**Справочник команд**

| Мнемоника | Краткое описание |
| --- | --- |
| ADC | Сложение с учётом переноса. |
| ADD | Сложение. |
| AND | Логическое И. |
| BSF | Прямое сканирование битов. |
| BSR | Обратное сканирование битов. |
| BT | Проверка бита. |
| BTС | Проверка и инверсия бита. |
| BTR | Проверка и сброс бита. |
| BTS | Проверка и установка бита. |
| CALL | Вызов процедуры или переключение задачи. |
| CBW | Преобразование байта в слово. |
| CLC | Сброс флага переноса. |
| CLD | Сброс флага направления. |
| CLI | Сброс флага прерывания. |
| CMC | Инверсия флага переноса. |
| CDQ | Преобразование двойного слова в учетверённое слово. |
| CMP | Сравнение. |
| CWD | Преобразование слова в двойное слово. |
| CWDE | Преобразование слова в двойное слово. |
| DEC | Декремент. |
| DIV | Деление без знака. |
| IDIV | Деление со знаком. |
| ENTER | Установка кадра стека. |
| IMUL | Умножение со знаком. |
| INC | Инкремент. |
| J(COND) | Условный переход. |
| JMP | Безусловный переход. |
| LAHF | Загрузка младшего байта регистра флагов в AH. |
| LEAVE | Восстановление кадра стека. |
| LOOP | Управление циклом со счётчиком в регистре CX/ECX. |
| LOOPNZ/LOOPNE | Управление циклом со счётчиком в регистре CX/ECX. |
| LOOPZ/LOOPE | Управление циклом со счётчиком в регистре CX/ECX. |
| MOV | Копирование. |
| MOVSX | Копирование со знаковым расширением. |
| MOVZX | Копирование с нулевым расширением. |
| MUL | Умножение без знака. |
| NEG | Изменение знака. |
| NOP | Нет операции. |
| NOT | Инверсия. |
| OR | Логическое ИЛИ. |
| POP | Извлечение из стека. |
| POPA/POPAD | Восстановление регистров общего назначения из стека. |
| POPF/POPFD | Извлечение регистра флагов из стека. |
| PUSH | Помещение в стек. |
| PUSHA/PUSHAD | Помещение регистров общего назначения в стек. |
| PUSHF/PUSHFD | Помещение регистра флагов в стек. |
| RCL | Циклический сдвиг влево через флаг переноса. |
| RCR | Циклический сдвиг вправо через флаг переноса. |
| RET/RETN/RETF | Возврат из процедуры. |
| ROL | Циклический сдвиг влево. |
| ROR | Циклический сдвиг вправо. |
| SAHF | Запись содержимого AH в младший байт регистра флагов. |
| SAL | Арифметический сдвиг влево. |
| SAR | Арифметический сдвиг вправо. |
| SBB | Вычитание с заёмом. |
| SHL | Логический сдвиг влево. |
| SHLD | Сдвиг двойной точности влево. |
| SHR | Логический сдвиг вправо. |
| SHRD | Сдвиг двойной точности вправо. |
| STC | Установка флага переноса. |
| STD | Установка флага направления. |
| STI | Установка флага прерывания. |
| SUB | Вычитание. |
| TEST | Логическое сравнение. |
| XCHG | Обмен. |
| XOR | Логическое исключающее ИЛИ. |

**ADC**

|  |  |
| --- | --- |
| Синтаксис | ADC приемник,источник |
| Машинный код | 14 ib — ADC AL,imm8 15 iw — ADC AX,imm16 15 id — ADC EAX,imm32 80 /2 ib — ADC r/m8,imm8 81 /2 iw — ADC r/m16,imm16 81 /2 id — ADC r/m32,imm32 83 /2 ib — ADC r/m16,imm8 83 /2 ib — ADC r/m32,imm8 10 /r — ADC r/m8,r8 11 /r — ADC r/m16,r16 11 /r — ADC r/m32,r32 12 /r — ADC r8,r/m8 13 /r — ADC r16,r/m16 13 /r — ADC r32,r/m32 |
| Описание | Сложение с учётом переноса. |
| Действие | Сложение двух целочисленных двоичных операндов с учётом флага переноса CF. приёмник = приёмник + источник + CF |
| Флаги | OF=r SF=r ZF=r AF=r PF=r CF=r |

