|  |  | 06 |  |  |  |  |  |  |  | PUSH | ES |  |  |  |  |  |  |  |  |  | Push Word, Doubleword or Quadword Onto the Stack |
|  |  | 07 |  |  |  |  |  |  |  | POP | **ES** |  |  |  |  |  |  |  |  |  | Pop a Value from the Stack |
|  |  | 08 |  | r |  |  |  |  | L | OR | **r/m8** | r8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Inclusive OR |
|  |  | 09 |  | r |  |  |  |  | L | OR | **r/m16/32** | r16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Inclusive OR |
|  |  | 0A |  | r |  |  |  |  |  | OR | **r8** | r/m8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Inclusive OR |
|  |  | 0B |  | r |  |  |  |  |  | OR | **r16/32** | r/m16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Inclusive OR |
|  |  | 0C |  |  |  |  |  |  |  | OR | **AL** | imm8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Inclusive OR |
|  |  | 0D |  |  |  |  |  |  |  | OR | **eAX** | imm16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Inclusive OR |
|  |  | 0E |  |  |  |  |  |  |  | PUSH | CS |  |  |  |  |  |  |  |  |  | Push Word, Doubleword or Quadword Onto the Stack |
|  |  | 10 |  | r |  |  |  |  | L | ADC | **r/m8** | r8 |  |  |  | .......c | o..szapc | o..szapc |  |  | Add with Carry |
|  |  | 11 |  | r |  |  |  |  | L | ADC | **r/m16/32** | r16/32 |  |  |  | .......c | o..szapc | o..szapc |  |  | Add with Carry |
|  |  | 12 |  | r |  |  |  |  |  | ADC | **r8** | r/m8 |  |  |  | .......c | o..szapc | o..szapc |  |  | Add with Carry |
|  |  | 13 |  | r |  |  |  |  |  | ADC | **r16/32** | r/m16/32 |  |  |  | .......c | o..szapc | o..szapc |  |  | Add with Carry |
|  |  | 14 |  |  |  |  |  |  |  | ADC | **AL** | imm8 |  |  |  | .......c | o..szapc | o..szapc |  |  | Add with Carry |
|  |  | 15 |  |  |  |  |  |  |  | ADC | **eAX** | imm16/32 |  |  |  | .......c | o..szapc | o..szapc |  |  | Add with Carry |
|  |  | 16 |  |  |  |  |  |  |  | PUSH | SS |  |  |  |  |  |  |  |  |  | Push Word, Doubleword or Quadword Onto the Stack |
|  |  | 17 |  |  |  |  |  |  |  | POP | **SS** |  |  |  |  |  |  |  |  |  | Pop a Value from the Stack |
|  |  | 18 |  | r |  |  |  |  | L | SBB | **r/m8** | r8 |  |  |  | .......c | o..szapc | o..szapc |  |  | Integer Subtraction with Borrow |
|  |  | 19 |  | r |  |  |  |  | L | SBB | **r/m16/32** | r16/32 |  |  |  | .......c | o..szapc | o..szapc |  |  | Integer Subtraction with Borrow |
|  |  | 1A |  | r |  |  |  |  |  | SBB | **r8** | r/m8 |  |  |  | .......c | o..szapc | o..szapc |  |  | Integer Subtraction with Borrow |
|  |  | 1B |  | r |  |  |  |  |  | SBB | **r16/32** | r/m16/32 |  |  |  | .......c | o..szapc | o..szapc |  |  | Integer Subtraction with Borrow |
|  |  | 1C |  |  |  |  |  |  |  | SBB | **AL** | imm8 |  |  |  | .......c | o..szapc | o..szapc |  |  | Integer Subtraction with Borrow |
|  |  | 1D |  |  |  |  |  |  |  | SBB | **eAX** | imm16/32 |  |  |  | .......c | o..szapc | o..szapc |  |  | Integer Subtraction with Borrow |
|  |  | 1E |  |  |  |  |  |  |  | PUSH | DS |  |  |  |  |  |  |  |  |  | Push Word, Doubleword or Quadword Onto the Stack |
|  |  | 1F |  |  |  |  |  |  |  | POP | **DS** |  |  |  |  |  |  |  |  |  | Pop a Value from the Stack |
|  |  | 20 |  | r |  |  |  |  | L | AND | **r/m8** | r8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical AND |
|  |  | 21 |  | r |  |  |  |  | L | AND | **r/m16/32** | r16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical AND |
|  |  | 22 |  | r |  |  |  |  |  | AND | **r8** | r/m8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical AND |
|  |  | 23 |  | r |  |  |  |  |  | AND | **r16/32** | r/m16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical AND |
|  |  | 24 |  |  |  |  |  |  |  | AND | **AL** | imm8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical AND |
|  |  | 25 |  |  |  |  |  |  |  | AND | **eAX** | imm16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical AND |
|  |  | 27 |  |  |  |  |  |  |  | DAA | ***AL*** |  |  |  |  | .....a.c | o..szapc | ...szapc | o....... |  | Decimal Adjust AL after Addition |
|  |  | 28 |  | r |  |  |  |  | L | SUB | **r/m8** | r8 |  |  |  |  | o..szapc | o..szapc |  |  | Subtract |
|  |  | 29 |  | r |  |  |  |  | L | SUB | **r/m16/32** | r16/32 |  |  |  |  | o..szapc | o..szapc |  |  | Subtract |
|  |  | 2A |  | r |  |  |  |  |  | SUB | **r8** | r/m8 |  |  |  |  | o..szapc | o..szapc |  |  | Subtract |
|  |  | 2B |  | r |  |  |  |  |  | SUB | **r16/32** | r/m16/32 |  |  |  |  | o..szapc | o..szapc |  |  | Subtract |
|  |  | 2C |  |  |  |  |  |  |  | SUB | **AL** | imm8 |  |  |  |  | o..szapc | o..szapc |  |  | Subtract |
|  |  | 2D |  |  |  |  |  |  |  | SUB | **eAX** | imm16/32 |  |  |  |  | o..szapc | o..szapc |  |  | Subtract |
|  |  | 2F |  |  |  |  |  |  |  | DAS | ***AL*** |  |  |  |  | .....a.c | o..szapc | ...szapc | o....... |  | Decimal Adjust AL after Subtraction |
|  |  | 30 |  | r |  |  |  |  | L | XOR | **r/m8** | r8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Exclusive OR |
|  |  | 31 |  | r |  |  |  |  | L | XOR | **r/m16/32** | r16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Exclusive OR |
|  |  | 32 |  | r |  |  |  |  |  | XOR | **r8** | r/m8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Exclusive OR |
|  |  | 33 |  | r |  |  |  |  |  | XOR | **r16/32** | r/m16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Exclusive OR |
|  |  | 34 |  |  |  |  |  |  |  | XOR | **AL** | imm8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Exclusive OR |
|  |  | 35 |  |  |  |  |  |  |  | XOR | **eAX** | imm16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Exclusive OR |
|  |  | 37 |  |  |  |  |  |  |  | AAA | ***AL*** | ***AH*** |  |  |  | .....a.. | o..szapc | .....a.c | o..sz.p. |  | ASCII Adjust After Addition |
|  |  | 38 |  | r |  |  |  |  |  | CMP | r/m8 | r8 |  |  |  |  | o..szapc | o..szapc |  |  | Compare Two Operands |
|  |  | 39 |  | r |  |  |  |  |  | CMP | r/m16/32 | r16/32 |  |  |  |  | o..szapc | o..szapc |  |  | Compare Two Operands |
|  |  | 3A |  | r |  |  |  |  |  | CMP | r8 | r/m8 |  |  |  |  | o..szapc | o..szapc |  |  | Compare Two Operands |
|  |  | 3B |  | r |  |  |  |  |  | CMP | r16/32 | r/m16/32 |  |  |  |  | o..szapc | o..szapc |  |  | Compare Two Operands |
|  |  | 3C |  |  |  |  |  |  |  | CMP | AL | imm8 |  |  |  |  | o..szapc | o..szapc |  |  | Compare Two Operands |
|  |  | 3D |  |  |  |  |  |  |  | CMP | eAX | imm16/32 |  |  |  |  | o..szapc | o..szapc |  |  | Compare Two Operands |
|  |  | 3F |  |  |  |  |  |  |  | AAS | ***AL*** | ***AH*** |  |  |  | .....a.. | o..szapc | .....a.c | o..sz.p. |  | ASCII Adjust AL After Subtraction |
|  |  | 40+r | |  |  |  |  |  |  | INC | **r16/32** |  |  |  |  |  | o..szap. | o..szap. |  |  | Increment by 1 |
|  |  | 48+r | |  |  |  |  |  |  | DEC | **r16/32** |  |  |  |  |  | o..szap. | o..szap. |  |  | Decrement by 1 |
|  |  | 50+r | |  |  |  |  |  |  | PUSH | r16/32 |  |  |  |  |  |  |  |  |  | Push Word, Doubleword or Quadword Onto the Stack |
|  |  | 58+r | |  |  |  |  |  |  | POP | **r16/32** |  |  |  |  |  |  |  |  |  | Pop a Value from the Stack |
|  |  | 60 |  |  | 01+ |  |  |  |  | PUSHA | *AX* | *CX* | *DX* | ... |  |  |  |  |  |  | Push All General-Purpose Registers |
