**TUGAS 2**

**INSTRUCTION SET**

**x86**

|  |  |
| --- | --- |
| **Opcode** | **Description** |
| [AAA](http://x86.renejeschke.de/html/file_module_x86_id_1.html) | ASCII Adjust After Addition |
| [AAD](http://x86.renejeschke.de/html/file_module_x86_id_2.html) | ASCII Adjust AX Before Division |
| [AAS](http://x86.renejeschke.de/html/file_module_x86_id_3.html) | ASCII Adjust AL After Subtraction |
| [ADC](http://x86.renejeschke.de/html/file_module_x86_id_4.html) | Add with Carry |
| [ADD](http://x86.renejeschke.de/html/file_module_x86_id_5.html) | Add |
| [ADDPD](http://x86.renejeschke.de/html/file_module_x86_id_6.html) | Add Packed Double-Precision Floating-Point Values |
| [ADDPS](http://x86.renejeschke.de/html/file_module_x86_id_7.html) | Add Packed Single-Precision Floating-Point Values |
| [ADDSD](http://x86.renejeschke.de/html/file_module_x86_id_8.html) | Add Scalar Double-Precision Floating-Point Values |
| [ADDSS](http://x86.renejeschke.de/html/file_module_x86_id_9.html) | Add Scalar Single-Precision Floating-Point Values |
| [ADDSUBPD](http://x86.renejeschke.de/html/file_module_x86_id_10.html) | Packed Double-FP Add/Subtract |
| [ADDSUBPS](http://x86.renejeschke.de/html/file_module_x86_id_11.html) | Packed Single-FP Add/Subtract |
| [AND](http://x86.renejeschke.de/html/file_module_x86_id_12.html) | Logical AND |
| [ANDPD](http://x86.renejeschke.de/html/file_module_x86_id_13.html) | Bitwise Logical AND of Packed Double-Precision Floating-Point Values |
| [ANDPS](http://x86.renejeschke.de/html/file_module_x86_id_14.html) | Bitwise Logical AND of Packed Single-Precision Floating-Point Values |
| [ANDNPD](http://x86.renejeschke.de/html/file_module_x86_id_15.html) | Bitwise Logical AND NOT of Packed Double-Precision Floating-Point Values |
| [ANDNPS](http://x86.renejeschke.de/html/file_module_x86_id_16.html) | Bitwise Logical AND NOT of Packed Single-Precision Floating-Point Values |
| [ARPL](http://x86.renejeschke.de/html/file_module_x86_id_17.html) | Adjust RPL Field of Segment Selector |
| [BOUND](http://x86.renejeschke.de/html/file_module_x86_id_18.html) | Check Array Index Against Bounds |
| [BSF](http://x86.renejeschke.de/html/file_module_x86_id_19.html) | Bit Scan Forward |
| [BSR](http://x86.renejeschke.de/html/file_module_x86_id_20.html) | Bit Scan Reverse |
| [BSWAP](http://x86.renejeschke.de/html/file_module_x86_id_21.html) | Byte Swap |
| [BT](http://x86.renejeschke.de/html/file_module_x86_id_22.html) | Bit Test |
| [BTC](http://x86.renejeschke.de/html/file_module_x86_id_23.html) | Bit Test and Complement |
| [BTR](http://x86.renejeschke.de/html/file_module_x86_id_24.html) | Bit Test and Reset |
| [BTS](http://x86.renejeschke.de/html/file_module_x86_id_25.html) | Bit Test and Set |
| [CALL](http://x86.renejeschke.de/html/file_module_x86_id_26.html) | Call Procedure |
| [CBW/CWDE](http://x86.renejeschke.de/html/file_module_x86_id_27.html) | Convert Byte to Word/Convert Word to Doubleword |
| [CLC](http://x86.renejeschke.de/html/file_module_x86_id_28.html) | Clear Carry Flag |
| [CLD](http://x86.renejeschke.de/html/file_module_x86_id_29.html) | Clear Direction Flag |
| [CLFLUSH](http://x86.renejeschke.de/html/file_module_x86_id_30.html) | Flush Cache Line |
| [CLI](http://x86.renejeschke.de/html/file_module_x86_id_31.html) | Clear Interrupt Flag |
| [CLTS](http://x86.renejeschke.de/html/file_module_x86_id_32.html) | Clear Task-Switched Flag in CR0 |
| [CMC](http://x86.renejeschke.de/html/file_module_x86_id_33.html) | Complement Carry Flag |
| [CMOVcc](http://x86.renejeschke.de/html/file_module_x86_id_34.html) | Conditional Move |
| [CMP](http://x86.renejeschke.de/html/file_module_x86_id_35.html) | Compare Two Operands |
| [CMPPD](http://x86.renejeschke.de/html/file_module_x86_id_36.html) | Compare Packed Double-Precision Floating-Point Values |
| [CMPPS](http://x86.renejeschke.de/html/file_module_x86_id_37.html) | Compare Packed Single-Precision Floating-Point Values |
| [CMPS/CMPSB/CMPSW/CMPSD](http://x86.renejeschke.de/html/file_module_x86_id_38.html) | Compare String Operands |
| [CMPSD](http://x86.renejeschke.de/html/file_module_x86_id_39.html) | Compare Scalar Double-Precision Floating-Point Values |
| [CMPSS](http://x86.renejeschke.de/html/file_module_x86_id_40.html) | Compare Scalar Single-Precision Floating-Point Values |
| [CMPXCHG](http://x86.renejeschke.de/html/file_module_x86_id_41.html) | Compare and Exchange |
| [CMPXCHG8B](http://x86.renejeschke.de/html/file_module_x86_id_42.html) | Compare and Exchange 8 Bytes |
| [COMISD](http://x86.renejeschke.de/html/file_module_x86_id_43.html) | Compare Scalar Ordered Double-Precision Floating- Point Values and Set EFLAGS |
| [COMISS](http://x86.renejeschke.de/html/file_module_x86_id_44.html) | Compare Scalar Ordered Single-Precision Floating- Point Values and Set EFLAGS |
| [CPUID](http://x86.renejeschke.de/html/file_module_x86_id_45.html) | CPU Identification |
| [CVTDQ2PD](http://x86.renejeschke.de/html/file_module_x86_id_46.html) | Convert Packed Doubleword Integers to Packed Double-Precision Floating-Point Values |
| [CVTDQ2PS](http://x86.renejeschke.de/html/file_module_x86_id_47.html) | Convert Packed Doubleword Integers to Packed Single-Precision Floating-Point Values |
| [CVTPD2DQ](http://x86.renejeschke.de/html/file_module_x86_id_48.html) | Convert Packed Double-Precision Floating-Point Values to Packed Doubleword Integers |
| [CVTPD2PI](http://x86.renejeschke.de/html/file_module_x86_id_49.html) | Convert Packed Double-Precision Floating-Point Values to Packed Doubleword Integers |
| [CVTPD2PS](http://x86.renejeschke.de/html/file_module_x86_id_50.html) | Convert Packed Double-Precision Floating-Point Values to Packed Single-Precision Floating-Point Values |
| [CVTPI2PD](http://x86.renejeschke.de/html/file_module_x86_id_51.html) | Convert Packed Doubleword Integers to Packed Double-Precision Floating-Point Values |
| [CVTPI2PS](http://x86.renejeschke.de/html/file_module_x86_id_52.html) | Convert Packed Doubleword Integers to Packed Single-Precision Floating-Point Values |
| [CVTPS2DQ](http://x86.renejeschke.de/html/file_module_x86_id_53.html) | Convert Packed Single-Precision Floating-Point Values to Packed Doubleword Integers |
| [CVTPS2PD](http://x86.renejeschke.de/html/file_module_x86_id_54.html) | Convert Packed Single-Precision Floating-Point Values to Packed Double-Precision Floating-Point Values |
| [CVTPS2PI](http://x86.renejeschke.de/html/file_module_x86_id_55.html) | Convert Packed Single-Precision Floating-Point Values to Packed Doubleword Integers |
