**MOV** *<операнд1>, <операнд2>* - значение второго операнда записывается в первый операнд.

**CMOV** *<операнд1><операнд2>* - как и MOV, только выполняется в зависимости от значения тех или иных флагов из р-ра FLAGS

**XCHG** *<операнд1>, <операнд2>* - меняет операнды местами. (Операнды должны быть одной длины)

**byte/word/dword ptr** *mov…* – указывает размер пересылаемых элементов (1,2,4 байта)

**ADD** *<операнд1>, <операнд2>* - сложение (сумма записывается на место первого операнда)

**SUB** <*операнд1>, <операнд2>* - вычитание (разность записывается на место первого операнда)

**MUL** *<операнд>* - беззнаковое умножение операнда на AL/AX/EAX, произведение записывается в AX/DX:AX/EDX:EAX

**IMUL** *<операнд>/<операнд><непосредственный операнд>/<операнд1>,<операнд2><непосредственный операнд>  
/<операнд1><операнд2>* - знаковое умножение (запись так же)

**DIV** *<операнд>* - беззнаковое деление, делимое берётся из AL/DX:AX/EDX:EAX, частное помещается в AL/AX/EAX, остаток помещается в AH/DX/EDX

**IDIV** *<операнд>* - знаковое деление (аналогично)

**NEG** *<операнд>* - меняет знак операнда (нельзя юзать константу)

**NOP** – ничего не делает

**CBW** - знаковое расширение AL до AX (с 1 до 2 байт)

**CWD -** знаковое расширение AX до DX:AX (с 2 до 4 байт)

**CWDE -** знаковое расширение AX до EAX (с 4 до 8 байт)

**CDQ -** знаковое расширение EAX до EDX:EAX (с 8 до 16 байт)

**PUSH –** заносит данные в стек

**POP –** отдаёт данные из стека

**PUSHA** – сохраняет в стеке содержимое регистров общего назначения

**POPA** – команда извлечение из стека 2-х байтовых регистров общего назначения

**TEST** *<приёмник> <источник>*- выполняет команду логического умножения без сохранения результата

**CMP** *<приёмник><источник>* – сравнение через вычитание их приёмника

**JUMP** *ADRESS*– команда безусловного перехода к ADRESS:

**LOOP** *метка*– команда организации циклов (кол-во повторение – значение в регистре ecx/cx)