|  |  | 60 |  |  | 03+ |  |  |  |  | PUSHAD | *EAX* | *ECX* | *EDX* | ... |  |  |  |  |  |  | Push All General-Purpose Registers |
|  |  | 61 |  |  | 01+ |  |  |  |  | POPA | ***DI*** | ***SI*** | ***BP*** | ... |  |  |  |  |  |  | Pop All General-Purpose Registers |
|  |  | 61 |  |  | 03+ |  |  |  |  | POPAD | ***EDI*** | ***ESI*** | ***EBP*** | ... |  |  |  |  |  |  | Pop All General-Purpose Registers |
|  |  | 62 |  | r | 01+ |  |  | f |  | BOUND | r16/32 | m16/32&16/32 | *eFlags* |  |  |  | ..i..... | ..i..... |  | ..i..... | Check Array Index Against Bounds |
|  |  | 63 |  | r | 02+ |  |  |  |  | ARPL | r/m16 | r16 |  |  |  |  | ....z... | ....z... |  |  | Adjust RPL Field of Segment Selector |
|  |  | 68 |  |  | 01+ |  |  |  |  | PUSH | imm16/32 |  |  |  |  |  |  |  |  |  | Push Word, Doubleword or Quadword Onto the Stack |
|  |  | 69 |  | r | 01+ |  |  |  |  | IMUL | **r16/32** | r/m16/32 | imm16/32 |  |  |  | o..szapc | o......c | ...szap. |  | Signed Multiply |
|  |  | 6A |  |  | 01+ |  |  |  |  | PUSH | imm8 |  |  |  |  |  |  |  |  |  | Push Word, Doubleword or Quadword Onto the Stack |
|  |  | 6B |  | r | 01+ |  |  |  |  | IMUL | **r16/32** | r/m16/32 | imm8 |  |  |  | o..szapc | o......c | ...szap. |  | Signed Multiply |
|  |  | 6C |  |  | 01+ |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | INS | **m8** | DX |  |  |  | .d...... |  |  |  |  | Input from Port to String |
| INSB | ***m8*** | *DX* |  |  |
|  |  | 6D |  |  | 01+ |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | INS | **m16** | DX |  |  |  | .d...... |  |  |  |  | Input from Port to String |
| INSW | ***m16*** | *DX* |  |  |
|  |  | 6D |  |  | 03+ |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | INS | **m16/32** | DX |  |  |  | .d...... |  |  |  |  | Input from Port to String |
| INSD | ***m32*** | *DX* |  |  |
|  |  | 6E |  |  | 01+ |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | OUTS | **DX** | m8 |  |  |  | .d...... |  |  |  |  | Output String to Port |
| OUTSB | ***DX*** | *m8* |  |  |
|  |  | 6F |  |  | 01+ |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | OUTS | **DX** | m16 |  |  |  | .d...... |  |  |  |  | Output String to Port |
| OUTSW | ***DX*** | *m16* |  |  |
|  |  | 6F |  |  | 03+ |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | OUTS | **DX** | m16/32 |  |  |  | .d...... |  |  |  |  | Output String to Port |
| OUTSD | ***DX*** | *m32* |  |  |
|  |  | 70 |  |  |  |  |  |  |  | JO | rel8 |  |  |  |  | o....... |  |  |  |  | Jump short if overflow (OF=1) |
|  |  | 71 |  |  |  |  |  |  |  | JNO | rel8 |  |  |  |  | o....... |  |  |  |  | Jump short if not overflow (OF=0) |
|  |  | 72 |  |  |  |  |  |  |  | JB | rel8 |  |  |  |  | .......c |  |  |  |  | Jump short if below/not above or equal/carry (CF=1) |
| JNAE | rel8 |  |  |  |
| JC | rel8 |  |  |  |
|  |  | 73 |  |  |  |  |  |  |  | JNB | rel8 |  |  |  |  | .......c |  |  |  |  | Jump short if not below/above or equal/not carry (CF=0) |
| JAE | rel8 |  |  |  |
| JNC | rel8 |  |  |  |
|  |  | 74 |  |  |  |  |  |  |  | JZ | rel8 |  |  |  |  | ....z... |  |  |  |  | Jump short if zero/equal (ZF=0) |
| JE | rel8 |  |  |  |
|  |  | 75 |  |  |  |  |  |  |  | JNZ | rel8 |  |  |  |  | ....z... |  |  |  |  | Jump short if not zero/not equal (ZF=1) |
| JNE | rel8 |  |  |  |
|  |  | 76 |  |  |  |  |  |  |  | JBE | rel8 |  |  |  |  | ....z..c |  |  |  |  | Jump short if below or equal/not above (CF=1 AND ZF=1) |
| JNA | rel8 |  |  |  |
|  |  | 77 |  |  |  |  |  |  |  | JNBE | rel8 |  |  |  |  | ....z..c |  |  |  |  | Jump short if not below or equal/above (CF=0 AND ZF=0) |
| JA | rel8 |  |  |  |
|  |  | 78 |  |  |  |  |  |  |  | JS | rel8 |  |  |  |  | ...s.... |  |  |  |  | Jump short if sign (SF=1) |
|  |  | 79 |  |  |  |  |  |  |  | JNS | rel8 |  |  |  |  | ...s.... |  |  |  |  | Jump short if not sign (SF=0) |
|  |  | 7A |  |  |  |  |  |  |  | JP | rel8 |  |  |  |  | ......p. |  |  |  |  | Jump short if parity/parity even (PF=1) |
| JPE | rel8 |  |  |  |
|  |  | 7B |  |  |  |  |  |  |  | JNP | rel8 |  |  |  |  | ......p. |  |  |  |  | Jump short if not parity/parity odd |
| JPO | rel8 |  |  |  |
|  |  | 7C |  |  |  |  |  |  |  | JL | rel8 |  |  |  |  | o..s.... |  |  |  |  | Jump short if less/not greater (SF!=OF) |
| JNGE | rel8 |  |  |  |
|  |  | 7D |  |  |  |  |  |  |  | JNL | rel8 |  |  |  |  | o..s.... |  |  |  |  | Jump short if not less/greater or equal (SF=OF) |
| JGE | rel8 |  |  |  |
|  |  | 7E |  |  |  |  |  |  |  | JLE | rel8 |  |  |  |  | o..sz... |  |  |  |  | Jump short if less or equal/not greater ((ZF=1) OR (SF!=OF)) |
| JNG | rel8 |  |  |  |
|  |  | 7F |  |  |  |  |  |  |  | JNLE | rel8 |  |  |  |  | o..sz... |  |  |  |  | Jump short if not less nor equal/greater ((ZF=0) AND (SF=OF)) |
| JG | rel8 |  |  |  |
|  |  | 80 |  | 0 |  |  |  |  | L | ADD | **r/m8** | imm8 |  |  |  |  | o..szapc | o..szapc |  |  | Add |
|  |  | 80 |  | 1 |  |  |  |  | L | OR | **r/m8** | imm8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Inclusive OR |
|  |  | 80 |  | 2 |  |  |  |  | L | ADC | **r/m8** | imm8 |  |  |  | .......c | o..szapc | o..szapc |  |  | Add with Carry |
|  |  | 80 |  | 3 |  |  |  |  | L | SBB | **r/m8** | imm8 |  |  |  | .......c | o..szapc | o..szapc |  |  | Integer Subtraction with Borrow |
|  |  | 80 |  | 4 |  |  |  |  | L | AND | **r/m8** | imm8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical AND |
|  |  | 80 |  | 5 |  |  |  |  | L | SUB | **r/m8** | imm8 |  |  |  |  | o..szapc | o..szapc |  |  | Subtract |
|  |  | 80 |  | 6 |  |  |  |  | L | XOR | **r/m8** | imm8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Exclusive OR |
|  |  | 80 |  | 7 |  |  |  |  |  | CMP | r/m8 | imm8 |  |  |  |  | o..szapc | o..szapc |  |  | Compare Two Operands |
|  |  | 81 |  | 0 |  |  |  |  | L | ADD | **r/m16/32** | imm16/32 |  |  |  |  | o..szapc | o..szapc |  |  | Add |
|  |  | 81 |  | 1 |  |  |  |  | L | OR | **r/m16/32** | imm16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Inclusive OR |
|  |  | 81 |  | 2 |  |  |  |  | L | ADC | **r/m16/32** | imm16/32 |  |  |  | .......c | o..szapc | o..szapc |  |  | Add with Carry |
|  |  | 81 |  | 3 |  |  |  |  | L | SBB | **r/m16/32** | imm16/32 |  |  |  | .......c | o..szapc | o..szapc |  |  | Integer Subtraction with Borrow |
|  |  | 81 |  | 4 |  |  |  |  | L | AND | **r/m16/32** | imm16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical AND |
|  |  | 81 |  | 5 |  |  |  |  | L | SUB | **r/m16/32** | imm16/32 |  |  |  |  | o..szapc | o..szapc |  |  | Subtract |
|  |  | 81 |  | 6 |  |  |  |  | L | XOR | **r/m16/32** | imm16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Exclusive OR |
|  |  | 81 |  | 7 |  |  |  |  |  | CMP | r/m16/32 | imm16/32 |  |  |  |  | o..szapc | o..szapc |  |  | Compare Two Operands |
|  |  | 82 |  | 0 |  |  |  |  | L | ADD | **r/m8** | imm8 |  |  |  |  | o..szapc | o..szapc |  |  | Add |
|  |  | 82 |  | 1 |  |  |  |  | L | OR | **r/m8** | imm8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Inclusive OR |
|  |  | 82 |  | 2 |  |  |  |  | L | ADC | **r/m8** | imm8 |  |  |  | .......c | o..szapc | o..szapc |  |  | Add with Carry |