| [CVTSD2SI](http://x86.renejeschke.de/html/file_module_x86_id_56.html) | Convert Scalar Double-Precision Floating-Point Value to Doubleword Integer |
| [CVTSD2SS](http://x86.renejeschke.de/html/file_module_x86_id_57.html) | Convert Scalar Double-Precision Floating-Point Value to Scalar Single-Precision Floating-Point Value |
| [CVTSI2SD](http://x86.renejeschke.de/html/file_module_x86_id_58.html) | Convert Doubleword Integer to Scalar Double- Precision Floating-Point Value |
| [CVTSI2SS](http://x86.renejeschke.de/html/file_module_x86_id_59.html) | Convert Doubleword Integer to Scalar Single- Precision Floating-Point Value |
| [CVTSS2SD](http://x86.renejeschke.de/html/file_module_x86_id_60.html) | Convert Scalar Single-Precision Floating-Point Value to Scalar Double-Precision Floating-Point Value |
| [CVTSS2SI](http://x86.renejeschke.de/html/file_module_x86_id_61.html) | Convert Scalar Single-Precision Floating-Point Value to Doubleword Integer |
| [CVTTPD2PI](http://x86.renejeschke.de/html/file_module_x86_id_62.html) | Convert with Truncation Packed Double-Precision Floating-Point Values to Packed Doubleword Integers |
| [CVTTPD2DQ](http://x86.renejeschke.de/html/file_module_x86_id_63.html) | Convert with Truncation Packed Double-Precision Floating-Point Values to Packed Doubleword Integers |
| [CVTTPS2DQ](http://x86.renejeschke.de/html/file_module_x86_id_64.html) | Convert with Truncation Packed Single-Precision Floating-Point Values to Packed Doubleword Integers |
| [CVTTPS2PI](http://x86.renejeschke.de/html/file_module_x86_id_65.html) | Convert with Truncation Packed Single-Precision Floating-Point Values to Packed Doubleword Integers |
| [CVTTSD2SI](http://x86.renejeschke.de/html/file_module_x86_id_66.html) | Convert with Truncation Scalar Double-Precision Floating-Point Value to Signed Doubleword Integer |
| [CVTTSS2SI](http://x86.renejeschke.de/html/file_module_x86_id_67.html) | Convert with Truncation Scalar Single-Precision Floating-Point Value to Doubleword Integer |
| [CWD/CDQ](http://x86.renejeschke.de/html/file_module_x86_id_68.html) | Convert Word to Doubleword/Convert Doubleword to Quadword |
| [DAA](http://x86.renejeschke.de/html/file_module_x86_id_69.html) | Decimal Adjust AL after Addition |
| [DAS](http://x86.renejeschke.de/html/file_module_x86_id_70.html) | Decimal Adjust AL after Subtraction |
| [DEC](http://x86.renejeschke.de/html/file_module_x86_id_71.html) | Decrement by 1 |
| [DIV](http://x86.renejeschke.de/html/file_module_x86_id_72.html) | Unsigned Divide |
| [DIVPD](http://x86.renejeschke.de/html/file_module_x86_id_73.html) | Divide Packed Double-Precision Floating-Point Values |
| [DIVPS](http://x86.renejeschke.de/html/file_module_x86_id_74.html) | Divide Packed Single-Precision Floating-Point Values |
| [DIVSD](http://x86.renejeschke.de/html/file_module_x86_id_75.html) | Divide Scalar Double-Precision Floating-Point Values |
| [DIVSS](http://x86.renejeschke.de/html/file_module_x86_id_76.html) | Divide Scalar Single-Precision Floating-Point Values |
| [EMMS](http://x86.renejeschke.de/html/file_module_x86_id_77.html) | Empty MMX Technology State |
| [ENTER](http://x86.renejeschke.de/html/file_module_x86_id_78.html) | Make Stack Frame for Procedure Parameters |
| [F2XM1](http://x86.renejeschke.de/html/file_module_x86_id_79.html) | Compute 2x-1 |
| [FABS](http://x86.renejeschke.de/html/file_module_x86_id_80.html) | Absolute Value |
| [FADD/FADDP/FIADD](http://x86.renejeschke.de/html/file_module_x86_id_81.html) | Add |
| [FBLD](http://x86.renejeschke.de/html/file_module_x86_id_82.html) | Load Binary Coded Decimal |
| [FBSTP](http://x86.renejeschke.de/html/file_module_x86_id_83.html) | Store BCD Integer and Pop |
| [FCHS](http://x86.renejeschke.de/html/file_module_x86_id_84.html) | Change Sign |
| [FCLEX/FNCLEX](http://x86.renejeschke.de/html/file_module_x86_id_85.html) | Clear Exceptions |
| [FCMOVcc](http://x86.renejeschke.de/html/file_module_x86_id_86.html) | Floating-Point Conditional Move |
| [FCOM/FCOMP/FCOMPP](http://x86.renejeschke.de/html/file_module_x86_id_87.html) | Compare Floating Point Values |
| [FCOMI/FCOMIP/FUCOMI/FUCOMIP](http://x86.renejeschke.de/html/file_module_x86_id_88.html) | Compare Floating Point Values and Set EFLAGS |
| [FCOS](http://x86.renejeschke.de/html/file_module_x86_id_89.html) | Cosine |
| [FDECSTP](http://x86.renejeschke.de/html/file_module_x86_id_90.html) | Decrement Stack-Top Pointer |
| [FDIV/FDIVP/FIDIV](http://x86.renejeschke.de/html/file_module_x86_id_91.html) | Divide |
| [FDIVR/FDIVRP/FIDIVR](http://x86.renejeschke.de/html/file_module_x86_id_92.html) | Reverse Divide |
| [FFREE](http://x86.renejeschke.de/html/file_module_x86_id_93.html) | Free Floating-Point Register |
| [FICOM/FICOMP](http://x86.renejeschke.de/html/file_module_x86_id_94.html) | Compare Integer |
| [FILD](http://x86.renejeschke.de/html/file_module_x86_id_95.html) | Load Integer |
| [FINCSTP](http://x86.renejeschke.de/html/file_module_x86_id_96.html) | Increment Stack-Top Pointer |
| [FINIT/FNINIT](http://x86.renejeschke.de/html/file_module_x86_id_97.html) | Initialize Floating-Point Unit |
| [FIST/FISTP](http://x86.renejeschke.de/html/file_module_x86_id_98.html) | Store Integer |
| [FISTTP](http://x86.renejeschke.de/html/file_module_x86_id_99.html) | Store Integer with Truncation |
| [FLD](http://x86.renejeschke.de/html/file_module_x86_id_100.html) | Load Floating Point Value |
| [FLD1/FLDL2T/FLDL2E/FLDPI/FLDLG2/FLDLN2/FLDZ](http://x86.renejeschke.de/html/file_module_x86_id_101.html) | Load Constant |
| [FLDCW](http://x86.renejeschke.de/html/file_module_x86_id_102.html) | Load x87 FPU Control Word |
| [FLDENV](http://x86.renejeschke.de/html/file_module_x86_id_103.html) | Load x87 FPU Environment |
| [FMUL/FMULP/FIMUL](http://x86.renejeschke.de/html/file_module_x86_id_104.html) | Multiply |
| [FNOP](http://x86.renejeschke.de/html/file_module_x86_id_105.html) | No operation |
| [FPATAN](http://x86.renejeschke.de/html/file_module_x86_id_106.html) | Partial Arctangent |
| [FPREM](http://x86.renejeschke.de/html/file_module_x86_id_107.html) | Partial Remainder |
| [FPREM1](http://x86.renejeschke.de/html/file_module_x86_id_108.html) | Partial Remainder |
| [FPTAN](http://x86.renejeschke.de/html/file_module_x86_id_109.html) | Partial Tangent |
| [FRNDINT](http://x86.renejeschke.de/html/file_module_x86_id_110.html) | Round to Integer |
| [FRSTOR](http://x86.renejeschke.de/html/file_module_x86_id_111.html) | Restore x87 FPU State |
| [FSAVE/FNSAVE](http://x86.renejeschke.de/html/file_module_x86_id_112.html) | Store x87 FPU State |
| [FSCALE](http://x86.renejeschke.de/html/file_module_x86_id_113.html) | Scale |