**ADD**

|  |  |
| --- | --- |
| Синтаксис | ADD приемник,источник |
| Машинный код | 04 ib — ADD AL,imm8 05 iw — ADD AX,imm16 05 id — ADD EAX,imm32 80 /0 ib — ADD r/m8,imm8 81 /0 iw — ADD r/m16,imm16 81 /0 id — ADD r/m32,imm32 83 /0 ib — ADD r/m16,imm8 83 /0 ib — ADD r/m32,imm8 00 /r — ADD r/m8,r8 01 /r — ADD r/m16,r16 01 /r — ADD r/m32,r32 02 /r — ADD r8,r/m8 03 /r — ADD r16,r/m16 03 /r — ADD r32,r/m32 |
| Описание | Сложение. |
| Действие | Сложение двух целочисленных двоичных операндов. приёмник = приёмник + источник |
| Флаги | OF=r SF=r ZF=r AF=r PF=r CF=r |

**AND**

|  |  |
| --- | --- |
| Синтаксис | AND приемник,источник |
| Машинный код | 24 ib — AND AL,imm8 25 iw — AND AX,imm16 25 id — AND EAX,imm32 80 /4 ib — AND r/m8,imm8 81 /4 iw — AND r/m16,imm16 81 /4 id — AND r/m32,imm32 83 /4 ib — AND r/m16,imm8 83 /4 ib — AND r/m32,imm8 20 /r — AND r/m8,r8 21 /r — AND r/m16,r16 21 /r — AND r/m32,r32 22 /r — AND r8,r/m8 23 /r — AND r16,r/m16 23 /r — AND r32,r/m32 |
| Описание | Логическое И. |
| Действие | Команда выполняет операцию логического И над соответствующими парами битов операндов. приёмник = приёмник AND источник |
| Флаги | OF=0 SF=r ZF=r AF=? PF=r CF=0 |

**BSF**

|  |  |
| --- | --- |
| Синтаксис | BSF результат,источник |
| Машинный код | OF BC — BSF r16,r/m16 OF BC — BSF r32,r/m32 |
| Описание | Прямое сканирование битов. |
| Действие | Команда осуществляет просмотр битов операнда *источник*, начиная с младшего. Номер позиции первого единичного бита записывается в регистр *результат*, флаг ZF устанавливается в 0. Если единичных битов нет, то флаг ZF устанавливается в 1, а регистр *результат* оказывается неопределенным. |
| Флаги | OF=? SF=? ZF=r AF=? PF=? CF=? |

**BSR**

|  |  |
| --- | --- |
| Синтаксис | BSR результат,источник |
| Машинный код | OF BD — BSR r16,r/m16 OF BD — BSR r32,r/m32 |
| Описание | Обратное сканирование битов. |
| Действие | Команда осуществляет просмотр битов операнда *источник*, начиная со старшего. Номер позиции первого единичного бита записывается в регистр *результат*, флаг ZF устанавливается в 0. Номер позиции считается, начиная с младшего бита. Если единичных битов нет, то флаг ZF устанавливается в 1, а регистр *результат* оказывается неопределенным. |
| Флаги | OF=? SF=? ZF=r AF=? PF=? CF=? |

**BT**

|  |  |
| --- | --- |
| Синтаксис | BT источник,индекс |
| Машинный код | OF A3 — BT r/m16,r16 OF A3 — BT r/m32,r32 OF BA /4 ib — BT r/m16,imm8 OF BA /4 ib — BT r/m32,imm8 |
| Описание | Проверка бита. |
| Действие | Номер проверяемого бита операнда *источник* задается операндом *индекс*. Команда копирует значение бита в флаг CF. |
| Флаги | OF=? SF=? ZF=? AF=? PF=? CF=r |

**BTC**

|  |  |
| --- | --- |
| Синтаксис | BTC источник,индекс |
| Машинный код | OF BB — BTC r/m16,r16 OF BB — BTC r/m32,r32 OF BA /7 ib — BTC r/m16,imm8 OF BA /7 ib — BTC r/m32,imm8 |
| Описание | Проверка и инверсия бита. |
| Действие | Номер проверяемого бита операнда *источник* задается операндом *индекс*. Команда копирует значение бита в флаг CF и затем инвертирует его. |
| Флаги | OF=? SF=? ZF=? AF=? PF=? CF=r |

**BTR**

|  |  |
| --- | --- |
| Синтаксис | BTR источник,индекс |
| Машинный код | OF B3 — BTR r/m16,r16 OF B3 — BTR r/m32,r32 OF BA /6 ib — BTR r/m16,imm8 OF BA /6 ib — BTR r/m32,imm8 |
| Описание | Проверка и сброс бита. |
| Действие | Номер проверяемого бита операнда *источник* задается операндом *индекс*. Команда копирует значение бита в флаг CF и затем сбрасывает его. |
| Флаги | OF=? SF=? ZF=? AF=? PF=? CF=r |

