

<code>mov dst, src</code>	src の値を dst にコピーする (move)。C 言語風に書くと <code>dst = src;</code>
<code>add dst, src</code>	src の値を dst に加える (add)。 <code>dst = dst + src;</code>
<code>adc dst, src</code>	src の値および CF の値を dst に加える (add with carry)。 <code>dst = dst + src + CF;</code>
<code>sub dst, src</code>	src の値を dst から減じる (subtract)。 <code>dst = dst - src;</code>
<code>sbb dst, src</code>	src の値および CF の値を dst から減じる (subtract with borrow)。 <code>dst = dst - src - CF;</code>
<code>inc dst</code>	dst の値を 1 増やす (increment)。 <code>dst = dst + 1;</code> または <code>dst++;</code>
<code>dec dst</code>	dst の値を 1 減じる (decrement)。 <code>dst = dst - 1;</code> または <code>dst--;</code>
<code>and dst, src</code>	src と dst のビットごとの論理積を dst に代入する。 <code>dst = dst & src;</code>
<code>or dst, src</code>	src と dst のビットごとの論理和を dst に代入する。 <code>dst = dst src;</code>
<code>xor dst, src</code>	src と dst のビットごとの排他的論理和を dst に代入する。 <code>dst = dst ^ src;</code>
<code>not dst</code>	dst の各ビットを反転する。 <code>dst = ~dst;</code>
<code>shl dst, cnt</code>	dst を cnt ビットだけ左シフトする (shift left)。 <code>dst = dst << cnt;</code>
<code>shr dst, cnt</code>	dst を cnt ビットだけ右シフトする (shift right)。 <code>dst = dst >>> cnt;</code>
<code>sar dst, cnt</code>	符号ビットを保存して dst を cnt ビットだけ右シフトする (shift arithmetic right)。 <code>dst = dst >> cnt;</code>
<code>rol dst, cnt</code>	dst を cnt ビットだけ左巡回シフトする (rotate left)。
<code>ror dst, cnt</code>	dst を cnt ビットだけ右巡回シフトする (rotate right)。
<code>rcl dst, cnt</code>	CF を含めて dst を cnt ビットだけ左巡回シフトする (rotate through carry left)。
<code>rcr dst, cnt</code>	CF を含めて dst を cnt ビットだけ右巡回シフトする (rotate through carry right)。
<code>mul m</code>	乗算を行う (multiply)。 <i>m</i> が 32 ビットの場合、EAX と <i>m</i> の積を計算し、上位 32 ビットを EDX に、下位 32 ビットを EAX に格納する。 <code>EDX:EAX = EAX * m;</code>
<code>div d</code>	除算を行う (divide)。 <i>d</i> が 32 ビットの場合、EDX を上位 32 ビット、EAX を下位 32 ビットとする値を <i>d</i> で割り、商を EAX に、剰余を EDX に格納する。 <code>EAX = EDX:EAX / d; EDX = EDX:EAX % d;</code>