| [FSIN](http://x86.renejeschke.de/html/file_module_x86_id_114.html) | Sine |
| [FSINCOS](http://x86.renejeschke.de/html/file_module_x86_id_115.html) | Sine and Cosine |
| [FSQRT](http://x86.renejeschke.de/html/file_module_x86_id_116.html) | Square Root |
| [FST/FSTP](http://x86.renejeschke.de/html/file_module_x86_id_117.html) | Store Floating Point Value |
| [FSTCW/FNSTCW](http://x86.renejeschke.de/html/file_module_x86_id_118.html) | Store x87 FPU Control Word |
| [FSTENV/FNSTENV](http://x86.renejeschke.de/html/file_module_x86_id_119.html) | Store x87 FPU Environment |
| [FSTSW/FNSTSW](http://x86.renejeschke.de/html/file_module_x86_id_120.html) | Store x87 FPU Status Word |
| [FSUB/FSUBP/FISUB](http://x86.renejeschke.de/html/file_module_x86_id_121.html) | Subtract |
| [FSUBR/FSUBRP/FISUBR](http://x86.renejeschke.de/html/file_module_x86_id_122.html) | Reverse Subtract |
| [FTST](http://x86.renejeschke.de/html/file_module_x86_id_123.html) | Test Floating Point Value |
| [FUCOM/FUCOMP/FUCOMPP](http://x86.renejeschke.de/html/file_module_x86_id_124.html) | Unordered Compare Floating Point Values |
| [FXAM](http://x86.renejeschke.de/html/file_module_x86_id_125.html) | Examine Floating Point Value |
| [FXCH](http://x86.renejeschke.de/html/file_module_x86_id_126.html) | Exchange Register Contents |
| [FXRSTOR](http://x86.renejeschke.de/html/file_module_x86_id_127.html) | Restore x87 FPU, MMX Technology, SSE, and SSE2 State |
| [FXSAVE](http://x86.renejeschke.de/html/file_module_x86_id_128.html) | Save x87 FPU, MMX Technology, SSE, and SSE2 State |
| [FXTRACT](http://x86.renejeschke.de/html/file_module_x86_id_129.html) | Extract Exponent and Mantissa |
| [FYL2X](http://x86.renejeschke.de/html/file_module_x86_id_130.html) | Compute y \* log\_2(x) |
| [FYL2XP1](http://x86.renejeschke.de/html/file_module_x86_id_131.html) | Compute y \* log\_2(x + 1) |
| [HADDPD](http://x86.renejeschke.de/html/file_module_x86_id_132.html) | Packed Double-FP Horizontal Add |
| [HADDPS](http://x86.renejeschke.de/html/file_module_x86_id_133.html) | Packed Single-FP Horizontal Add |
| [HLT](http://x86.renejeschke.de/html/file_module_x86_id_134.html) | Halt |
| [HSUBPD](http://x86.renejeschke.de/html/file_module_x86_id_135.html) | Packed Double-FP Horizontal Subtract |
| [HSUBPS](http://x86.renejeschke.de/html/file_module_x86_id_136.html) | Packed Single-FP Horizontal Subtract |
| [IDIV](http://x86.renejeschke.de/html/file_module_x86_id_137.html) | Signed Divide |
| [IMUL](http://x86.renejeschke.de/html/file_module_x86_id_138.html) | Signed Multiply |
| [IN](http://x86.renejeschke.de/html/file_module_x86_id_139.html) | Input from Port |
| [INC](http://x86.renejeschke.de/html/file_module_x86_id_140.html) | Increment by 1 |
| [INS/INSB/INSW/INSD](http://x86.renejeschke.de/html/file_module_x86_id_141.html) | Input from Port to String |
| [INT n/INTO/INT 3](http://x86.renejeschke.de/html/file_module_x86_id_142.html) | Call to Interrupt Procedure |
| [INVD](http://x86.renejeschke.de/html/file_module_x86_id_143.html) | Invalidate Internal Caches |
| [INVLPG](http://x86.renejeschke.de/html/file_module_x86_id_144.html) | Invalidate TLB Entry |
| [IRET/IRETD](http://x86.renejeschke.de/html/file_module_x86_id_145.html) | Interrupt Return |
| [Jcc](http://x86.renejeschke.de/html/file_module_x86_id_146.html) | Jump if Condition Is Met |
| [JMP](http://x86.renejeschke.de/html/file_module_x86_id_147.html) | Jump |
| [LAHF](http://x86.renejeschke.de/html/file_module_x86_id_148.html) | Load Status Flags into AH Register |
| [LAR](http://x86.renejeschke.de/html/file_module_x86_id_149.html) | Load Access Rights Byte |
| [LDDQU](http://x86.renejeschke.de/html/file_module_x86_id_150.html) | Load Unaligned Integer 128 Bits |
| [LDMXCSR](http://x86.renejeschke.de/html/file_module_x86_id_151.html) | Load MXCSR Register |
| [LDS/LES/LFS/LGS/LSS](http://x86.renejeschke.de/html/file_module_x86_id_152.html) | Load Far Pointer |
| [LEA](http://x86.renejeschke.de/html/file_module_x86_id_153.html) | Load Effective Address |
| [LEAVE](http://x86.renejeschke.de/html/file_module_x86_id_154.html) | High Level Procedure Exit |
| [LFENCE](http://x86.renejeschke.de/html/file_module_x86_id_155.html) | Load Fence |
| [LGDT/LIDT](http://x86.renejeschke.de/html/file_module_x86_id_156.html) | Load Global/Interrupt Descriptor Table Register |
| [LLDT](http://x86.renejeschke.de/html/file_module_x86_id_157.html) | Load Local Descriptor Table Register |
| [LMSW](http://x86.renejeschke.de/html/file_module_x86_id_158.html) | Load Machine Status Word |
| [LOCK](http://x86.renejeschke.de/html/file_module_x86_id_159.html) | Assert LOCK# Signal Prefix |
| [LODS/LODSB/LODSW/LODSD](http://x86.renejeschke.de/html/file_module_x86_id_160.html) | Load String |
| [LOOP/LOOPcc](http://x86.renejeschke.de/html/file_module_x86_id_161.html) | Loop According to ECX Counter |
| [LSL](http://x86.renejeschke.de/html/file_module_x86_id_162.html) | Load Segment Limit |
| [LTR](http://x86.renejeschke.de/html/file_module_x86_id_163.html) | Load Task Register |
| [MASKMOVDQU](http://x86.renejeschke.de/html/file_module_x86_id_164.html) | Store Selected Bytes of Double Quadword |
| [MASKMOVQ](http://x86.renejeschke.de/html/file_module_x86_id_165.html) | Store Selected Bytes of Quadword |
| [MAXPD](http://x86.renejeschke.de/html/file_module_x86_id_166.html) | Return Maximum Packed Double-Precision Floating- Point Values |
| [MAXPS](http://x86.renejeschke.de/html/file_module_x86_id_167.html) | Return Maximum Packed Single-Precision Floating-Point Values |
| [MAXSD](http://x86.renejeschke.de/html/file_module_x86_id_168.html) | Return Maximum Scalar Double-Precision Floating-Point Value |
| [MAXSS](http://x86.renejeschke.de/html/file_module_x86_id_169.html) | Return Maximum Scalar Single-Precision Floating-Point Value |
| [MFENCE](http://x86.renejeschke.de/html/file_module_x86_id_170.html) | Memory Fence |
| [MINPD](http://x86.renejeschke.de/html/file_module_x86_id_171.html) | Return Minimum Packed Double-Precision Floating-Point Values |
| [MINPS](http://x86.renejeschke.de/html/file_module_x86_id_172.html) | Return Minimum Packed Single-Precision Floating-Point Values |
| [MINSD](http://x86.renejeschke.de/html/file_module_x86_id_173.html) | Return Minimum Scalar Double-Precision Floating-Point Value |
| [MINSS](http://x86.renejeschke.de/html/file_module_x86_id_174.html) | Return Minimum Scalar Single-Precision Floating-Point Value |
| [MONITOR](http://x86.renejeschke.de/html/file_module_x86_id_175.html) | Setup Monitor Address |
| [MOV](http://x86.renejeschke.de/html/file_module_x86_id_176.html) | Move |