**BTS**

|  |  |
| --- | --- |
| Синтаксис | BTS источник,индекс |
| Машинный код | OF AB — BTS r/m16,r16 OF AB — BTS r/m32,r32 OF BA /5 ib — BTS r/m16,imm8 OF BA /5 ib — BTS r/m32,imm8 |
| Описание | Проверка и установка бита. |
| Действие | Номер проверяемого бита операнда *источник* задается операндом *индекс*. Команда копирует значение бита в флаг CF и затем устанавливает его в 1. |
| Флаги | OF=? SF=? ZF=? AF=? PF=? CF=r |

**CALL**

|  |  |
| --- | --- |
| Синтаксис | CALL операнд |
| Машинный код | E8 cw — CALL rel16 E8 cd — CALL rel32 FF /2 — CALL r/m16 FF /2 — CALL r/m32 9A cd — CALL ptr16:16 9A cp — CALL ptr16:32 FF /3 — CALL m16:16 FF /3 — CALL m16:32 |
| Описание | Вызов процедуры или переключение задачи. |
| Действие | Далее перечислены возможные варианты задания операнда:   * *rel16/32* — близкий относительный переход. Значение rel16/32 трактуется как знаковое и является смещением перехода относительно следующей за CALL команды в сегменте кода, то есть адрес перехода равен (EIP/IP) + (rel16/32). В стек заносится значение EIP/IP; * *r16(32)/m16(32)* — близкий абсолютный косвенный переход. Операнд — регистр г16(32) или ячейка памяти m16(32), содержащие адрес перехода в текущем сегменте кода. В EIP/IP загружается содержимое из r16(32) или m16(32). В стек заносится содержимое EIP/IP; * *ptr16:16(32)* — дальний абсолютный переход. Операнд — компоненты полного адреса в виде 4- или 6-разрядного указателя, по которому необходимо произвести переход; * *m16:16(32)* — дальний абсолютный косвенный переход. Операнд — адрес ячейки памяти размером 32(48) битов со структурой m16:16(32), содержащей компоненты адреса перехода.   Особенности выполнения команды CALL в двух последних вариантах определяются режимом работы процессора:   * в реальном режиме или режиме виртуального процессора 8086 (PE = 0 или PE = 1 и VM = 1) в EIP/IP и CS загружаются значения смещения и адреса сегмента из указателя в памяти или команды CALL. В стек заносится содержимое EIP/IP и CS; * в защищенном режиме (РЕ = 1 и VM = 0) анализируется байт прав доступа AR в дескрипторе целевого сегмента. В зависимости от его значения производится пять типов перехода к цели: передача управления подчиненному сегменту кода, передача управления неподчиненному сегменту кода, передача управления через шлюз вызова, переключение задачи (селектор соответствует дескриптору — шлюзу задачи), переключение задачи (селектор соответствует дескриптору — сегменту TSS). |
| Флаги | Изменяются только при переключении задачи. При переключении задачи значения флагов изменяются содержимым регистра EFLAGS в сегменте состояния TSS задачи, на которую производится переключение. |

**CBW/CWDE**

|  |  |
| --- | --- |
| Синтаксис | CBW/CWDE |
| Машинный код | 98 |
| Описание | CBW — преобразование байта в слово, CWDE — преобразование слова в двойное слово. |
| Действие | Команда копирует знаковый бит регистра AL/AX на все биты регистра АН/ЕАХ. |

**CLC**

|  |  |
| --- | --- |
| Синтаксис | CLC |
| Машинный код | F8 |
| Описание | Сброс флага переноса. |
| Флаги | CF = 0. |

**CLD**

|  |  |
| --- | --- |
| Синтаксис | CLD |
| Машинный код | FC |
| Описание | Сброс флага направления. |
| Флаги | DF = 0. |

**CLI**

|  |  |
| --- | --- |
| Синтаксис | CLI |
| Машинный код | FA |
| Описание | Сброс флага прерывания. |
| Флаги | IF = 0. |

**CMC**

|  |  |
| --- | --- |
| Синтаксис | CMC |
| Машинный код | F5 |
| Описание | Инверсия флага переноса. |
| Флаги | CF = r. |

**CWD/CDQ**

|  |  |
| --- | --- |
| Синтаксис | CWD/CDQ |
| Машинный код | 99 |
| Описание | CWD — преобразование слова в двойное слово, CDQ — преобразование двойного слова в учетверённое слово. |
| Действие | Команда CWD копирует значение старшего бита регистра АХ на все биты регистра DX; команда CDQ копирует знаковый бит регистра ЕАХ на все биты регистра EDX. |