|  |  | 82 |  | 3 |  |  |  |  | L | SBB | **r/m8** | imm8 |  |  |  | .......c | o..szapc | o..szapc |  |  | Integer Subtraction with Borrow |
|  |  | 82 |  | 4 |  |  |  |  | L | AND | **r/m8** | imm8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical AND |
|  |  | 82 |  | 5 |  |  |  |  | L | SUB | **r/m8** | imm8 |  |  |  |  | o..szapc | o..szapc |  |  | Subtract |
|  |  | 82 |  | 6 |  |  |  |  | L | XOR | **r/m8** | imm8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Exclusive OR |
|  |  | 82 |  | 7 |  |  |  |  |  | CMP | r/m8 | imm8 |  |  |  |  | o..szapc | o..szapc |  |  | Compare Two Operands |
|  |  | 83 |  | 0 |  |  |  |  | L | ADD | **r/m16/32** | imm8 |  |  |  |  | o..szapc | o..szapc |  |  | Add |
|  |  | 83 |  | 1 | 03+ |  |  |  | L | OR | **r/m16/32** | imm8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Inclusive OR |
|  |  | 83 |  | 2 |  |  |  |  | L | ADC | **r/m16/32** | imm8 |  |  |  | .......c | o..szapc | o..szapc |  |  | Add with Carry |
|  |  | 83 |  | 3 |  |  |  |  | L | SBB | **r/m16/32** | imm8 |  |  |  | .......c | o..szapc | o..szapc |  |  | Integer Subtraction with Borrow |
|  |  | 83 |  | 4 | 03+ |  |  |  | L | AND | **r/m16/32** | imm8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical AND |
|  |  | 83 |  | 5 |  |  |  |  | L | SUB | **r/m16/32** | imm8 |  |  |  |  | o..szapc | o..szapc |  |  | Subtract |
|  |  | 83 |  | 6 | 03+ |  |  |  | L | XOR | **r/m16/32** | imm8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Exclusive OR |
|  |  | 83 |  | 7 |  |  |  |  |  | CMP | r/m16/32 | imm8 |  |  |  |  | o..szapc | o..szapc |  |  | Compare Two Operands |
|  |  | 84 |  | r |  |  |  |  |  | TEST | r/m8 | r8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Compare |
|  |  | 85 |  | r |  |  |  |  |  | TEST | r/m16/32 | r16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Compare |
|  |  | 86 |  | r |  |  |  |  | L | XCHG | **r8** | **r/m8** |  |  |  |  |  |  |  |  | Exchange Register/Memory with Register |
|  |  | 87 |  | r |  |  |  |  | L | XCHG | **r16/32** | **r/m16/32** |  |  |  |  |  |  |  |  | Exchange Register/Memory with Register |
|  |  | 88 |  | r |  |  |  |  |  | MOV | **r/m8** | r8 |  |  |  |  |  |  |  |  | Move |
|  |  | 89 |  | r |  |  |  |  |  | MOV | **r/m16/32** | r16/32 |  |  |  |  |  |  |  |  | Move |
|  |  | 8A |  | r |  |  |  |  |  | MOV | **r8** | r/m8 |  |  |  |  |  |  |  |  | Move |
|  |  | 8B |  | r |  |  |  |  |  | MOV | **r16/32** | r/m16/32 |  |  |  |  |  |  |  |  | Move |
|  |  | 8C |  | r |  |  |  |  |  | MOV | **m16** | Sreg |  |  |  |  |  |  |  |  | Move |
| MOV | **r16/32** | Sreg |  |  |
|  |  | 8D |  | r |  |  |  |  |  | LEA | **r16/32** | m |  |  |  |  |  |  |  |  | Load Effective Address |
|  |  | 8E |  | r |  |  |  |  |  | MOV | **Sreg** | r/m16 |  |  |  |  |  |  |  |  | Move |
|  |  | 8F |  | 0 |  |  |  |  |  | POP | **r/m16/32** |  |  |  |  |  |  |  |  |  | Pop a Value from the Stack |
|  |  | 90+r | |  |  |  |  |  |  | XCHG | **r16/32** | **eAX** |  |  |  |  |  |  |  |  | Exchange Register/Memory with Register |
|  |  | 90 |  |  |  | D[1](http://ref.x86asm.net/coder32.html#gen_note_90_NOP) |  |  |  | NOP |  |  |  |  |  |  |  |  |  |  | No Operation |
| F3 |  | 90 |  |  | P4+ |  |  |  |  | PAUSE |  |  |  |  | sse2 |  |  |  |  |  | Spin Loop Hint |
|  |  | 98 |  |  |  |  |  |  |  | CBW | ***AX*** | *AL* |  |  |  |  |  |  |  |  | Convert Byte to Word |
|  |  | 98 |  |  | 03+ |  |  |  |  | CWDE | ***EAX*** | *AX* |  |  |  |  |  |  |  |  | Convert Word to Doubleword |
|  |  | 99 |  |  |  |  |  |  |  | CWD | ***DX*** | *AX* |  |  |  |  |  |  |  |  | Convert Word to Doubleword |
|  |  | 99 |  |  | 03+ |  |  |  |  | CDQ | ***EDX*** | *EAX* |  |  |  |  |  |  |  |  | Convert Doubleword to Quadword |
|  |  | 9A |  |  |  |  |  |  |  | CALLF | ptr16:16/32 |  |  |  |  |  |  |  |  |  | Call Procedure |
|  |  | 9B |  |  |  |  |  |  |  | FWAIT |  |  |  |  |  |  | 0123 |  | 0123 |  | Check pending unmasked floating-point exceptions |
| WAIT |  |  |  |  |
| 9B |  |  |  |  |  |  |  |  |  | *no mnemonic* | | | | |  |  | 0123 |  | 0123 |  | Wait Prefix |
|  |  | 9C |  |  |  |  |  |  |  | PUSHF | *Flags* |  |  |  |  |  |  |  |  |  | Push FLAGS Register onto the Stack |
|  |  | 9C |  |  | 03+ |  |  |  |  | PUSHFD | *EFlags* |  |  |  |  |  |  |  |  |  | Push eFLAGS Register onto the Stack |
|  |  | 9D |  |  |  |  |  |  |  | POPF | ***Flags*** |  |  |  |  |  |  |  |  |  | Pop Stack into FLAGS Register |
|  |  | 9D |  |  | 03+ |  |  |  |  | POPFD | ***EFlags*** |  |  |  |  |  |  |  |  |  | Pop Stack into eFLAGS Register |
|  |  | 9E |  |  |  |  |  |  |  | SAHF | *AH* |  |  |  |  |  | ...szapc | ...szapc |  |  | Store AH into Flags |
|  |  | 9F |  |  |  |  |  |  |  | LAHF | ***AH*** |  |  |  |  | ...szapc |  |  |  |  | Load Status Flags into AH Register |
|  |  | A0 |  |  |  |  |  |  |  | MOV | **AL** | moffs8 |  |  |  |  |  |  |  |  | Move |
|  |  | A1 |  |  |  |  |  |  |  | MOV | **eAX** | moffs16/32 |  |  |  |  |  |  |  |  | Move |
|  |  | A2 |  |  |  |  |  |  |  | MOV | **moffs8** | AL |  |  |  |  |  |  |  |  | Move |
|  |  | A3 |  |  |  |  |  |  |  | MOV | **moffs16/32** | eAX |  |  |  |  |  |  |  |  | Move |
|  |  | A4 |  |  |  |  |  |  |  | MOVS | **m8** | m8 |  |  |  | .d...... |  |  |  |  | Move Data from String to String |
| MOVSB | ***m8*** | *m8* |  |  |
|  |  | A5 |  |  |  |  |  |  |  | MOVS | **m16** | m16 |  |  |  | .d...... |  |  |  |  | Move Data from String to String |
| MOVSW | ***m16*** | *m16* |  |  |
|  |  | A5 |  |  | 03+ |  |  |  |  | MOVS | **m16/32** | m16/32 |  |  |  | .d...... |  |  |  |  | Move Data from String to String |
| MOVSD | ***m32*** | *m32* |  |  |
|  |  | A6 |  |  |  |  |  |  |  | CMPS | m8 | m8 |  |  |  | .d...... | o..szapc | o..szapc |  |  | Compare String Operands |
| CMPSB | *m8* | *m8* |  |  |
|  |  | A7 |  |  |  |  |  |  |  | CMPS | m16 | m16 |  |  |  | .d...... | o..szapc | o..szapc |  |  | Compare String Operands |
| CMPSW | *m16* | *m16* |  |  |
|  |  | A7 |  |  | 03+ |  |  |  |  | CMPS | m16/32 | m16/32 |  |  |  | .d...... | o..szapc | o..szapc |  |  | Compare String Operands |
| CMPSD | *m32* | *m32* |  |  |
|  |  | A8 |  |  |  |  |  |  |  | TEST | AL | imm8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Compare |
|  |  | A9 |  |  |  |  |  |  |  | TEST | eAX | imm16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Compare |
|  |  | AA |  |  |  |  |  |  |  | STOS | **m8** | *AL* |  |  |  | .d...... |  |  |  |  | Store String |
| STOSB | ***m8*** | *AL* |  |  |
|  |  | AB |  |  |  |  |  |  |  | STOS | **m16** | *AX* |  |  |  | .d...... |  |  |  |  | Store String |
| STOSW | ***m16*** | *AX* |  |  |
|  |  | AB |  |  | 03+ |  |  |  |  | STOS | **m16/32** | *eAX* |  |  |  | .d...... |  |  |  |  | Store String |
| STOSD | ***m32*** | *EAX* |  |  |
|  |  | AC |  |  |  |  |  |  |  | LODS | ***AL*** | m8 |  |  |  | .d...... |  |  |  |  | Load String |
| LODSB | ***AL*** | *m8* |  |  |
|  |  | AD |  |  |  |  |  |  |  | LODS | ***AX*** | m16 |  |  |  | .d...... |  |  |  |  | Load String |
| LODSW | ***AX*** | *m16* |  |  |
|  |  | AD |  |  | 03+ |  |  |  |  | LODS | ***eAX*** | m16/32 |  |  |  | .d...... |  |  |  |  | Load String |
| LODSD | ***EAX*** | *m32* |  |  |