| [MOV](http://x86.renejeschke.de/html/file_module_x86_id_177.html) | Move to/from Control Registers |
| [MOV](http://x86.renejeschke.de/html/file_module_x86_id_178.html) | Move to/from Debug Registers |
| [MOVAPD](http://x86.renejeschke.de/html/file_module_x86_id_179.html) | Move Aligned Packed Double-Precision Floating-Point Values |
| [MOVAPS](http://x86.renejeschke.de/html/file_module_x86_id_180.html) | Move Aligned Packed Single-Precision Floating-Point Values |
| [MOVD](http://x86.renejeschke.de/html/file_module_x86_id_181.html) | Move Doubleword |
| [MOVDDUP](http://x86.renejeschke.de/html/file_module_x86_id_182.html) | Move One Double-FP and Duplicate |
| [MOVDQA](http://x86.renejeschke.de/html/file_module_x86_id_183.html) | Move Aligned Double Quadword |
| [MOVDQU](http://x86.renejeschke.de/html/file_module_x86_id_184.html) | Move Unaligned Double Quadword |
| [MOVDQ2Q](http://x86.renejeschke.de/html/file_module_x86_id_185.html) | Move Quadword from XMM to MMX Technology Register |
| [MOVHLPS](http://x86.renejeschke.de/html/file_module_x86_id_186.html) | Move Packed Single-Precision Floating-Point Values High to Low |
| [MOVHPD](http://x86.renejeschke.de/html/file_module_x86_id_187.html) | Move High Packed Double-Precision Floating-Point Value |
| [MOVHPS](http://x86.renejeschke.de/html/file_module_x86_id_188.html) | Move High Packed Single-Precision Floating-Point Values |
| [MOVLHPS](http://x86.renejeschke.de/html/file_module_x86_id_189.html) | Move Packed Single-Precision Floating-Point Values Low to High |
| [MOVLPD](http://x86.renejeschke.de/html/file_module_x86_id_190.html) | Move Low Packed Double-Precision Floating-Point Value |
| [MOVLPS](http://x86.renejeschke.de/html/file_module_x86_id_191.html) | Move Low Packed Single-Precision Floating-Point Values |
| [MOVMSKPD](http://x86.renejeschke.de/html/file_module_x86_id_192.html) | Extract Packed Double-Precision Floating-Point Sign Mask |
| [MOVMSKPS](http://x86.renejeschke.de/html/file_module_x86_id_193.html) | Extract Packed Single-Precision Floating-Point Sign Mask |
| [MOVNTDQ](http://x86.renejeschke.de/html/file_module_x86_id_194.html) | Store Double Quadword Using Non-Temporal Hint |
| [MOVNTI](http://x86.renejeschke.de/html/file_module_x86_id_195.html) | Store Doubleword Using Non-Temporal Hint |
| [MOVNTPD](http://x86.renejeschke.de/html/file_module_x86_id_196.html) | Store Packed Double-Precision Floating-Point Values Using Non-Temporal Hint |
| [MOVNTPS](http://x86.renejeschke.de/html/file_module_x86_id_197.html) | Store Packed Single-Precision Floating-Point Values Using Non-Temporal Hint |
| [MOVNTQ](http://x86.renejeschke.de/html/file_module_x86_id_198.html) | Store of Quadword Using Non-Temporal Hint |
| [MOVSHDUP](http://x86.renejeschke.de/html/file_module_x86_id_199.html) | Move Packed Single-FP High and Duplicate |
| [MOVSLDUP](http://x86.renejeschke.de/html/file_module_x86_id_200.html) | Move Packed Single-FP Low and Duplicate |
| [MOVQ](http://x86.renejeschke.de/html/file_module_x86_id_201.html) | Move Quadword |
| [MOVQ2DQ](http://x86.renejeschke.de/html/file_module_x86_id_202.html) | Move Quadword from MMX Technology to XMM Register |
| [MOVS/MOVSB/MOVSW/MOVSD](http://x86.renejeschke.de/html/file_module_x86_id_203.html) | Move Data from String to String |
| [MOVSD](http://x86.renejeschke.de/html/file_module_x86_id_204.html) | Move Scalar Double-Precision Floating-Point Value |
| [MOVSS](http://x86.renejeschke.de/html/file_module_x86_id_205.html) | Move Scalar Single-Precision Floating-Point Values |
| [MOVSX](http://x86.renejeschke.de/html/file_module_x86_id_206.html) | Move with Sign-Extension |
| [MOVUPD](http://x86.renejeschke.de/html/file_module_x86_id_207.html) | Move Unaligned Packed Double-Precision Floating- Point Values |
| [MOVUPS](http://x86.renejeschke.de/html/file_module_x86_id_208.html) | Move Unaligned Packed Single-Precision Floating- Point Values |
| [MOVZX](http://x86.renejeschke.de/html/file_module_x86_id_209.html) | Move with Zero-Extend |
| [MUL](http://x86.renejeschke.de/html/file_module_x86_id_210.html) | Unsigned Multiply |
| [MULPD](http://x86.renejeschke.de/html/file_module_x86_id_211.html) | Multiply Packed Double-Precision Floating-Point Values |
| [MULPS](http://x86.renejeschke.de/html/file_module_x86_id_212.html) | Multiply Packed Single-Precision Floating-Point Values |
| [MULSD](http://x86.renejeschke.de/html/file_module_x86_id_213.html) | Multiply Scalar Double-Precision Floating-Point Values |
| [MULSS](http://x86.renejeschke.de/html/file_module_x86_id_214.html) | Multiply Scalar Single-Precision Floating-Point Values |
| [MWAIT](http://x86.renejeschke.de/html/file_module_x86_id_215.html) | Monitor Wait |
| [NEG](http://x86.renejeschke.de/html/file_module_x86_id_216.html) | Two's Complement Negation |
| [NOP](http://x86.renejeschke.de/html/file_module_x86_id_217.html) | No Operation |
| [NOT](http://x86.renejeschke.de/html/file_module_x86_id_218.html) | One's Complement Negation |
| [OR](http://x86.renejeschke.de/html/file_module_x86_id_219.html) | Logical Inclusive OR |
| [ORPD](http://x86.renejeschke.de/html/file_module_x86_id_220.html) | Bitwise Logical OR of Double-Precision Floating-Point Values |
| [ORPS](http://x86.renejeschke.de/html/file_module_x86_id_221.html) | Bitwise Logical OR of Single-Precision Floating-Point Values |
| [OUT](http://x86.renejeschke.de/html/file_module_x86_id_222.html) | Output to Port |
| [OUTS/OUTSB/OUTSW/OUTSD](http://x86.renejeschke.de/html/file_module_x86_id_223.html) | Output String to Port |
| [PACKSSWB/PACKSSDW](http://x86.renejeschke.de/html/file_module_x86_id_224.html) | Pack with Signed Saturation |
| [PACKUSWB](http://x86.renejeschke.de/html/file_module_x86_id_225.html) | Pack with Unsigned Saturation |
| [PADDB/PADDW/PADDD](http://x86.renejeschke.de/html/file_module_x86_id_226.html) | Add Packed Integers |
| [PADDQ](http://x86.renejeschke.de/html/file_module_x86_id_227.html) | Add Packed Quadword Integers |
| [PADDSB/PADDSW](http://x86.renejeschke.de/html/file_module_x86_id_228.html) | Add Packed Signed Integers with Signed Saturation |
| [PADDUSB/PADDUSW](http://x86.renejeschke.de/html/file_module_x86_id_229.html) | Add Packed Unsigned Integers with Unsigned Saturation |
| [PAND](http://x86.renejeschke.de/html/file_module_x86_id_230.html) | Logical AND |
| [PANDN](http://x86.renejeschke.de/html/file_module_x86_id_231.html) | Logical AND NOT |
| [PAUSE](http://x86.renejeschke.de/html/file_module_x86_id_232.html) | Spin Loop Hint |
| [PAVGB/PAVGW](http://x86.renejeschke.de/html/file_module_x86_id_233.html) | Average Packed Integers |