**CBW/CWDE**

|  |  |
| --- | --- |
| Синтаксис | CBW/CWDE |
| Машинный код | 98 |
| Описание | CBW — преобразование байта в слово, CWDE — преобразование слова в двойное слово. |
| Действие | Команда копирует знаковый бит регистра AL/AX на все биты регистра АН/ЕАХ. |

**DEC**

|  |  |
| --- | --- |
| Синтаксис | DEC операнд |
| Машинный код | FE/1 — DEC r/m8 FF /1 — DEC r/m16 FF /1 — DECr/m32 48+rw — DEC r16 48+rd — DEC r32 |
| Описание | Декремент. |
| Действие | Уменьшение операнда на 1. Команда не воздействует на флаг CF. |
| Флаги | OF=r SF=r ZF=r AF=r PF=r |

**DIV**

|  |  |
| --- | --- |
| Синтаксис | DIV делитель |
| Машинный код | F6 /6 — DIV r/m8 F7 /6 — DIV r/m16 F7 /6 — DIV r/m32 |
| Описание | Деление без знака. |
| Действие | Команда выполняет целочисленное деление без учёта знака. Делимое задается неявно, и его размер зависит от размера делителя, который явно указывается в команде. Местоположения делимого, делителя, частного и остатка — в зависимости от их размерности:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Размер операнда | Делимое | Делитель | Частное | Остаток | | Байт | AX | r/m8 | AL | AH | | Слово | DX:AX | r/m16 | AX | DX | | Двойное слово | EAX:EDX | r/m32 | EAX | EDX | |
| Флаги | OF=? CF=? SF=? ZF=? AF=? PF=? |

**IDIV**

|  |  |
| --- | --- |
| Синтаксис | IDIV делитель |
| Машинный код | F6 /7 — IDIV r/m8 F7 /7 — IDIV r/m16 F7 /7 — IDIV r/m32 |
| Описание | Деление со знаком. |
| Действие | Команда выполняет целочисленное деление со знаком. Делимое задается неявно, и его размер зависит от размера делителя, который явно указывается в команде. Местоположения делимого, делителя, частного и остатка — в зависимости от их размерности:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Размер операнда | Делимое | Делитель | Частное | Остаток | | Байт | AX | r/m8 | AL | AH | | Слово | DX:AX | r/m16 | AX | DX | | Двойное слово | EDX:EAX | r/m32 | EAX | EDX |   Остаток всегда имеет знак делимого. Знак частного зависит от состояния знаковых битов (старших разрядов) делимого и делителя. |
| Флаги | OF=? CF=? SF=? ZF=? AF=? PF=? |

**ENTER**

|  |  |
| --- | --- |
| Синтаксис | ENTER размер\_кадра,лексический\_уровень |
| Машинный код | C8 iw ib — ENTER imm16,imm8 |
| Описание | Установка кадра стека. |
| Действие | Операнд *размер\_кадра* определяет размер кадра стека, операнд *лексический\_уровень* задает лексическую вложенность кадра стека. Значение лексического уровня определяет количество указателей кадра стека, копируемых в область дисплея нового кадра стека из предыдущего кадра. Оба операнда непосредственные. |
| Флаги | Не изменяются. |

**IMUL**

|  |  |
| --- | --- |
| Синтаксис | IMUL множитель\_1 IMUL множитель\_1, множитель\_2 IMUL произведение, множитель\_1, множитель\_2 |
| Машинный код | F6 /5 — IMUL r/m8 F7 /5 — IMUL r/m16 F7 /5 — IMUL r/m32 OF AF /r — IMUL r16,r/m16 OF AF /r — IMUL r32,r/m32 6B /r ib — IMUL r16,r/m16,imm8 68 /r ib — IMUL r32,r/m32,imm8 6B /r ib — IMUL r16,imm8 6B /r ib — IMUL r32,imm8 69 /r iw — IMUL r16,r/m16,imm16 69 /r id — IMUL r32,r/m32,imm32 69 /r iw — IMUL r16,imm16 69 /r id — IMUL r32,imm32 |
| Описание | Умножение со знаком. |
| Действие | Команда имеет три формы, различающиеся количеством операндов:   * С одним операндом — требует явного указания местоположения только одного сомножителя, который может быть расположен в ячейке памяти или регистре. Местоположение произведения зависит от размерности множителей:  |  |  |  |  | | --- | --- | --- | --- | | Размер операндов | Первый множитель | Второй множитель | Результат | | Байт | AL | r/m8 | AX | | Слово | AX | r/m16 | DX:AX | | Двойное слово | EAX | r/m32 | EDX:EAX |  * С двумя операндами — первый операнд определяет местоположение первого сомножителя. На его место впоследствии будет записан результат. Второй операнд определяет местоположение второго сомножителя. * С тремя операндами — первый операнд определяет местоположение результата, второй операнд — местоположение первого сомножителя, третий операнд может быть непосредственно заданным значением размером в байт, слово или двойное слово. |
| Флаги | * Для однооперандной команды: OF=1 CF=1 — значимые биты переносятся в верхнюю половину результата; OF=0 CF=0 — результат помещается точно в младшей половине результата. Состояние остальных флагов: SF=? ZF=? AF=? PF=?. * Для двух- и трехоперандной команды: OF=1 CF=1 — результат слишком большой и усекается; OF=0 CF=0 — размер результата точно соответствует операнду назначения. Состояние остальных флагов: SF=? ZF=? AF=? PF=?. |