|  |  | AE |  |  |  |  |  |  |  | SCAS | m8 | *AL* |  |  |  | .d...... | o..szapc | o..szapc |  |  | Scan String |
| SCASB | *m8* | *AL* |  |  |
|  |  | AF |  |  |  |  |  |  |  | SCAS | m16 | *AX* |  |  |  | .d...... | o..szapc | o..szapc |  |  | Scan String |
| SCASW | *m16* | *AX* |  |  |
|  |  | AF |  |  | 03+ |  |  |  |  | SCAS | m16/32 | *eAX* |  |  |  | .d...... | o..szapc | o..szapc |  |  | Scan String |
| SCASD | *m32* | *EAX* |  |  |
|  |  | B0+r | |  |  |  |  |  |  | MOV | **r8** | imm8 |  |  |  |  |  |  |  |  | Move |
|  |  | B8+r | |  |  |  |  |  |  | MOV | **r16/32** | imm16/32 |  |  |  |  |  |  |  |  | Move |
|  |  | C0 |  | 0 | 01+ |  |  |  |  | ROL | **r/m8** | imm8 |  |  |  |  | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | C0 |  | 1 | 01+ |  |  |  |  | ROR | **r/m8** | imm8 |  |  |  |  | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | C0 |  | 2 | 01+ |  |  |  |  | RCL | **r/m8** | imm8 |  |  |  | .......c | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | C0 |  | 3 | 01+ |  |  |  |  | RCR | **r/m8** | imm8 |  |  |  | .......c | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | C0 |  | 4 | 01+ |  |  |  |  | SHL | **r/m8** | imm8 |  |  |  |  | o..szapc | o..sz.pc | o....a.c |  | Shift |
| SAL | **r/m8** | imm8 |  |  |
|  |  | C0 |  | 5 | 01+ |  |  |  |  | SHR | **r/m8** | imm8 |  |  |  |  | o..szapc | o..sz.pc | o....a.c |  | Shift |
|  |  | C0 |  | 6 | 01+ | U[2](http://ref.x86asm.net/coder32.html#gen_note_SAL_C0_4_C1_4_D0_4_D1_4) |  |  |  | SAL | **r/m8** | imm8 |  |  |  |  | o..szapc | o..sz.pc | o....a.c |  | Shift |
| SHL | **r/m8** | imm8 |  |  |
|  |  | C0 |  | 7 | 01+ |  |  |  |  | SAR | **r/m8** | imm8 |  |  |  |  | o..szapc | o..sz.pc | o....a.. |  | Shift |
|  |  | C1 |  | 0 | 01+ |  |  |  |  | ROL | **r/m16/32** | imm8 |  |  |  |  | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | C1 |  | 1 | 01+ |  |  |  |  | ROR | **r/m16/32** | imm8 |  |  |  |  | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | C1 |  | 2 | 01+ |  |  |  |  | RCL | **r/m16/32** | imm8 |  |  |  | .......c | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | C1 |  | 3 | 01+ |  |  |  |  | RCR | **r/m16/32** | imm8 |  |  |  | .......c | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | C1 |  | 4 | 01+ |  |  |  |  | SHL | **r/m16/32** | imm8 |  |  |  |  | o..szapc | o..sz.pc | o....a.c |  | Shift |
| SAL | **r/m16/32** | imm8 |  |  |
|  |  | C1 |  | 5 | 01+ |  |  |  |  | SHR | **r/m16/32** | imm8 |  |  |  |  | o..szapc | o..sz.pc | o....a.c |  | Shift |
|  |  | C1 |  | 6 | 01+ | U[2](http://ref.x86asm.net/coder32.html#gen_note_SAL_C0_4_C1_4_D0_4_D1_4) |  |  |  | SAL | **r/m16/32** | imm8 |  |  |  |  | o..szapc | o..sz.pc | o....a.c |  | Shift |
| SHL | **r/m16/32** | imm8 |  |  |
|  |  | C1 |  | 7 | 01+ |  |  |  |  | SAR | **r/m16/32** | imm8 |  |  |  |  | o..szapc | o..sz.pc | o....a.. |  | Shift |
|  |  | C2 |  |  |  |  |  |  |  | RETN | imm16 |  |  |  |  |  |  |  |  |  | Return from procedure |
|  |  | C3 |  |  |  |  |  |  |  | RETN |  |  |  |  |  |  |  |  |  |  | Return from procedure |
|  |  | C4 |  | r |  |  |  |  |  | LES | ***ES*** | **r16/32** | m16:16/32 |  |  |  |  |  |  |  | Load Far Pointer |
|  |  | C5 |  | r |  |  |  |  |  | LDS | ***DS*** | **r16/32** | m16:16/32 |  |  |  |  |  |  |  | Load Far Pointer |
|  |  | C6 |  | 0 |  |  |  |  |  | MOV | **r/m8** | imm8 |  |  |  |  |  |  |  |  | Move |
|  |  | C7 |  | 0 |  |  |  |  |  | MOV | **r/m16/32** | imm16/32 |  |  |  |  |  |  |  |  | Move |
|  |  | C8 |  |  | 01+ |  |  |  |  | ENTER | ***eBP*** | imm16 | imm8 |  |  |  |  |  |  |  | Make Stack Frame for Procedure Parameters |
|  |  | C9 |  |  | 01+ |  |  |  |  | LEAVE | ***eBP*** |  |  |  |  |  |  |  |  |  | High Level Procedure Exit |
|  |  | CA |  |  |  |  |  | f |  | RETF | imm16 |  |  |  |  |  |  |  |  |  | Return from procedure |
|  |  | CB |  |  |  |  |  | f |  | RETF |  |  |  |  |  |  |  |  |  |  | Return from procedure |
|  |  | CC |  |  |  |  |  | f |  | INT | 3 | *eFlags* |  |  |  |  | ..i..... | ..i..... |  | ..i..... | Call to Interrupt Procedure |
|  |  | CD |  |  |  |  |  | f |  | INT | imm8 | *eFlags* |  |  |  |  | ..i..... | ..i..... |  | ..i..... | Call to Interrupt Procedure |
|  |  | CE |  |  |  |  |  | f |  | INTO | *eFlags* |  |  |  |  | o....... | ..i..... | ..i..... |  | ..i..... | Call to Interrupt Procedure |
|  |  | CF |  |  |  |  |  | f |  | IRET | ***Flags*** |  |  |  |  |  |  |  |  |  | Interrupt Return |
|  |  | CF |  |  | 03+ |  |  | f |  | IRETD | ***EFlags*** |  |  |  |  |  |  |  |  |  | Interrupt Return |
|  |  | D0 |  | 0 |  |  |  |  |  | ROL | **r/m8** | 1 |  |  |  |  | o..szapc | o..szapc |  |  | Rotate |
|  |  | D0 |  | 1 |  |  |  |  |  | ROR | **r/m8** | 1 |  |  |  |  | o..szapc | o..szapc |  |  | Rotate |
|  |  | D0 |  | 2 |  |  |  |  |  | RCL | **r/m8** | 1 |  |  |  | .......c | o..szapc | o..szapc |  |  | Rotate |
|  |  | D0 |  | 3 |  |  |  |  |  | RCR | **r/m8** | 1 |  |  |  | .......c | o..szapc | o..szapc |  |  | Rotate |
|  |  | D0 |  | 4 |  |  |  |  |  | SHL | **r/m8** | 1 |  |  |  |  | o..szapc | o..sz.pc | .....a.. |  | Shift |
| SAL | **r/m8** | 1 |  |  |
|  |  | D0 |  | 5 |  |  |  |  |  | SHR | **r/m8** | 1 |  |  |  |  | o..szapc | o..sz.pc | .....a.. |  | Shift |
|  |  | D0 |  | 6 |  | U[2](http://ref.x86asm.net/coder32.html#gen_note_SAL_C0_4_C1_4_D0_4_D1_4) |  |  |  | SAL | **r/m8** | 1 |  |  |  |  | o..szapc | o..sz.pc | .....a.. |  | Shift |
| SHL | **r/m8** | 1 |  |  |
|  |  | D0 |  | 7 |  |  |  |  |  | SAR | **r/m8** | 1 |  |  |  |  | o..szapc | o..sz.pc | .....a.. |  | Shift |
|  |  | D1 |  | 0 |  |  |  |  |  | ROL | **r/m16/32** | 1 |  |  |  |  | o..szapc | o..szapc |  |  | Rotate |
|  |  | D1 |  | 1 |  |  |  |  |  | ROR | **r/m16/32** | 1 |  |  |  |  | o..szapc | o..szapc |  |  | Rotate |
|  |  | D1 |  | 2 |  |  |  |  |  | RCL | **r/m16/32** | 1 |  |  |  | .......c | o..szapc | o..szapc |  |  | Rotate |
|  |  | D1 |  | 3 |  |  |  |  |  | RCR | **r/m16/32** | 1 |  |  |  | .......c | o..szapc | o..szapc |  |  | Rotate |
|  |  | D1 |  | 4 |  |  |  |  |  | SHL | **r/m16/32** | 1 |  |  |  |  | o..szapc | o..sz.pc | .....a.. |  | Shift |
| SAL | **r/m16/32** | 1 |  |  |
|  |  | D1 |  | 5 |  |  |  |  |  | SHR | **r/m16/32** | 1 |  |  |  |  | o..szapc | o..sz.pc | .....a.. |  | Shift |
|  |  | D1 |  | 6 |  | U[2](http://ref.x86asm.net/coder32.html#gen_note_SAL_C0_4_C1_4_D0_4_D1_4) |  |  |  | SAL | **r/m16/32** | 1 |  |  |  |  | o..szapc | o..sz.pc | .....a.. |  | Shift |
| SHL | **r/m16/32** | 1 |  |  |
|  |  | D1 |  | 7 |  |  |  |  |  | SAR | **r/m16/32** | 1 |  |  |  |  | o..szapc | o..sz.pc | .....a.. |  | Shift |
|  |  | D2 |  | 0 |  |  |  |  |  | ROL | **r/m8** | CL |  |  |  |  | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | D2 |  | 1 |  |  |  |  |  | ROR | **r/m8** | CL |  |  |  |  | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | D2 |  | 2 |  |  |  |  |  | RCL | **r/m8** | CL |  |  |  | .......c | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | D2 |  | 3 |  |  |  |  |  | RCR | **r/m8** | CL |  |  |  | .......c | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | D2 |  | 4 |  |  |  |  |  | SHL | **r/m8** | CL |  |  |  |  | o..szapc | o..sz.pc | o....a.c |  | Shift |