| [PCMPEQB/PCMPEQW/PCMPEQD](http://x86.renejeschke.de/html/file_module_x86_id_234.html) | Compare Packed Data for Equal |
| [PCMPGTB/PCMPGTW/PCMPGTD](http://x86.renejeschke.de/html/file_module_x86_id_235.html) | Compare Packed Signed Integers for Greater Than |
| [PEXTRW](http://x86.renejeschke.de/html/file_module_x86_id_236.html) | Extract Word |
| [PINSRW](http://x86.renejeschke.de/html/file_module_x86_id_237.html) | Insert Word |
| [PMADDWD](http://x86.renejeschke.de/html/file_module_x86_id_238.html) | Multiply and Add Packed Integers |
| [PMAXSW](http://x86.renejeschke.de/html/file_module_x86_id_239.html) | Maximum of Packed Signed Word Integers |
| [PMAXUB](http://x86.renejeschke.de/html/file_module_x86_id_240.html) | Maximum of Packed Unsigned Byte Integers |
| [PMINSW](http://x86.renejeschke.de/html/file_module_x86_id_241.html) | Minimum of Packed Signed Word Integers |
| [PMINUB](http://x86.renejeschke.de/html/file_module_x86_id_242.html) | Minimum of Packed Unsigned Byte Integers |
| [PMOVMSKB](http://x86.renejeschke.de/html/file_module_x86_id_243.html) | Move Byte Mask |
| [PMULHUW](http://x86.renejeschke.de/html/file_module_x86_id_244.html) | Multiply Packed Unsigned Integers and Store High Result |
| [PMULHW](http://x86.renejeschke.de/html/file_module_x86_id_245.html) | Multiply Packed Signed Integers and Store High Result |
| [PMULLW](http://x86.renejeschke.de/html/file_module_x86_id_246.html) | Multiply Packed Signed Integers and Store Low Result |
| [PMULUDQ](http://x86.renejeschke.de/html/file_module_x86_id_247.html) | Multiply Packed Unsigned Doubleword Integers |
| [POP](http://x86.renejeschke.de/html/file_module_x86_id_248.html) | Pop a Value from the Stack |
| [POPA/POPAD](http://x86.renejeschke.de/html/file_module_x86_id_249.html) | Pop All General-Purpose Registers |
| [POPF/POPFD](http://x86.renejeschke.de/html/file_module_x86_id_250.html) | Pop Stack into EFLAGS Register |
| [POR](http://x86.renejeschke.de/html/file_module_x86_id_251.html) | Bitwise Logical OR |
| [PREFETCHh](http://x86.renejeschke.de/html/file_module_x86_id_252.html) | Prefetch Data Into Caches |
| [PSADBW](http://x86.renejeschke.de/html/file_module_x86_id_253.html) | Compute Sum of Absolute Differences |
| [PSHUFD](http://x86.renejeschke.de/html/file_module_x86_id_254.html) | Shuffle Packed Doublewords |
| [PSHUFHW](http://x86.renejeschke.de/html/file_module_x86_id_255.html) | Shuffle Packed High Words |
| [PSHUFLW](http://x86.renejeschke.de/html/file_module_x86_id_256.html) | Shuffle Packed Low Words |
| [PSHUFW](http://x86.renejeschke.de/html/file_module_x86_id_257.html) | Shuffle Packed Words |
| [PSLLDQ](http://x86.renejeschke.de/html/file_module_x86_id_258.html) | Shift Double Quadword Left Logical |
| [PSLLW/PSLLD/PSLLQ](http://x86.renejeschke.de/html/file_module_x86_id_259.html) | Shift Packed Data Left Logical |
| [PSRAW/PSRAD](http://x86.renejeschke.de/html/file_module_x86_id_260.html) | Shift Packed Data Right Arithmetic |
| [PSRLDQ](http://x86.renejeschke.de/html/file_module_x86_id_261.html) | Shift Double Quadword Right Logical |
| [PSRLW/PSRLD/PSRLQ](http://x86.renejeschke.de/html/file_module_x86_id_262.html) | Shift Packed Data Right Logical |
| [PSUBB/PSUBW/PSUBD](http://x86.renejeschke.de/html/file_module_x86_id_263.html) | Subtract Packed Integers |
| [PSUBQ](http://x86.renejeschke.de/html/file_module_x86_id_264.html) | Subtract Packed Quadword Integers |
| [PSUBSB/PSUBSW](http://x86.renejeschke.de/html/file_module_x86_id_265.html) | Subtract Packed Signed Integers with Signed Saturation |
| [PSUBUSB/PSUBUSW](http://x86.renejeschke.de/html/file_module_x86_id_266.html) | Subtract Packed Unsigned Integers with Unsigned Saturation |
| [PUNPCKHBW/PUNPCKHWD/PUNPCKHDQ/PUNPCKHQDQ](http://x86.renejeschke.de/html/file_module_x86_id_267.html) | Unpack High Data |
| [PUNPCKLBW/PUNPCKLWD/PUNPCKLDQ/PUNPCKLQDQ](http://x86.renejeschke.de/html/file_module_x86_id_268.html) | Unpack Low Data |
| [PUSH](http://x86.renejeschke.de/html/file_module_x86_id_269.html) | Push Word or Doubleword Onto the Stack |
| [PUSHA/PUSHAD](http://x86.renejeschke.de/html/file_module_x86_id_270.html) | Push All General-Purpose Registers |
| [PUSHF/PUSHFD](http://x86.renejeschke.de/html/file_module_x86_id_271.html) | Push EFLAGS Register onto the Stack |
| [PXOR](http://x86.renejeschke.de/html/file_module_x86_id_272.html) | Logical Exclusive OR |
| [RCL/RCR/ROL/ROR](http://x86.renejeschke.de/html/file_module_x86_id_273.html) | Rotate |
| [RCPPS](http://x86.renejeschke.de/html/file_module_x86_id_274.html) | Compute Reciprocals of Packed Single-Precision Floating-Point Values |
| [RCPSS](http://x86.renejeschke.de/html/file_module_x86_id_275.html) | Compute Reciprocal of Scalar Single-Precision Floating- Point Values |
| [RDMSR](http://x86.renejeschke.de/html/file_module_x86_id_276.html) | Read from Model Specific Register |
| [RDPMC](http://x86.renejeschke.de/html/file_module_x86_id_277.html) | Read Performance-Monitoring Counters |
| [RDTSC](http://x86.renejeschke.de/html/file_module_x86_id_278.html) | Read Time-Stamp Counter |
| [REP/REPE/REPZ/REPNE/REPNZ](http://x86.renejeschke.de/html/file_module_x86_id_279.html) | Repeat String Operation Prefix |
| [RET](http://x86.renejeschke.de/html/file_module_x86_id_280.html) | Return from Procedure |
| [RSM](http://x86.renejeschke.de/html/file_module_x86_id_281.html) | Resume from System Management Mode |
| [RSQRTPS](http://x86.renejeschke.de/html/file_module_x86_id_282.html) | Compute Reciprocals of Square Roots of Packed Single-Precision Floating-Point Values |
| [RSQRTSS](http://x86.renejeschke.de/html/file_module_x86_id_283.html) | Compute Reciprocal of Square Root of Scalar Single- Precision Floating-Point Value |
| [SAHF](http://x86.renejeschke.de/html/file_module_x86_id_284.html) | Store AH into Flags |
| [SAL/SAR/SHL/SHR](http://x86.renejeschke.de/html/file_module_x86_id_285.html) | Shift |
| [SBB](http://x86.renejeschke.de/html/file_module_x86_id_286.html) | Integer Subtraction with Borrow |
| [SCAS/SCASB/SCASW/SCASD](http://x86.renejeschke.de/html/file_module_x86_id_287.html) | Scan String |
| [SETcc](http://x86.renejeschke.de/html/file_module_x86_id_288.html) | Set Byte on Condition |
| [SFENCE](http://x86.renejeschke.de/html/file_module_x86_id_289.html) | Store Fence |
| [SGDT](http://x86.renejeschke.de/html/file_module_x86_id_290.html) | Store Global Descriptor Table Register |
| [SHLD](http://x86.renejeschke.de/html/file_module_x86_id_291.html) | Double Precision Shift Left |