**INC**

|  |  |
| --- | --- |
| Синтаксис | INC операнд |
| Машинный код | FE /0 — INC r/m8 FF /0 — INC r/m16 FF /0 — INC r/m32 40+rw — INC r16 40+rd — INC r32 |
| Описание | Инкремент. |
| Действие | Увеличение операнда на 1. Команда не воздействует на флаг CF. |
| Флаги | OF=r SF=r ZF=r AF=r PF=r |

**J(COND)**

|  |  |
| --- | --- |
| Синтаксис | J(COND) метка |
| Машинный код | 70 cb — JO rel8 71 cb — JNO rel8 72 cb — JC/JNAE/JB rel8 73 cb — JNC/JAE/JNB rel8 74 cb — JZ/JE rel8 75 cb — JNZ/JNE rel8 76 cb — JNA/JBE rel8 77 cb — JA/JNBE rel8 78 cb — JS rel8 79 cb — JNS rel8 7A cb — JP rel8 7B cb — JNP rel8 7C cb — JL/JNGE rel8 7D cb — JGE/JNL rel8 7E cb — JLE/JNG rel8 7F cb — JG/JNLE rel8 E3 cb — JCXZ/JECXZ rel8 0F 80 cw/cd — JO rel16/32 0F 81 cw/cd — JNO rel16/32 0F 82 cw/cd — JC/JNAE/JB rel16/32 0F 83 cw/cd — JNC/JAE/JNB rel16/32 0F 84 cw/cd — JZ/JE rel16/32 0F 85 cw/cd — JNZ/JNE rel16/32 0F 86 cw/cd — JNA/JBE rel16/32 0F 87 cw/cd — JA/JNBE rel16/32 0F 88 cw/cd — JS rel16/32 0F 89 cw/cd — JNS rel16/32 0F 8A cw/cd — JP rel16/32 0F 8B cw/cd — JNP rel16/32 0F 8C cw/cd — JL/JNGE rel16/32 0F 8D cw/cd — JGE/JNL rel16/32 0F 8E cw/cd — JLE/JNG rel16/32 0F 8F cw/cd — JG/JNLE rel16/32 |
| Описание | Условный переход. |
| Действие | Команды условного перехода, в зависимости от своей мнемоники, анализируют флаги, и если проверяемое условие истинно, то производится переход к ячейке, обозначенной операндом. Если проверяемое условие ложно, то производится переход к следующей команде.  Идентификатор *метка* преобразуется ассемблером в непосредственные значения rel8(use16) и rel16/rel32 (use32), которые во время выполнения добавляются к текущему значению IP/EIP. Мнемоника команд условного перехода показана в таблице (логические условия «больше» и «меньше» относятся к сравнениям целочисленных значений со знаком, а «выше» и «ниже» — к сравнениям целочисленных значений без знака).   |  |  |  | | --- | --- | --- | | **Команда** | **Переход, если** | **Условие перехода** | | JZ/JE | нуль или равно | ZF=1 | | JNZ/JNE | не нуль или не равно | ZF=0 | | JC/JNAE/JB | есть переполнение/не выше и не равно/ниже | CF=1 | | JNC/JAE/JNB | нет переполнения/выше или равно/не ниже | CF=0 | | JP | число единичных бит чётное | PF=1 | | JNP | число единичных бит нечётное | PF=0 | | JS | знак равен 1 | SF=1 | | JNS | знак равен 0 | SF=0 | | JO | есть переполнение | OF=1 | | JNO | нет переполнения | OF=0 | | JA/JNBE | выше/не ниже и не равно | CF=0 и ZF=0 | | JNA/JBE | не выше/ниже или равно | CF=1 или ZF=1 | | JG/JNLE | больше/не меньше и не равно | ZF=0 и SF=OF | | JGE/JNL | больше или равно/не меньше | SF=OF | | JL/JNGE | меньше/не больше и не равно | SF≠OF | | JLE/JNG | меньше или равно/не больше | ZF=1 или SF≠OF | | JCXZ/JECXZ | содержимое CX/ECX равно нулю | CX/ECX=0 | |
| Флаги | Не изменяются. |