| SAL | **r/m8** | CL |  |  |
|  |  | D2 |  | 5 |  |  |  |  |  | SHR | **r/m8** | CL |  |  |  |  | o..szapc | o..sz.pc | o....a.c |  | Shift |
|  |  | D2 |  | 6 |  | U[2](http://ref.x86asm.net/coder32.html#gen_note_SAL_C0_4_C1_4_D0_4_D1_4) |  |  |  | SAL | **r/m8** | CL |  |  |  |  | o..szapc | o..sz.pc | o....a.c |  | Shift |
| SHL | **r/m8** | CL |  |  |
|  |  | D2 |  | 7 |  |  |  |  |  | SAR | **r/m8** | CL |  |  |  |  | o..szapc | o..sz.pc | o....a.. |  | Shift |
|  |  | D3 |  | 0 |  |  |  |  |  | ROL | **r/m16/32** | CL |  |  |  |  | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | D3 |  | 1 |  |  |  |  |  | ROR | **r/m16/32** | CL |  |  |  |  | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | D3 |  | 2 |  |  |  |  |  | RCL | **r/m16/32** | CL |  |  |  | .......c | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | D3 |  | 3 |  |  |  |  |  | RCR | **r/m16/32** | CL |  |  |  | .......c | o..szapc | o..szapc | o....... |  | Rotate |
|  |  | D3 |  | 4 |  |  |  |  |  | SHL | **r/m16/32** | CL |  |  |  |  | o..szapc | o..sz.pc | o....a.c |  | Shift |
| SAL | **r/m16/32** | CL |  |  |
|  |  | D3 |  | 5 |  |  |  |  |  | SHR | **r/m16/32** | CL |  |  |  |  | o..szapc | o..sz.pc | o....a.c |  | Shift |
|  |  | D3 |  | 6 |  | U[2](http://ref.x86asm.net/coder32.html#gen_note_SAL_C0_4_C1_4_D0_4_D1_4) |  |  |  | SAL | **r/m16/32** | CL |  |  |  |  | o..szapc | o..sz.pc | o....a.c |  | Shift |
| SHL | **r/m16/32** | CL |  |  |
|  |  | D3 |  | 7 |  |  |  |  |  | SAR | **r/m16/32** | CL |  |  |  |  | o..szapc | o..sz.pc | .....a.. |  | Shift |
|  |  | D4 | 0A |  |  |  |  |  |  | AAM | ***AL*** | ***AH*** |  |  |  |  | o..szapc | ...sz.p. | o....a.c |  | ASCII Adjust AX After Multiply |
|  |  | D4 |  |  |  |  |  |  |  | *AMX* | ***AL*** | ***AH*** | imm8 |  |  |  | o..szapc | ...sz.p. | o....a.c |  | Adjust AX After Multiply |
|  |  | D5 | 0A |  |  |  |  |  |  | AAD | ***AL*** | ***AH*** |  |  |  |  | o..szapc | ...sz.p. | o....a.c |  | ASCII Adjust AX Before Division |
|  |  | D5 |  |  |  |  |  |  |  | *ADX* | ***AL*** | ***AH*** | imm8 |  |  |  | o..szapc | ...sz.p. | o....a.c |  | Adjust AX Before Division |
|  |  | D6 |  |  | 02+ | D[3](http://ref.x86asm.net/coder32.html#gen_note_undefined_D6_F1) |  |  |  | *undefined* | | | | |  |  |  |  |  |  | Undefined and Reserved; Does not Generate #UD |
|  |  | D6 |  |  | 02+ | U[4](http://ref.x86asm.net/coder32.html#gen_note_u_SALC_D6) |  |  |  | SALC | ***AL*** |  |  |  |  | .......c |  |  |  |  | Set AL If Carry |
| SETALC | ***AL*** |  |  |  |
|  |  | D7 |  |  |  |  |  |  |  | XLAT | ***AL*** | m8 |  |  |  |  |  |  |  |  | Table Look-up Translation |
| XLATB | ***AL*** | *m8* |  |  |
|  |  | D8 |  | 0 |  |  |  |  |  | FADD | ***ST*** | m32real |  |  |  |  | 0123 | .1.. | 0.23 |  | Add |
| FADD | **ST** | STi |  |  |
|  |  | D8 |  | 1 |  |  |  |  |  | FMUL | ***ST*** | m32real |  |  |  |  | 0123 | .1.. | 0.23 |  | Multiply |
| FMUL | **ST** | STi |  |  |
|  |  | D8 |  | 2 |  |  |  |  |  | FCOM | *ST* | STi/m32real |  |  |  |  | 0123 | 0123 |  |  | Compare Real |
|  |  | D8 | D1 | 2 |  |  |  |  |  | FCOM | *ST* | *ST1* |  |  |  |  | 0123 | 0123 |  |  | Compare Real |
|  |  | D8 |  | 3 |  |  |  |  | p | FCOMP | *ST* | STi/m32real |  |  |  |  | 0123 | 0123 |  |  | Compare Real and Pop |
|  |  | D8 | D9 | 3 |  |  |  |  | p | FCOMP | *ST* | *ST1* |  |  |  |  | 0123 | 0123 |  |  | Compare Real and Pop |
|  |  | D8 |  | 4 |  |  |  |  |  | FSUB | ***ST*** | m32real |  |  |  |  | 0123 | .1.. | 0.23 |  | Subtract |
| FSUB | **ST** | STi |  |  |
|  |  | D8 |  | 5 |  |  |  |  |  | FSUBR | ***ST*** | m32real |  |  |  |  | 0123 | .1.. | 0.23 |  | Reverse Subtract |
| FSUBR | **ST** | STi |  |  |
|  |  | D8 |  | 6 |  |  |  |  |  | FDIV | ***ST*** | m32real |  |  |  |  | 0123 | .1.. | 0.23 |  | Divide |
| FDIV | **ST** | STi |  |  |
|  |  | D8 |  | 7 |  |  |  |  |  | FDIVR | ***ST*** | m32real |  |  |  |  | 0123 | .1.. | 0.23 |  | Reverse Divide |
| FDIVR | **ST** | STi |  |  |
|  |  | D9 |  | 0 |  |  |  |  | s | FLD | ***ST*** | STi/m32real |  |  |  |  | 0123 | .1.. | 0.23 |  | Load Floating Point Value |
|  |  | D9 |  | 1 |  |  |  |  |  | FXCH | ***ST*** | **STi** |  |  |  |  | 0123 | .1.. | 0.23 |  | Exchange Register Contents |
|  |  | D9 | C9 | 1 |  |  |  |  |  | FXCH | ***ST*** | ***ST1*** |  |  |  |  | 0123 | .1.. | 0.23 |  | Exchange Register Contents |
|  |  | D9 |  | 2 |  |  |  |  |  | FST | **m32real** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Store Floating Point Value |
|  |  | D9 | D0 | 2 |  |  |  |  |  | FNOP |  |  |  |  |  |  | 0123 |  | 0123 |  | No Operation |
|  |  | D9 |  | 3 |  |  |  |  | p | FSTP | **m32real** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Store Floating Point Value and Pop |
|  |  | D9 |  | 3 | 03+ | U[9](http://ref.x86asm.net/coder32.html#gen_note_x87_fpu_undoc_aliases) |  |  | p | *FSTP1* | **STi** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Store Floating Point Value and Pop |
|  |  | D9 |  | 4 |  |  |  |  |  | FLDENV | m14/28 |  |  |  |  |  | 0123 | 0123 |  |  | Load x87 FPU Environment |
|  |  | D9 | E0 | 4 |  |  |  |  |  | FCHS | ***ST*** |  |  |  |  |  | 0123 | .1.. | 0.23 |  | Change Sign |
|  |  | D9 | E1 | 4 |  |  |  |  |  | FABS | ***ST*** |  |  |  |  |  | 0123 | .1.. | 0.23 |  | Absolute Value |
|  |  | D9 | E4 | 4 |  |  |  |  |  | FTST | *ST* |  |  |  |  |  | 0123 | 0123 |  |  | Test |
|  |  | D9 | E5 | 4 |  |  |  |  |  | FXAM | *ST* |  |  |  |  |  | 0123 | 0123 |  |  | Examine |
|  |  | D9 |  | 5 |  |  |  |  |  | FLDCW | m16 |  |  |  |  |  | 0123 |  | 0123 |  | Load x87 FPU Control Word |
|  |  | D9 | E8 | 5 |  |  |  |  | s | FLD1 | ***ST*** |  |  |  |  |  | 0123 | .1.. | 0.23 |  | Load Constant +1.0 |
|  |  | D9 | E9 | 5 |  |  |  |  | s | FLDL2T | ***ST*** |  |  |  |  |  | 0123 | .1.. | 0.23 |  | Load Constant log210 |
|  |  | D9 | EA | 5 |  |  |  |  | s | FLDL2E | ***ST*** |  |  |  |  |  | 0123 | .1.. | 0.23 |  | Load Constant log2e |
|  |  | D9 | EB | 5 |  |  |  |  | s | FLDPI | ***ST*** |  |  |  |  |  | 0123 | .1.. | 0.23 |  | Load Constant π |
|  |  | D9 | EC | 5 |  |  |  |  | s | FLDLG2 | ***ST*** |  |  |  |  |  | 0123 | .1.. | 0.23 |  | Load Constant log102 |
|  |  | D9 | ED | 5 |  |  |  |  | s | FLDLN2 | ***ST*** |  |  |  |  |  | 0123 | .1.. | 0.23 |  | Load Constant loge2 |
|  |  | D9 | EE | 5 |  |  |  |  | s | FLDZ | ***ST*** |  |  |  |  |  | 0123 | .1.. | 0.23 |  | Load Constant +0.0 |
|  |  | D9 |  | 6 |  |  |  |  |  | FNSTENV | **m14/28** |  |  |  |  |  | 0123 |  | 0123 |  | Store x87 FPU Environment |
| 9B |  | D9 |  | 6 |  |  |  |  |  | FSTENV | **m14/28** |  |  |  |  |  | 0123 |  | 0123 |  | Store x87 FPU Environment |
|  |  | D9 | F0 | 6 |  |  |  |  |  | F2XM1 | ***ST*** |  |  |  |  |  | 0123 | .1.. | 0.23 |  | Compute 2x-1 |
|  |  | D9 | F1 | 6 |  |  |  |  | p | FYL2X | ***ST1*** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Compute y × log2x and Pop |
|  |  | D9 | F2 | 6 |  |  |  |  | s | FPTAN | ***ST*** |  |  |  |  |  | 0123 | .12. | 0..3 |  | Partial Tangent |
|  |  | D9 | F3 | 6 |  |  |  |  | p | FPATAN | ***ST1*** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Partial Arctangent and Pop |