| [SHRD](http://x86.renejeschke.de/html/file_module_x86_id_292.html) | Double Precision Shift Right |
| [SHUFPD](http://x86.renejeschke.de/html/file_module_x86_id_293.html) | Shuffle Packed Double-Precision Floating-Point Values |
| [SHUFPS](http://x86.renejeschke.de/html/file_module_x86_id_294.html) | Shuffle Packed Single-Precision Floating-Point Values |
| [SIDT](http://x86.renejeschke.de/html/file_module_x86_id_295.html) | Store Interrupt Descriptor Table Register |
| [SLDT](http://x86.renejeschke.de/html/file_module_x86_id_296.html) | Store Local Descriptor Table Register |
| [SMSW](http://x86.renejeschke.de/html/file_module_x86_id_297.html) | Store Machine Status Word |
| [SQRTPD](http://x86.renejeschke.de/html/file_module_x86_id_298.html) | Compute Square Roots of Packed Double-Precision Floating-Point Values |
| [SQRTPS](http://x86.renejeschke.de/html/file_module_x86_id_299.html) | Compute Square Roots of Packed Single-Precision Floating-Point Values |
| [SQRTSD](http://x86.renejeschke.de/html/file_module_x86_id_300.html) | Compute Square Root of Scalar Double-Precision Floating-Point Value |
| [SQRTSS](http://x86.renejeschke.de/html/file_module_x86_id_301.html) | Compute Square Root of Scalar Single-Precision Floating-Point Value |
| [STC](http://x86.renejeschke.de/html/file_module_x86_id_302.html) | Set Carry Flag |
| [STD](http://x86.renejeschke.de/html/file_module_x86_id_303.html) | Set Direction Flag |
| [STI](http://x86.renejeschke.de/html/file_module_x86_id_304.html) | Set Interrupt Flag |
| [STMXCSR](http://x86.renejeschke.de/html/file_module_x86_id_305.html) | Store MXCSR Register State |
| [STOS/STOSB/STOSW/STOSD](http://x86.renejeschke.de/html/file_module_x86_id_306.html) | Store String |
| [STR](http://x86.renejeschke.de/html/file_module_x86_id_307.html) | Store Task Register |
| [SUB](http://x86.renejeschke.de/html/file_module_x86_id_308.html) | Subtract |
| [SUBPD](http://x86.renejeschke.de/html/file_module_x86_id_309.html) | Subtract Packed Double-Precision Floating-Point Values |
| [SUBPS](http://x86.renejeschke.de/html/file_module_x86_id_310.html) | Subtract Packed Single-Precision Floating-Point Values |
| [SUBSD](http://x86.renejeschke.de/html/file_module_x86_id_311.html) | Subtract Scalar Double-Precision Floating-Point Values |
| [SUBSS](http://x86.renejeschke.de/html/file_module_x86_id_312.html) | Subtract Scalar Single-Precision Floating-Point Values |
| [SYSENTER](http://x86.renejeschke.de/html/file_module_x86_id_313.html) | Fast System Call |
| [SYSEXIT](http://x86.renejeschke.de/html/file_module_x86_id_314.html) | Fast Return from Fast System Call |
| [TEST](http://x86.renejeschke.de/html/file_module_x86_id_315.html) | Logical Compare |
| [UCOMISD](http://x86.renejeschke.de/html/file_module_x86_id_316.html) | Unordered Compare Scalar Double-Precision Floating- Point Values and Set EFLAGS |
| [UCOMISS](http://x86.renejeschke.de/html/file_module_x86_id_317.html) | Unordered Compare Scalar Single-Precision Floating- Point Values and Set EFLAGS |
| [UD2](http://x86.renejeschke.de/html/file_module_x86_id_318.html) | Undefined Instruction |
| [UNPCKHPD](http://x86.renejeschke.de/html/file_module_x86_id_319.html) | Unpack and Interleave High Packed Double- Precision Floating-Point Values |
| [UNPCKHPS](http://x86.renejeschke.de/html/file_module_x86_id_320.html) | Unpack and Interleave High Packed Single-Precision Floating-Point Values |
| [UNPCKLPD](http://x86.renejeschke.de/html/file_module_x86_id_321.html) | Unpack and Interleave Low Packed Double-Precision Floating-Point Values |
| [UNPCKLPS](http://x86.renejeschke.de/html/file_module_x86_id_322.html) | Unpack and Interleave Low Packed Single-Precision Floating-Point Values |
| [VERR/VERW](http://x86.renejeschke.de/html/file_module_x86_id_323.html) | Verify a Segment for Reading or Writing |
| [WAIT/FWAIT](http://x86.renejeschke.de/html/file_module_x86_id_324.html) | Wait |
| [WBINVD](http://x86.renejeschke.de/html/file_module_x86_id_325.html) | Write Back and Invalidate Cache |
| [WRMSR](http://x86.renejeschke.de/html/file_module_x86_id_326.html) | Write to Model Specific Register |
| [XADD](http://x86.renejeschke.de/html/file_module_x86_id_327.html) | Exchange and Add |
| [XCHG](http://x86.renejeschke.de/html/file_module_x86_id_328.html) | Exchange Register/Memory with Register |
| [XLAT/XLATB](http://x86.renejeschke.de/html/file_module_x86_id_329.html) | Table Look-up Translation |
| [XOR](http://x86.renejeschke.de/html/file_module_x86_id_330.html) | Logical Exclusive OR |
| [XORPD](http://x86.renejeschke.de/html/file_module_x86_id_331.html) | Bitwise Logical XOR for Double-Precision Floating-Point Values |
| [XORPS](http://x86.renejeschke.de/html/file_module_x86_id_332.html) | Bitwise Logical XOR for Single-Precision Floating-Point Values |

ARM

|  |  |  |
| --- | --- | --- |
| **Instruction** | **Meaning** | **Earliest CPU / Comments** |
| [*ABS*](https://www.heyrick.co.uk/assembler/fpops.html#unop) | Absolute Value | Floating Point 1 |
| [*ACS*](https://www.heyrick.co.uk/assembler/fpops.html#unop) | Arc Cosine | Floating Point 5 |
| [**ADC**](https://www.heyrick.co.uk/assembler/mov.html#adc) | Add with Carry | - |
| *ADC* | Thumb: Add with Carry | Thumb |
| [**ADD**](https://www.heyrick.co.uk/assembler/mov.html#add) | Add | - |
| *ADD* | Thumb: Add | Thumb |
| [*ADF*](https://www.heyrick.co.uk/assembler/fpops.html#binop) | Add | Floating Point 1 3 |
| [ADR](https://www.heyrick.co.uk/assembler/pseudo.html#adr) | Get address of object (within 4K) | This is an assembler pseudo-instruction |
| [ADRL](https://www.heyrick.co.uk/assembler/pseudo.html#adrl) | Get address of object (beyond 4K) | This is an assembler pseudo-instruction |
| [ALIGN](https://www.heyrick.co.uk/assembler/pseudo.html#align) | Set the program counter to the next word boundary | This is an assembler pseudo-instruction |
| [**AND**](https://www.heyrick.co.uk/assembler/mov.html#and) | Logical AND | - |
| *AND* | Thumb: Logical AND | Thumb |
| [ASL](https://www.heyrick.co.uk/assembler/shift.html#asl) | Arithmetic Shift Left | This is an option, not an instruction |
| [*ASN*](https://www.heyrick.co.uk/assembler/fpops.html#unop) | Arc Sine | Floating Point 5 |
| [ASR](https://www.heyrick.co.uk/assembler/shift.html#asr) | Arithmetic Shift Right | This is an option, not an instruction;  available on Thumb. |
| [*ATN*](https://www.heyrick.co.uk/assembler/fpops.html#unop) | Arc Tangent | Floating Point 5 |
| [**B**](https://www.heyrick.co.uk/assembler/bl.html#b) | Branch | - |
| *B* | Thumb: Branch | Thumb |
| [**BIC**](https://www.heyrick.co.uk/assembler/mov.html#bic) | Bit Clear | - |