**JMP**

|  |  |
| --- | --- |
| Синтаксис | JMP операнд |
| Машинный код | EB cb — JMP rel8 E9 cw — JMP rel16 E9 cd — JMP rel32 FF /4 — JMP r/m16 FF /4 — JMP r/m32 EA cd — JMP ptr16:16 EA cp — JMP ptr16:32 FF /5 — JMP m16:16 FF /5 — JMP m16:32 |
| Описание | Безусловный переход. |
| Действие | Далее перечислены возможные варианты задания операнда:   * *rel8/16/32* — короткий относительный переход. Значение rel8/16/32 трактуется как знаковое и является смещением перехода относительно следующей за JMP команды в сегменте кода, то есть адрес перехода равен (EIP/IP) + (rel8/16/32); * *r16(32)/m16(32)* — близкий абсолютный косвенный переход. Операнд — регистр г16(32) или ячейка памяти m16(32), содержащие адрес перехода в текущем сегменте кода; * *ptr16:16(32)* — дальний абсолютный переход. Операнд — компоненты полного адреса в виде 4- или 6-разрядного указателя, по которому необходимо произвести переход; * *m16:16(32)* — дальний абсолютный косвенный переход. Операнд — адрес ячейки памяти размером 32(48) битов со структурой m16:16(32), содержащей компоненты адреса перехода.   Выполнение команды при дальнем переходе зависит от режима работы процессора:   * в реальном режиме или режиме виртуального процессора 8086 команда JMP передает управление по адресу, определяемому операндом, который может задаваться прямо (ptr16:16(32)) или косвенно (m1б:16(32)); * в защищенном режиме выполняются три типа переходов: дальний переход в подчиненный или неподчиненный сегмент кода; дальний переход через шлюз вызова; переключение задачи.   Команду JMP нельзя использовать для передачи управления между уровнями привилегий. |
| Флаги | Изменяются только при переключении задачи. |

**LAHF**

|  |  |
| --- | --- |
| Синтаксис | LAHF |
| Машинный код | 9F |
| Описание | Загрузка младшего байта регистра флагов в AH. |
| Действие | EFLAGS/FLAGS(SF:ZF:0:AF:0:PF:1:CF) → АН, здесь в скобках указаны порядок следования флагов, загружаемых в АН, и значения полей с фиксированным значением. |
| Флаги | Не изменяются. |

**LEAVE**

|  |  |
| --- | --- |
| Синтаксис | LEAVE |
| Машинный код | C9 — LEAVE |
| Описание | Восстановление кадра стека. |
| Действие | Команда копирует регистр ЕВР/ВР в ESP/SP, тем самым восстанавливая в регистре ESP/SP то значение, которое было до вызова процедуры. Далее в регистр ЕВР/ВР значение извлекается из новой вершины стека, тем самым восстанавливается кадр вызывающей процедуры. |
| Флаги | Не изменяются. |

**LOOP**

|  |  |
| --- | --- |
| Синтаксис | LOOP метка |
| Машинный код | E2 cb — LOOP rel8 |
| Описание | Управление циклом со счетчиком в регистре СХ/ЕСХ. |
| Действие | Выполнить декремент содержимого регистра СХ/ЕСХ; проанализировать регистр СХ/ЕСХ: если СХ/ЕСХ=0, передать управление следующей за LOOP команде, если СХ/ЕСХ≠0, передать управление команде, метка которой указана в качестве операнда. Смещение метки относительно текущего значения регистра IР/ЕIР должно быть в диапазоне -128…+127 байт. |
| Флаги | Не изменяются. |

**LOOPNZ/LOOPNE**

|  |  |
| --- | --- |
| Синтаксис | LOOPNZ/LOOPNE метка |
| Машинный код | E0 cb — LOOPNZ/LOOPNE rel8 |
| Описание | Управление циклом со счетчиком в регистре СХ/ЕСХ. |
| Действие | Выполнить декремент содержимого регистра СХ/ЕСХ; проанализировать регистр СХ/ЕСХ и флаг ZF. Если СХ/ЕСХ=0 или ZF=1, передать управление следующей команде, если СХ/ЕСХ≠0 и ZF=0, передать управление команде, метка которой указана в качестве операнда. Смещение метки относительно текущего значения регистра IР/ЕIР должно быть в диапазоне -128…+127 байт. |
| Флаги | Не изменяются. |