|  |  | D9 | F4 | 6 |  |  |  |  | s | FXTRACT | ***ST*** |  |  |  |  |  | 0123 | .1.. | 0.23 |  | Extract Exponent and Significand |
|  |  | D9 | F5 | 6 |  |  |  |  |  | FPREM1 | ***ST*** | *ST1* |  |  |  |  | 0123 | 0123 |  |  | IEEE Partial Remainder |
|  |  | D9 | F6 | 6 |  |  |  |  |  | FDECSTP |  |  |  |  |  |  | 0123 | .1.. | 0.23 | .0.. | Decrement Stack-Top Pointer |
|  |  | D9 | F7 | 6 |  |  |  |  |  | FINCSTP |  |  |  |  |  |  | 0123 | .1.. | 0.23 | .0.. | Increment Stack-Top Pointer |
|  |  | D9 |  | 7 |  |  |  |  |  | FNSTCW | **m16** |  |  |  |  |  | 0123 |  | 0123 |  | Store x87 FPU Control Word |
| 9B |  | D9 |  | 7 |  |  |  |  |  | FSTCW | **m16** |  |  |  |  |  | 0123 |  | 0123 |  | Store x87 FPU Control Word |
|  |  | D9 | F8 | 7 |  |  |  |  |  | FPREM | ***ST*** | *ST1* |  |  |  |  | 0123 | 0123 |  |  | Partial Remainder (for compatibility with i8087 and i287) |
|  |  | D9 | F9 | 7 |  |  |  |  | p | FYL2XP1 | ***ST1*** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Compute y × log2(x+1) and Pop |
|  |  | D9 | FA | 7 |  |  |  |  |  | FSQRT | ***ST*** |  |  |  |  |  | 0123 | .1.. | 0.23 |  | Square Root |
|  |  | D9 | FB | 7 |  |  |  |  | s | FSINCOS | ***ST*** |  |  |  |  |  | 0123 | .12. | 0..3 |  | Sine and Cosine |
|  |  | D9 | FC | 7 |  |  |  |  |  | FRNDINT | ***ST*** |  |  |  |  |  | 0123 | .1.. | 0.23 |  | Round to Integer |
|  |  | D9 | FD | 7 |  |  |  |  |  | FSCALE | ***ST*** | *ST1* |  |  |  |  | 0123 | .1.. | 0.23 |  | Scale |
|  |  | D9 | FE | 7 |  |  |  |  |  | FSIN | ***ST*** |  |  |  |  |  | 0123 | .12. | 0..3 |  | Sine |
|  |  | D9 | FF | 7 |  |  |  |  |  | FCOS | ***ST*** |  |  |  |  |  | 0123 | .12. | 0..3 |  | Cosine |
|  |  | DA |  | 0 |  |  |  |  |  | FIADD | ***ST*** | m32int |  |  |  |  | 0123 | .1.. | 0.23 |  | Add |
|  |  | DA |  | 0 | PP+ |  |  |  |  | FCMOVB | **ST** | STi |  |  |  | .......c | 0123 | .1.. | 0.23 |  | FP Conditional Move - below (CF=1) |
|  |  | DA |  | 1 |  |  |  |  |  | FIMUL | ***ST*** | m32int |  |  |  |  | 0123 | .1.. | 0.23 |  | Multiply |
|  |  | DA |  | 1 | PP+ |  |  |  |  | FCMOVE | **ST** | STi |  |  |  | ....z... | 0123 | .1.. | 0.23 |  | FP Conditional Move - equal (ZF=1) |
|  |  | DA |  | 2 |  |  |  |  |  | FICOM | *ST* | m32int |  |  |  |  | 0123 | 0123 |  |  | Compare Integer |
|  |  | DA |  | 2 | PP+ |  |  |  |  | FCMOVBE | **ST** | STi |  |  |  | ....z... | 0123 | .1.. | 0.23 |  | FP Conditional Move - below or equal (CF=1 or ZF=1) |
|  |  | DA |  | 3 |  |  |  |  | p | FICOMP | *ST* | m32int |  |  |  |  | 0123 | 0123 |  |  | Compare Integer and Pop |
|  |  | DA |  | 3 | PP+ |  |  |  |  | FCMOVU | **ST** | STi |  |  |  | ......p. | 0123 | .1.. | 0.23 |  | FP Conditional Move - unordered (PF=1) |
|  |  | DA |  | 4 |  |  |  |  |  | FISUB | ***ST*** | m32int |  |  |  |  | 0123 | .1.. | 0.23 |  | Subtract |
|  |  | DA |  | 5 |  |  |  |  |  | FISUBR | ***ST*** | m32int |  |  |  |  | 0123 | .1.. | 0.23 |  | Reverse Subtract |
|  |  | DA | E9 | 5 | 03+ |  |  |  | P | FUCOMPP | *ST* | *ST1* |  |  |  |  | 0123 | 0123 |  |  | Unordered Compare Floating Point Values and Pop Twice |
|  |  | DA |  | 6 |  |  |  |  |  | FIDIV | ***ST*** | m32int |  |  |  |  | 0123 | .1.. | 0.23 |  | Divide |
|  |  | DA |  | 7 |  |  |  |  |  | FIDIVR | ***ST*** | m32int |  |  |  |  | 0123 | .1.. | 0.23 |  | Reverse Divide |
|  |  | DB |  | 0 |  |  |  |  | s | FILD | ***ST*** | m32int |  |  |  |  | 0123 | .1.. | 0.23 |  | Load Integer |
|  |  | DB |  | 0 | PP+ |  |  |  |  | FCMOVNB | **ST** | STi |  |  |  | .......c | 0123 | .1.. | 0.23 |  | FP Conditional Move - not below (CF=0) |
|  |  | DB |  | 1 | P4++ |  |  |  | p | FISTTP | **m32int** | *ST* |  |  | sse3 |  | 0123 | .1.. | 0.23 | .0.. | Store Integer with Truncation and Pop |
|  |  | DB |  | 1 | PP+ |  |  |  |  | FCMOVNE | **ST** | STi |  |  |  | ....z... | 0123 | .1.. | 0.23 |  | FP Conditional Move - not equal (ZF=0) |
|  |  | DB |  | 2 |  |  |  |  |  | FIST | **m32int** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Store Integer |
|  |  | DB |  | 2 | PP+ |  |  |  |  | FCMOVNBE | **ST** | STi |  |  |  | ....z... | 0123 | .1.. | 0.23 |  | FP Conditional Move - below or equal (CF=0 and ZF=0) |
|  |  | DB |  | 3 |  |  |  |  | p | FISTP | **m32int** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Store Integer and Pop |
|  |  | DB |  | 3 | PP+ |  |  |  |  | FCMOVNU | **ST** | STi |  |  |  | ......p. | 0123 | .1.. | 0.23 |  | FP Conditional Move - not unordered (PF=0) |
|  |  | DB | E0 | 4 | 01+ | D[6](http://ref.x86asm.net/coder32.html#gen_note_FNENI_DBE0_FNDISI_DBE1) |  |  |  | FNENI *nop* |  |  |  |  |  |  |  |  |  |  | Treated as Integer NOP |
|  |  | DB | E1 | 4 | 01+ | D[6](http://ref.x86asm.net/coder32.html#gen_note_FNENI_DBE0_FNDISI_DBE1) |  |  |  | FNDISI *nop* |  |  |  |  |  |  |  |  |  |  | Treated as Integer NOP |
|  |  | DB | E2 | 4 |  |  |  |  |  | FNCLEX |  |  |  |  |  |  | 0123 |  | 0123 |  | Clear Exceptions |
| 9B |  | DB | E2 | 4 |  |  |  |  |  | FCLEX |  |  |  |  |  |  | 0123 |  | 0123 |  | Clear Exceptions |
|  |  | DB | E3 | 4 |  |  |  |  |  | FNINIT |  |  |  |  |  |  | 0123 |  |  | 0000 | Initialize Floating-Point Unit |
| 9B |  | DB | E3 | 4 |  |  |  |  |  | FINIT |  |  |  |  |  |  | 0123 |  |  | 0000 | Initialize Floating-Point Unit |
|  |  | DB | E4 | 4 | 03+ | D[7](http://ref.x86asm.net/coder32.html#gen_note_FNSETPM_DBE4) |  |  |  | FNSETPM *nop* |  |  |  |  |  |  |  |  |  |  | Treated as Integer NOP |
|  |  | DB |  | 5 |  |  |  |  | s | FLD | ***ST*** | m80real |  |  |  |  | 0123 | .1.. | 0.23 |  | Load Floating Point Value |
|  |  | DB |  | 5 | PP+ |  |  |  |  | FUCOMI | ST | STi |  |  |  |  | o...z.pc .1.. | o...z.pc .1.. |  | o....... | Unordered Compare Floating Point Values and Set EFLAGS |
|  |  | DB |  | 6 | PP+ |  |  |  |  | FCOMI | ST | STi |  |  |  |  | o...z.pc .1.. | o...z.pc .1.. |  | o....... | Compare Floating Point Values and Set EFLAGS |
|  |  | DB |  | 7 |  |  |  |  | p | FSTP | **m80real** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Store Floating Point Value and Pop |
|  |  | DC |  | 0 |  |  |  |  |  | FADD | ***ST*** | m64real |  |  |  |  | 0123 | .1.. | 0.23 |  | Add |
|  |  | DC |  | 0 |  |  |  |  |  | FADD | **STi** | ST |  |  |  |  | 0123 | .1.. | 0.23 |  | Add |
|  |  | DC |  | 1 |  |  |  |  |  | FMUL | ***ST*** | m64real |  |  |  |  | 0123 | .1.. | 0.23 |  | Multiply |
|  |  | DC |  | 1 |  |  |  |  |  | FMUL | **STi** | ST |  |  |  |  | 0123 | .1.. | 0.23 |  | Multiply |
|  |  | DC |  | 2 |  |  |  |  |  | FCOM | *ST* | m64real |  |  |  |  | 0123 | 0123 |  |  | Compare Real |
|  |  | DC |  | 2 | 03+ | U[9](http://ref.x86asm.net/coder32.html#gen_note_x87_fpu_undoc_aliases) |  |  |  | *FCOM2* | *ST* | STi |  |  |  |  | 0123 | 0123 |  |  | Compare Real |
|  |  | DC |  | 3 |  |  |  |  | p | FCOMP | *ST* | m64real |  |  |  |  | 0123 | 0123 |  |  | Compare Real and Pop |
|  |  | DC |  | 3 | 03+ | U[9](http://ref.x86asm.net/coder32.html#gen_note_x87_fpu_undoc_aliases) |  |  | p | *FCOMP3* | *ST* | STi |  |  |  |  | 0123 | 0123 |  |  | Compare Real and Pop |
|  |  | DC |  | 4 |  |  |  |  |  | FSUB | ***ST*** | m64real |  |  |  |  | 0123 | .1.. | 0.23 |  | Subtract |