| *BIC* | Thumb: Bit Clear | Thumb |
| *BKPT* | Thumb: Breakpoint | Thumb |
| [**BL**](https://www.heyrick.co.uk/assembler/bl.html#bl) | Branch with Link | - |
| *BL* | Thumb: Long Branch with Link | Thumb |
| *BLX* | Thumb: Branch with Link and Exchange | Thumb |
| *BX* | Thumb: Branch and Exchange | Thumb |
| [**CDP**](https://www.heyrick.co.uk/assembler/coprocmnd.html#cdp) | Co-processor data operation | - |
| *CDP2* | CDP, *non-conditional* so more co-processor commands possible | ARMv5 |
| *CLZ* | Count Leading Zeros | ARMv5 |
| [*CMF*](https://www.heyrick.co.uk/assembler/fpops.html#cmf) | Compare floating point value | Floating Point 1 3 |
| [**CMN**](https://www.heyrick.co.uk/assembler/cmp.html#cmn) | Compare negated values | - |
| *CMN* | Thumb: Compare negated values | Thumb |
| [**CMP**](https://www.heyrick.co.uk/assembler/cmp.html#cmp) | Compare values | - |
| *CMP* | Thumb: Compare values | Thumb |
| [*CNF*](https://www.heyrick.co.uk/assembler/fpops.html#cnf) | Compare negated floating point values | Floating Point 1 |
| [*COS*](https://www.heyrick.co.uk/assembler/fpops.html#unop) | Cosine | Floating Point 5 |
| [DCx](https://www.heyrick.co.uk/assembler/pseudo.html#dcx) | Define byte (B), halfword (W), word (D), string (S), or floating point (F) value. Some assemblers allow DCFS, DCFD, etc for FP precision. | This is an assembler pseudo-instruction |
| [*DVF*](https://www.heyrick.co.uk/assembler/fpops.html#binop) | Divide | Floating Point 1 3 |
| [**EOR**](https://www.heyrick.co.uk/assembler/mov.html#eor) | Exclusive-OR two values | - |
| *EOR* | Thumb: Logical Exclusive-OR | Thumb |
| [EQUx](https://www.heyrick.co.uk/assembler/pseudo.html#equx) | Define byte (B), halfword (W), word (D), string (S), or floating point (F) value. Some assemblers allow EQUFS, EQUFD, etc for FP precision. | This is an assembler pseudo-instruction |
| [*EXP*](https://www.heyrick.co.uk/assembler/fpops.html#unop) | Exponent | Floating Point 5 |
| *FABS* | VFP: Absolute | VFP |
| *FADD* | VFP: Addition | VFP |
| *FCMP* | VFP: Compare | VFP |
| *FCVTDS* | VFP: Single to Double | VFP |
| *FCVTSD* | VFP: Double to Single | VFP |
| *FCPY* | VFP: Copy [*like MVF*] | VFP |
| *FDIV* | VFP: Division | VFP |
| [*FDV*](https://www.heyrick.co.uk/assembler/fpops.html#binop) | Fast Divide | Floating Point 1 |
| [*FIX*](https://www.heyrick.co.uk/assembler/fpops.html#fix) | Convert floating value to an integer | Floating Point 1 3 |
| *FLD* | VFP: Load VFP registers | VFP |
| *FLDMDB* | VFP: Load multiple VFP registers, decr. before | VFP |
| *FLDMIA* | VFP: Load multiple VFP registers, incr. after | VFP |
| [*FLT*](https://www.heyrick.co.uk/assembler/fpops.html#flt) | Convert integer to a floating value | Floating Point 1 3 |
| *FMAC* | VFP: Multiply with Accumulate | VFP |
| *FMDHR* | VFP: Transfer ARM register to upper half of Double | VFP |
| *FMDLR* | VFP: Transfer ARM register to lower half of Double | VFP |
| *FMRDH* | VFP: Transfer upper half of Double to ARM register | VFP |
| *FMRDL* | VFP: Transfer lower half of Double to ARM register | VFP |
| [*FML*](https://www.heyrick.co.uk/assembler/fpops.html#binop) | Fast multiply | Floating Point 1 |
| *FMSC* | VFP: Multiply with Negate and Accumulate | VFP |
| *FMRS* | VFP: Transfer Single to ARM register | VFP |
| *FMSR* | VFP: Transfer ARM register to Single | VFP |
| *FMUL* | VFP: Multiply | VFP |
| *FMRX* | VFP: Transfer VFP system register to ARM register | VFP |
| *FMSTAT* | VFP: Transfer FPSCR flags to CPSR | VFP |
| *FMXR* | VFP: Transfer ARM register to VFP system register | VFP |
| *FNEG* | VFP: Copy Negative [*like MVN*] | VFP |
| *FNMAC* | VFP: Multiply with Deduct | VFP |
| *FNMSC* | VFP: Multiply with Negate and Deduct | VFP |
| *FNMUL* | VFP: Negative Multiply | VFP |
| [*FRD*](https://www.heyrick.co.uk/assembler/fpops.html#binop) | Fast reverse divide | Floating Point 1 |
| *FSITO* | VFP: Signed Integer to Float | VFP |
| *FSQRT* | VFP: Square Root | VFP |
| *FST* | VFP: Save VFP registers | VFP |
| *FSTMDB* | VFP: Save multiple VFP registers, decr. before | VFP |
| *FSTMIA* | VFP: Save multiple VFP registers, incr. after | VFP |
| *FSUB* | VFP: Subtraction | VFP |
| *FTOSI* | VFP: Float to Signed Integer | VFP |
| *FTOUI* | VFP: Float to Unsigned Integer | VFP |
| *FUITO* | VFP: Unsigned Integer to Float | VFP |
| [**LDC**](https://www.heyrick.co.uk/assembler/coprocmnd.html#ldc) | Load from memory to co-processor | - |
| *LDC2* | LDC, *non-conditional* so more co-processor commands possible | ARMv5 |
| [*LDF*](https://www.heyrick.co.uk/assembler/fpops.html#ldf) | Load floating point value | Floating Point 1 3 |
| [**LDM**](https://www.heyrick.co.uk/assembler/str.html#ldm) | Load multiple registers | - |
| *LDMIA* | Thumb: Load multiple registers | Thumb |
| [**LDR**](https://www.heyrick.co.uk/assembler/str.html#ldr) | Load register (32 bit) | - |
| *LDR* | Thumb: Load register (32 bits?) | Thumb |
| [**LDRB**](https://www.heyrick.co.uk/assembler/str.html#ldr) | Load byte (8 bit) into register | - |
| *LDRB* | Thumb: Load byte (8 bit) into register | Thumb |
| **LDRH** | Load halfword (16 bit) into register | StrongARM |
| *LDRH* | Thumb: Load halfwit (boo!) into register | Thumb |
| **LDRSB** | Load signed byte (sign + 7 bit) into register | StrongARM |
| *LDRSB* | Thumb: Load signed byte (sign + 7 bit) into register | Thumb |
| **LDRSH** | Load signed halfword (sign + 15 bit) into register | StrongARM |
| *LDRSH* | Thumb: Load signed halfword (sign + 15 bit) into register | Thumb |
| [*LFM*](https://www.heyrick.co.uk/assembler/fpops.html#lfm) | Load multiple floating point values | Floating Point 1 |
| [*LGN*](https://www.heyrick.co.uk/assembler/fpops.html#unop) | Logarithm to base e | Floating Point 5 |
| [*LOG*](https://www.heyrick.co.uk/assembler/fpops.html#unop) | Logarithm to base 10 | Floating Point 5 |
| [LSL](https://www.heyrick.co.uk/assembler/shift.html#lsl) | Logical Shift Left | This is an option, not an instruction;  available on Thumb. |
| [LSR](https://www.heyrick.co.uk/assembler/shift.html#lsr) | Logical Shift Right | This is an option, not an instruction;  available on Thumb. |
| [**MCR**](https://www.heyrick.co.uk/assembler/coprocmnd.html#mcr) | Co-processor register transfer (ARM to co-processor) | - |
| *MCR2* | MCR, *non-conditional* so more co-processor commands possible | ARMv5 |
| *MCRR* | MCR, with two registers transferred at one time | ARMv5TE |
| [**MLA**](https://www.heyrick.co.uk/assembler/mul.html#mla) | Multiply with Accumulate | - |
| [*MNF*](https://www.heyrick.co.uk/assembler/fpops.html#mnf) | Move negated | Floating Point 1 |