**LOOPZ/LOOPE**

|  |  |
| --- | --- |
| Синтаксис | LOOPZ/LOOPE метка |
| Машинный код | E1 cb — LOOPZ/LOOPE rel8 |
| Описание | Управление циклом со счетчиком в регистре СХ/ЕСХ. |
| Действие | Выполнить декремент содержимого регистра СХ/ЕСХ; проанализировать регистр СХ/ЕСХ и флаг ZF. Если СХ/ЕСХ=0 или ZF=0, передать управление следующей команде, если СХ/ЕСХ≠0 и ZF=1, передать управление команде, метка которой указана в качестве операнда. Смещение метки относительно текущего значения регистра IР/ЕIР должно быть в диапазоне -128…+127 байт. |
| Флаги | Не изменяются. |

**MOV**

|  |  |
| --- | --- |
| Синтаксис | MOV приемник,источник |
| Машинный код | 88 /r — MOV r/m8,r8 89 /r — MOV r/m16,r16 89 /r — MOV r/m32,r32 8А /r — MOV r8,r/m8 8В /r — MOV r16,r/m16 8В /r — MOV r32,r/m32 8C /r — MOV r/m16,Sreg\*\* 8E /r — MOV Sreg,r/m16 А0 — MOV AL,moffs8\* A1 — MOV AX,moffs16\* A1 — MOV EAX,moffs32\* A2 — MOV moffs8\*,AL A3 — MOV moffs16\*,AX A3 — MOV moffs32\*,EAX B0+rb — MOV r8,imm8 B8+rw — MOV r16,imm16 B8+rd — MOV r32,imm32 C6 /0 — MOV r/m8,imm8 C7 /0 — MOV r/m16,imm16 C7 /0 — MOV r/m32,imm32 |
| Описание | Копирование. |
| Действие | Копирование содержимого операнда источник в операнд приемник. |
| Флаги | OF = ? SF = ? ZF = ? AF = ? PF = ? CF = ? |

**MOVSX**

|  |  |
| --- | --- |
| Синтаксис | MOVSX приемник, источник |
| Машинный код | OF BE/r — MOVSX r16,r/m8 OF BE/r — MOVSX r32,r/m8 OF BF/r — MOVSX r32,r/m16 |
| Описание | Копирование со знаковым расширением. |
| Действие | Команда преобразует операнд со знаком в эквивалентный ему операнд со знаком большей размерности. Для этого содержимое операнда источник начиная с младших разрядов записывается в операнд приемник. Старшие биты операнда приемник заполняются значением знакового разряда операнда источник. |

**MOVZX**

|  |  |
| --- | --- |
| Синтаксис | MOVZX приемник, источник |
| Машинный код | OF B6/r — MOVZX r16,r/m8 OF B6/r — MOVZX r32,r/m8 OF B7/r — MOVZX r32,r/m16 |
| Описание | Копирование с нулевым расширением. |
| Действие | Команда преобразует операнд без знака в эквивалентный ему операнд без знака большей размерности. Для этого содержимое операнда источник, начиная с его младших разрядов, записывается в операнд приемник. Старшие разряды операнда приемник заполняются нулем. |

**MUL**

|  |  |
| --- | --- |
| Синтаксис | MUL множитель |
| Машинный код | F6 /4 — MUL r/m8 F7 /4 — MUL r/m16 F7 /4 — MUL r/m32 |
| Описание | Умножение без знака. |
| Действие | Команда выполняет целочисленное умножение без учёта знака. Явно задается один из множителей. Второй множитель задается неявно в регистре AL/AX/EAX (это местоположение фиксировано). Местоположение результата умножения определяется кодом операции и размером множителей:   |  |  |  |  | | --- | --- | --- | --- | | Размер операндов | Первый множитель | Второй множитель | Результат | | Байт | AL | r/m8 | AX | | Слово | AX | r/m16 | DX:AX | | Двойное слово | EAX | r/m32 | EDX:EAX | |
| Флаги | Если старшая половина результата нулевая: OF=CF=0 SF=? ZF=? AF=? PF=?. Если старшая половина результата ненулевая: OF=CF=1 SF=? ZF=? AF=? PF=?. |