|  |  | DC |  | 4 |  |  |  |  |  | FSUBR | **STi** | ST |  |  |  |  | 0123 | .1.. | 0.23 |  | Reverse Subtract |
|  |  | DC |  | 5 |  |  |  |  |  | FSUBR | ***ST*** | m64real |  |  |  |  | 0123 | .1.. | 0.23 |  | Reverse Subtract |
|  |  | DC |  | 5 |  |  |  |  |  | FSUB | **STi** | ST |  |  |  |  | 0123 | .1.. | 0.23 |  | Subtract |
|  |  | DC |  | 6 |  |  |  |  |  | FDIV | ***ST*** | m64real |  |  |  |  | 0123 | .1.. | 0.23 |  | Divide |
|  |  | DC |  | 6 |  |  |  |  |  | FDIVR | **STi** | ST |  |  |  |  | 0123 | .1.. | 0.23 |  | Reverse Divide |
|  |  | DC |  | 7 |  |  |  |  |  | FDIVR | ***ST*** | m64real |  |  |  |  | 0123 | .1.. | 0.23 |  | Reverse Divide |
|  |  | DC |  | 7 |  |  |  |  |  | FDIV | **STi** | ST |  |  |  |  | 0123 | .1.. | 0.23 |  | Divide and Pop |
|  |  | DD |  | 0 |  |  |  |  | s | FLD | ***ST*** | m64real |  |  |  |  | 0123 | .1.. | 0.23 |  | Load Floating Point Value |
|  |  | DD |  | 0 |  |  |  |  |  | FFREE | STi |  |  |  |  |  | 0123 |  | 0123 |  | Free Floating-Point Register |
|  |  | DD |  | 1 | P4++ |  |  |  | p | FISTTP | **m64int** | *ST* |  |  | sse3 |  | 0123 | .1.. | 0.23 | .0.. | Store Integer with Truncation and Pop |
|  |  | DD |  | 1 | 03+ | U[9](http://ref.x86asm.net/coder32.html#gen_note_x87_fpu_undoc_aliases) |  |  |  | *FXCH4* | ***ST*** | **STi** |  |  |  |  | 0123 | .1.. | 0.23 |  | Exchange Register Contents |
|  |  | DD |  | 2 |  |  |  |  |  | FST | **m64real** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Store Floating Point Value |
|  |  | DD |  | 2 |  |  |  |  |  | FST | ***ST*** | STi |  |  |  |  | 0123 | .1.. | 0.23 |  | Store Floating Point Value |
|  |  | DD |  | 3 |  |  |  |  | p | FSTP | **m64real** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Store Floating Point Value and Pop |
|  |  | DD |  | 3 |  |  |  |  | p | FSTP | ***ST*** | STi |  |  |  |  | 0123 | .1.. | 0.23 |  | Store Floating Point Value and Pop |
|  |  | DD |  | 4 |  |  |  |  |  | FRSTOR | ***ST*** | ***ST1*** | ***ST2*** | ... |  |  | 0123 | 0123 |  |  | Restore x87 FPU State |
|  |  | DD |  | 4 | 03+ |  |  |  |  | FUCOM | *ST* | STi |  |  |  |  | 0123 | 0123 |  |  | Unordered Compare Floating Point Values |
|  |  | DD | E1 | 4 | 03+ |  |  |  |  | FUCOM | *ST* | *ST1* |  |  |  |  | 0123 | 0123 |  |  | Unordered Compare Floating Point Values |
|  |  | DD |  | 5 | 03+ |  |  |  | p | FUCOMP | *ST* | STi |  |  |  |  | 0123 | 0123 |  |  | Unordered Compare Floating Point Values and Pop |
|  |  | DD | E9 | 5 | 03+ |  |  |  | p | FUCOMP | *ST* | *ST1* |  |  |  |  | 0123 | 0123 |  |  | Unordered Compare Floating Point Values and Pop |
|  |  | DD |  | 6 |  |  |  |  |  | FNSAVE | **m94/108** | *ST* | *ST1* | ... |  |  | 0123 | 0123 |  | 0000 | Store x87 FPU State |
| 9B |  | DD |  | 6 |  |  |  |  |  | FSAVE | **m94/108** | *ST* | *ST1* | ... |  |  | 0123 | 0123 |  | 0000 | Store x87 FPU State |
|  |  | DD |  | 7 |  |  |  |  |  | FNSTSW | **m16** |  |  |  |  |  | 0123 |  | 0123 |  | Store x87 FPU Status Word |
| 9B |  | DD |  | 7 |  |  |  |  |  | FSTSW | **m16** |  |  |  |  |  | 0123 |  | 0123 |  | Store x87 FPU Status Word |
|  |  | DE |  | 0 |  |  |  |  |  | FIADD | ***ST*** | m16int |  |  |  |  | 0123 | .1.. | 0.23 |  | Add |
|  |  | DE |  | 0 |  |  |  |  | p | FADDP | **STi** | ST |  |  |  |  | 0123 | .1.. | 0.23 |  | Add and Pop |
|  |  | DE | C1 | 0 |  |  |  |  | p | FADDP | ***ST1*** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Add and Pop |
|  |  | DE |  | 1 |  |  |  |  |  | FIMUL | ***ST*** | m16int |  |  |  |  | 0123 | .1.. | 0.23 |  | Multiply |
|  |  | DE |  | 1 |  |  |  |  | p | FMULP | **STi** | ST |  |  |  |  | 0123 | .1.. | 0.23 |  | Multiply and Pop |
|  |  | DE | C9 | 1 |  |  |  |  | p | FMULP | ***ST1*** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Multiply and Pop |
|  |  | DE |  | 2 |  |  |  |  |  | FICOM | *ST* | m16int |  |  |  |  | 0123 | 0123 |  |  | Compare Integer |
|  |  | DE |  | 2 | 03+ | U[9](http://ref.x86asm.net/coder32.html#gen_note_x87_fpu_undoc_aliases) |  |  | p | *FCOMP5* | *ST* | STi |  |  |  |  | 0123 | 0123 |  |  | Compare Real and Pop |
|  |  | DE |  | 3 |  |  |  |  | p | FICOMP | *ST* | m16int |  |  |  |  | 0123 | 0123 |  |  | Compare Integer and Pop |
|  |  | DE | D9 | 3 |  |  |  |  | P | FCOMPP | *ST* | *ST1* |  |  |  |  | 0123 | 0123 |  |  | Compare Real and Pop Twice |
|  |  | DE |  | 4 |  |  |  |  |  | FISUB | ***ST*** | m16int |  |  |  |  | 0123 | .1.. | 0.23 |  | Subtract |
|  |  | DE |  | 4 |  |  |  |  | p | FSUBRP | **STi** | ST |  |  |  |  | 0123 | .1.. | 0.23 |  | Reverse Subtract and Pop |
|  |  | DE | E1 | 4 |  |  |  |  | p | FSUBRP | ***ST1*** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Reverse Subtract and Pop |
|  |  | DE |  | 5 |  |  |  |  |  | FISUBR | ***ST*** | m16int |  |  |  |  | 0123 | .1.. | 0.23 |  | Reverse Subtract |
|  |  | DE |  | 5 |  |  |  |  | p | FSUBP | **STi** | ST |  |  |  |  | 0123 | .1.. | 0.23 |  | Subtract and Pop |
|  |  | DE | E9 | 5 |  |  |  |  | p | FSUBP | ***ST1*** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Subtract and Pop |
|  |  | DE |  | 6 |  |  |  |  |  | FIDIV | ***ST*** | m16int |  |  |  |  | 0123 | .1.. | 0.23 |  | Divide |
|  |  | DE |  | 6 |  |  |  |  | p | FDIVRP | **STi** | ST |  |  |  |  | 0123 | .1.. | 0.23 |  | Reverse Divide and Pop |
|  |  | DE | F1 | 6 |  |  |  |  | p | FDIVRP | ***ST1*** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Reverse Divide and Pop |
|  |  | DE |  | 7 |  |  |  |  |  | FIDIVR | ***ST*** | m16int |  |  |  |  | 0123 | .1.. | 0.23 |  | Reverse Divide |
|  |  | DE |  | 7 |  |  |  |  | p | FDIVP | **STi** | ST |  |  |  |  | 0123 | .1.. | 0.23 |  | Divide and Pop |
|  |  | DE | F9 | 7 |  |  |  |  | p | FDIVP | ***ST1*** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Divide and Pop |
|  |  | DF |  | 0 |  |  |  |  | s | FILD | ***ST*** | m16int |  |  |  |  | 0123 | .1.. | 0.23 |  | Load Integer |
|  |  | DF |  | 0 |  | D[8](http://ref.x86asm.net/coder32.html#gen_note_FFREEP_DF_1) |  |  | p | *FFREEP* | STi |  |  |  |  |  | 0123 |  | 0123 |  | Free Floating-Point Register and Pop |
|  |  | DF |  | 1 | P4++ |  |  |  | p | FISTTP | **m16int** | *ST* |  |  | sse3 |  | 0123 | .1.. | 0.23 | .0.. | Store Integer with Truncation and Pop |
|  |  | DF |  | 1 | 03+ | U[9](http://ref.x86asm.net/coder32.html#gen_note_x87_fpu_undoc_aliases) |  |  |  | *FXCH7* | ***ST*** | **STi** |  |  |  |  | 0123 | .1.. | 0.23 |  | Exchange Register Contents |
|  |  | DF |  | 2 |  |  |  |  |  | FIST | **m16int** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Store Integer |
|  |  | DF |  | 2 | 03+ | U[9](http://ref.x86asm.net/coder32.html#gen_note_x87_fpu_undoc_aliases) |  |  | p | *FSTP8* | **STi** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Store Floating Point Value and Pop |
|  |  | DF |  | 3 |  |  |  |  | p | FISTP | **m16int** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Store Integer and Pop |
|  |  | DF |  | 3 | 03+ | U[9](http://ref.x86asm.net/coder32.html#gen_note_x87_fpu_undoc_aliases) |  |  | p | *FSTP9* | **STi** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Store Floating Point Value and Pop |
|  |  | DF |  | 4 |  |  |  |  | s | FBLD | ***ST*** | m80dec |  |  |  |  | 0123 | .1.. | 0.23 |  | Load Binary Coded Decimal |