| [**MOV**](https://www.heyrick.co.uk/assembler/mov.html#mov) | Move value/register into a register | - |
| *MOV* | Thumb: Move value/register into a register | Thumb |
| [**MRC**](https://www.heyrick.co.uk/assembler/coprocmnd.html#mrc) | Co-processor register transfer (co-processor to ARM) | - |
| *MRC2* | MRC, *non-conditional* so more co-processor commands possible | ARMv5 |
| *MRRC* | MRC, with two registers transferred at one time | ARMv5TE |
| [**MRS**](https://www.heyrick.co.uk/assembler/psr.html#32bit) | Move status flags to a register | ARM 6 |
| [**MSR**](https://www.heyrick.co.uk/assembler/psr.html#32bit) | Move contents of a register to the status flags | ARM 6 |
| [*MUF*](https://www.heyrick.co.uk/assembler/fpops.html) | Multiply | Floating Point 1 3 |
| [**MUL**](https://www.heyrick.co.uk/assembler/mul.html#binop) | Multiply | - |
| *MUL* | Thumb: Multiply | Thumb |
| [*MVF*](https://www.heyrick.co.uk/assembler/fpops.html#mvf) | Move value/float register into a float register | Floating Point 1 3 |
| [**MVN**](https://www.heyrick.co.uk/assembler/mov.html#mvn) | Move negated | - |
| *MVN* | Thumb: Move negated | Thumb |
| *NEG* | Thumb Negate | Thumb |
| *NOP* | Thumb: No Operation | Thumb |
| [*NRM*](https://www.heyrick.co.uk/assembler/fpops.html#unop) | Normalise | Floating Point 1 |
| [OPT](https://www.heyrick.co.uk/assembler/opt.html) | Select assembly options | This is an assembler pseudo-instruction |
| [**ORR**](https://www.heyrick.co.uk/assembler/mov.html#orr) | Logical OR | - |
| *ORR* | Thumb: Logical OR | Thumb |
| *PLD* | PreLoaD | ARMv5 |
| [*POL*](https://www.heyrick.co.uk/assembler/fpops.html#binop) | Polar Angle | Floating Point 5 |
| *POP* | Thumb: Pop registers from stack | Thumb |
| [*POW*](https://www.heyrick.co.uk/assembler/fpops.html#binop) | Power | Floating Point 5 |
| *PUSH* | Thumb: Push registers onto stack | Thumb |
| *QADD* | Add, saturating | ARMv5E |
| *QDADD* | Add, double saturating | ARMv5E |
| *QDSUB* | Subtract, double saturating | ARMv5E |
| *QSUB* | Subtact, saturating | ARMv5E |
| [*RDF*](https://www.heyrick.co.uk/assembler/fpops.html#binop) | Reverse Divide | Floating Point 1 |
| [*RFC*](https://www.heyrick.co.uk/assembler/fpops.html#rfc) | Read FP control register | Floating Point 1 4 |
| [*RFS*](https://www.heyrick.co.uk/assembler/fpops.html#rfs) | Read FP status register | Floating Point 1 3 |
| [*RMF*](https://www.heyrick.co.uk/assembler/fpops.html#rmf) | Remainder | Floating Point 2 3 |
| [*RND*](https://www.heyrick.co.uk/assembler/fpops.html#unop) | Round to integral value | Floating Point 2 3 |
| [ROR](https://www.heyrick.co.uk/assembler/shift.html#ror) | Rotate Right | This is an option, not an instruction;  available on Thumb. |
| [*RPW*](https://www.heyrick.co.uk/assembler/fpops.html#binop) | Reverse Power | Floating Point 5 |
| [RRX](https://www.heyrick.co.uk/assembler/shift.html#rrx) | Rotate Right with extend | This is an option, not an instruction |
| [**RSB**](https://www.heyrick.co.uk/assembler/mov.html#rsb) | Reverse Subtract | - |
| [**RSC**](https://www.heyrick.co.uk/assembler/mov.html#rsc) | Reverse Subtract with Carry | - |
| [*RSF*](https://www.heyrick.co.uk/assembler/fpops.html#rsf) | Reverse Subtract | Floating Point 1 |
| [**SBC**](https://www.heyrick.co.uk/assembler/mov.html#sbc) | Subtract with Carry | - |
| *SBC* | Thumb: Subtract with Carry | Thumb |
| [*SFM*](https://www.heyrick.co.uk/assembler/fpops.html#sfm) | Store Muliple Floating point values | Floating Point 1 |
| [*SIN*](https://www.heyrick.co.uk/assembler/fpops.html#unop) | Sine | Floating Point 5 |
| *SMLA* | Signed Multiply with Accumulate of 16 bit \* 16 bit values | ARMv5E |
| **SMLAL** | Signed Long (sign + 63 bit) Multiply with Accumulate | StrongARM |
| *SMLAL* | Signed Multiply with Accumulate of 16 bit \* 16 bit values, result is sign extended to 32 bits, then added to a 64 bit value. | ARMv5E |
| *SMLAW* | Signed Multiply with Accumulate of 32 bit \* 16 bit values | ARMv5E |
| *SMUL* | Signed Multiply of 16 bit \* 16 bit values | ARMv5E |
| **SMULL** | Signed Long (sign + 63 bit) Multiply | StrongARM |
| *SMULW* | Signed Multiply of 32 bit \* 16 bit values | ARMv5E |
| [*SQT*](https://www.heyrick.co.uk/assembler/fpops.html#unop) | Square Root | Floating Point 2 3 |
| [**STC**](https://www.heyrick.co.uk/assembler/coprocmnd.html#stc) | Co-processor data transfer | - |
| *STC2* | STC, *non-conditional* so more co-processor commands possible | ARMv5 |
| [*STF*](https://www.heyrick.co.uk/assembler/fpops.html#stf) | Store floating point value | Floating Point 1 3 |
| [**STM**](https://www.heyrick.co.uk/assembler/str.html#stm) | Store multiple registers | - |
| *STMIA* | Thumb: Store multiple registers | Thumb |
| [**STR**](https://www.heyrick.co.uk/assembler/str.html#str) | Store a register (32 bit) | - |
| *STR* | Thumb: Store register (32 bit?) | Thumb |
| [**STRB**](https://www.heyrick.co.uk/assembler/str.html#str) | Store a byte (8 bit) from a register | - |
| *STRB* | Thumb: Store byte (8 bit) | Thumb |
| **STRH** | Store a halfword (16 bit) from a register | StrongARM |
| *STRH* | Thumb: Store halfword (16 bit) | Thumb |
| **STRSB** | Store a signed byte (sign + 7 bit) from a register | StrongARM |
| **STRSH** | Store a signed half-word (sign + 15 bit) from a register | StrongARM |
| [**SUB**](https://www.heyrick.co.uk/assembler/mov.html#sub) | Subtract | - |
| *SUB* | Thumb: Subtract | Thumb |
| [*SUF*](https://www.heyrick.co.uk/assembler/fpops.html#binop) | Subtract | Floating Point 1 3 |
| [**SWI**](https://www.heyrick.co.uk/assembler/swi.html#swi) | Cause a SoftWare Interrupt | - |
| *SWI* | Thumb: SoftWare Interrupt | Thumb |
| [**SWP**](https://www.heyrick.co.uk/assembler/mov.html#swp) | Swap register with memory | ARM 3 |
| [*TAN*](https://www.heyrick.co.uk/assembler/fpops.html#unop) | Tangent | Floating Point 5 |
| [**TEQ**](https://www.heyrick.co.uk/assembler/cmp.html#teq) | Test Equivalence (notional EOR) | - |
| [**TST**](https://www.heyrick.co.uk/assembler/cmp.html#tst) | Test bits (notional AND) | - |
| *TST* | Thumb: Test bits | Thumb |
| **UMLAL** | Unsigned Long (64 bit) Multiply with Accumulate | StrongARM |
| **UMULL** | Unsigned Long (64 bit) Multiply | StrongARM |
| [*URD*](https://www.heyrick.co.uk/assembler/fpops.html#unop) | Unnormalised round | Floating Point 1 |
| [WFC](https://www.heyrick.co.uk/assembler/fpops.html#wfc) | Write FP control register | Floating Point 1 4 |
| [*WFS*](https://www.heyrick.co.uk/assembler/fpops.html#wfs) | Write FP status register | Floating Point 1 3 |