|  |  | DF | E0 | 4 | 02+ |  |  |  |  | FNSTSW | **AX** |  |  |  |  |  | 0123 |  | 0123 |  | Store x87 FPU Status Word |
| 9B |  | DF | E0 | 4 | 02+ |  |  |  |  | FSTSW | **AX** |  |  |  |  |  | 0123 |  | 0123 |  | Store x87 FPU Status Word |
|  |  | DF |  | 5 |  |  |  |  | s | FILD | ***ST*** | m64int |  |  |  |  | 0123 | .1.. | 0.23 |  | Load Integer |
|  |  | DF |  | 5 | PP+ |  |  |  | p | FUCOMIP | ST | STi |  |  |  |  | o...z.pc .1.. | o...z.pc .1.. |  | o....... | Unordered Compare Floating Point Values and Set EFLAGS and Pop |
|  |  | DF |  | 6 |  |  |  |  | p | FBSTP | **m80dec** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Store BCD Integer and Pop |
|  |  | DF |  | 6 | PP+ |  |  |  | p | FCOMIP | ST | STi |  |  |  |  | o...z.pc .1.. | o...z.pc .1.. |  | o....... | Compare Floating Point Values and Set EFLAGS and Pop |
|  |  | DF |  | 7 |  |  |  |  | p | FISTP | **m64int** | *ST* |  |  |  |  | 0123 | .1.. | 0.23 |  | Store Integer and Pop |
|  |  | E0 |  |  |  |  |  |  |  | LOOPNZ | ***eCX*** | rel8 |  |  |  | ....z... |  |  |  |  | Decrement count; Jump short if count!=0 and ZF=0 |
| LOOPNE | ***eCX*** | rel8 |  |  |
|  |  | E1 |  |  |  |  |  |  |  | LOOPZ | ***eCX*** | rel8 |  |  |  | ....z... |  |  |  |  | Decrement count; Jump short if count!=0 and ZF=1 |
| LOOPE | ***eCX*** | rel8 |  |  |
|  |  | E2 |  |  |  |  |  |  |  | LOOP | ***eCX*** | rel8 |  |  |  |  |  |  |  |  | Decrement count; Jump short if count!=0 |
|  |  | E3 |  |  |  |  |  |  |  | JCXZ | rel8 | *CX* |  |  |  |  |  |  |  |  | Jump short if eCX register is 0 |
| JECXZ | rel8 | *ECX* |  |  |
|  |  | E4 |  |  |  |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | IN | **AL** | imm8 |  |  |  |  |  |  |  |  | Input from Port |
|  |  | E5 |  |  |  |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | IN | **eAX** | imm8 |  |  |  |  |  |  |  |  | Input from Port |
|  |  | E6 |  |  |  |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | OUT | **imm8** | AL |  |  |  |  |  |  |  |  | Output to Port |
|  |  | E7 |  |  |  |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | OUT | **imm8** | eAX |  |  |  |  |  |  |  |  | Output to Port |
|  |  | E8 |  |  |  |  |  |  |  | CALL | rel16/32 |  |  |  |  |  |  |  |  |  | Call Procedure |
|  |  | E9 |  |  |  |  |  |  |  | JMP | rel16/32 |  |  |  |  |  |  |  |  |  | Jump |
|  |  | EA |  |  |  |  |  |  |  | JMPF | ptr16:16/32 |  |  |  |  |  |  |  |  |  | Jump |
|  |  | EB |  |  |  |  |  |  |  | JMP | rel8 |  |  |  |  |  |  |  |  |  | Jump |
|  |  | EC |  |  |  |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | IN | **AL** | DX |  |  |  |  |  |  |  |  | Input from Port |
|  |  | ED |  |  |  |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | IN | **eAX** | DX |  |  |  |  |  |  |  |  | Input from Port |
|  |  | EE |  |  |  |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | OUT | **DX** | AL |  |  |  |  |  |  |  |  | Output to Port |
|  |  | EF |  |  |  |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | OUT | **DX** | eAX |  |  |  |  |  |  |  |  | Output to Port |
|  |  | F4 |  |  |  |  |  | 0 |  | HLT |  |  |  |  |  |  |  |  |  |  | Halt |
|  |  | F5 |  |  |  |  |  |  |  | CMC |  |  |  |  |  | .......c | .......c | .......c |  |  | Complement Carry Flag |
|  |  | F6 |  | 0 |  |  |  |  |  | TEST | r/m8 | imm8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Compare |
|  |  | F6 |  | 1 |  | U[12](http://ref.x86asm.net/coder32.html#gen_note_TEST_F6_1_F7_1) |  |  |  | TEST | r/m8 | imm8 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Compare |
|  |  | F6 |  | 2 |  |  |  |  |  | NOT | **r/m8** |  |  |  |  |  |  |  |  |  | One's Complement Negation |
|  |  | F6 |  | 3 |  |  |  |  |  | NEG | **r/m8** |  |  |  |  |  | o..szapc | o..szapc |  |  | Two's Complement Negation |
|  |  | F6 |  | 4 |  |  |  |  |  | MUL | ***AX*** | *AL* | r/m8 |  |  |  | o..szapc | o......c | ...szap. |  | Unsigned Multiply |
|  |  | F6 |  | 5 |  |  |  |  |  | IMUL | ***AX*** | *AL* | r/m8 |  |  |  | o..szapc | o......c | ...szap. |  | Signed Multiply |
|  |  | F6 |  | 6 |  |  |  |  |  | DIV | ***AL*** | ***AH*** | *AX* | r/m8 |  |  | o..szapc |  | o..szapc |  | Unsigned Divide |
|  |  | F6 |  | 7 |  |  |  |  |  | IDIV | ***AL*** | ***AH*** | *AX* | r/m8 |  |  | o..szapc |  | o..szapc |  | Signed Divide |
|  |  | F7 |  | 0 |  |  |  |  |  | TEST | r/m16/32 | imm16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Compare |
|  |  | F7 |  | 1 |  | U[12](http://ref.x86asm.net/coder32.html#gen_note_TEST_F6_1_F7_1) |  |  |  | TEST | r/m16/32 | imm16/32 |  |  |  |  | o..szapc | o..sz.pc | .....a.. | o......c | Logical Compare |
|  |  | F7 |  | 2 |  |  |  |  |  | NOT | **r/m16/32** |  |  |  |  |  |  |  |  |  | One's Complement Negation |
|  |  | F7 |  | 3 |  |  |  |  |  | NEG | **r/m16/32** |  |  |  |  |  | o..szapc | o..szapc |  |  | Two's Complement Negation |
|  |  | F7 |  | 4 |  |  |  |  |  | MUL | ***eDX*** | ***eAX*** | r/m16/32 |  |  |  | o..szapc | o......c | ...szap. |  | Unsigned Multiply |
|  |  | F7 |  | 5 |  |  |  |  |  | IMUL | ***eDX*** | ***eAX*** | r/m16/32 |  |  |  | o..szapc | o......c | ...szap. |  | Signed Multiply |
|  |  | F7 |  | 6 |  |  |  |  |  | DIV | ***eDX*** | ***eAX*** | r/m16/32 |  |  |  | o..szapc |  | o..szapc |  | Unsigned Divide |
|  |  | F7 |  | 7 |  |  |  |  |  | IDIV | ***eDX*** | ***eAX*** | r/m16/32 |  |  |  | o..szapc |  | o..szapc |  | Signed Divide |
|  |  | F8 |  |  |  |  |  |  |  | CLC |  |  |  |  |  |  | .......c | .......c |  | .......c | Clear Carry Flag |
|  |  | F9 |  |  |  |  |  |  |  | STC |  |  |  |  |  |  | .......c | .......c |  | .......C | Set Carry Flag |
|  |  | FA |  |  |  |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | CLI |  |  |  |  |  |  | ..i..... | ..i..... |  | ..i..... | Clear Interrupt Flag |
|  |  | FB |  |  |  |  |  | f[1](http://ref.x86asm.net/coder32.html#rflags_iopl) |  | STI |  |  |  |  |  |  | ..i..... | ..i..... |  | ..I..... | Set Interrupt Flag |
|  |  | FC |  |  |  |  |  |  |  | CLD |  |  |  |  |  |  | .d...... | .d...... |  | .d...... | Clear Direction Flag |
|  |  | FD |  |  |  |  |  |  |  | STD |  |  |  |  |  |  | .d...... | .d...... |  | .D...... | Set Direction Flag |
|  |  | FE |  | 0 |  |  |  |  |  | INC | **r/m8** |  |  |  |  |  | o..szap. | o..szap. |  |  | Increment by 1 |
|  |  | FE |  | 1 |  |  |  |  |  | DEC | **r/m8** |  |  |  |  |  | o..szap. | o..szap. |  |  | Decrement by 1 |
|  |  | FF |  | 0 |  |  |  |  |  | INC | **r/m16/32** |  |  |  |  |  | o..szap. | o..szap. |  |  | Increment by 1 |
|  |  | FF |  | 1 |  |  |  |  |  | DEC | **r/m16/32** |  |  |  |  |  | o..szap. | o..szap. |  |  | Decrement by 1 |
|  |  | FF |  | 2 |  |  |  |  |  | CALL | r/m16/32 |  |  |  |  |  |  |  |  |  | Call Procedure |
|  |  | FF |  | 3 |  |  |  |  |  | CALLF | m16:16/32 |  |  |  |  |  |  |  |  |  | Call Procedure |
|  |  | FF |  | 4 |  |  |  |  |  | JMP | r/m16/32 |  |  |  |  |  |  |  |  |  | Jump |
|  |  | FF |  | 5 |  |  |  |  |  | JMPF | m16:16/32 |  |  |  |  |  |  |  |  |  | Jump |
|  |  | FF |  | 6 |  |  |  |  |  | PUSH | r/m16/32 |  |  |  |  |  |  |  |  |  | Push Word, Doubleword or Quadword Onto the Stack |