**NEG**

|  |  |
| --- | --- |
| Синтаксис | NEG операнд |
| Машинный код | F6 /3 — NEG r/m8 F7 /3 — NEG r/m16 F7 /3 — NEG r/m32 |
| Описание | Изменение знака. |
| Действие | Команда вычисляет двоичное дополнение операнда. |
| Флаги | Если операнд равен нулю: CF=0 OF=r SF=r ZF=r AF=r PF=r. Если операнд не равен нулю: CF=1 OF=r SF=r ZF=r AF=r PF=r. |

**NOP**

|  |  |
| --- | --- |
| Синтаксис | NOP |
| Машинный код | 90 |
| Описание | Нет операции. |
| Действие | Отсутствует. Единственный эффект от команды NOP — инкремент регистра EIP. |
| Флаги | Не изменяются. |

**NOT**

|  |  |
| --- | --- |
| Синтаксис | NOT операнд |
| Машинный код | F6 /2 — NOT r/m8 F7 /2 — NOT r/m16 F7 /2 — NOT r/m32 |
| Описание | Инверсия. |
| Действие | Инвертирование всех битов операнда. |
| Флаги | Не изменяются. |

**OR**

|  |  |
| --- | --- |
| Синтаксис | OR приемник,источник |
| Машинный код | 0C ib — OR AL,imm8 0D iw — OR AX,imm16 0D id — OR EAX,imm32 80 /1 ib — OR r/m8,imm8 81 /1 iw — OR r/m16,imm16 81 /1 id — OR r/m32,imm32 83 /1 ib — OR r/m16,imm8 83 /1 ib — OR r/m32,imm8 08 /r — OR r/m8,r8 09 /r — OR r/m16,r16 09 /r — OR r/m32,r32 0A /r — OR r8,r/m8 0B /r — OR r16,r/m16 0B /r — OR r32,r/m32 |
| Описание | Логическое ИЛИ. |
| Действие | Команда выполняет операцию логического ИЛИ над соответствующими парами битов операндов. приёмник = приёмник OR источник |
| Флаги | OF=0 SF=r ZF=r AF=? PF=r CF=0 |

**POP**

|  |  |
| --- | --- |
| Синтаксис | POP приёмник |
| Машинный код | 8F /0 — POP r/m16 8F /0 — POP r/m32 58+rw — POP r16 58+rd — POP r32 1F — POP DS 07 — POP ES 17 — POP SS 0F A1 — POP FS 0F A9 — POP GS |
| Описание | Извлечение из стека. |
| Действие | Команда восстанавливает содержимое вершины стека в регистр, ячейку памяти или сегментный регистр, после чего значение ESP(SP) увеличивается на 4(2). Недопустимо восстановление значения в сегментный регистр CS. |
| Флаги | Не изменяются. |

**POPA/POPAD**

|  |  |
| --- | --- |
| Синтаксис | POPA/POPAD |
| Машинный код | 61 — POPA/POPAD |
| Описание | Восстановление регистров общего назначения из стека. |
| Действие | Команда POPA/POPAD восстанавливает содержимое всех регистров общего назначения (DI/EDI, SI/ESI, ВР/ЕВР, SP/ESP, BX/EBX, DX/EDX, СХ/ЕСХ, АХ/ЕАХ) из стека, после чего значение указателя стека SP/ESP увеличивается на 16(32). Содержимое DI/EDI восстанавливается первым. Содержимое SP/ESP при этом не восстанавливается. Какие именно регистры — 16- или 32-разрядные — извлекаются из стека, зависит от установленного размера операнда. |
| Флаги | Не изменяются. |

**PUSHF/PUSHFD**

|  |  |
| --- | --- |
| Синтаксис | PUSHF/PUSHFD |
| Машинный код | 9C — PUSHF/PUSHFD |
| Описание | Помещение регистра флагов в стек. |
| Действие | Команда уменьшает значение указателя стека SP/ESP на 2(4), после чего помещает в вершину стека содержимое регистра FLAGS/EFLAGS. При этом флаги VM и RF (биты 16 и 17) не копируются, вместо этого значения для этих флагов в образе EFLAGS, сохраненном на стеке, устанавливаются равными нулю. |
| Флаги | Не изменяются. |

**RCL**

|  |  |
| --- | --- |
| Синтаксис | RCL операнд, количество\_сдвигов |
| Машинный код | D0 /2 — RCL r/m8,1 D2 /2 — RCL r/m8,CL С0 /2 ib — RCL r/m8,imm8 D1 /2 — RCL r/m16,1 D3 /2 — RCL r/m16,CL C1 /2 ib — RCL r/m16,imm8 D1 /2 — RCL r/m32,1 D3 /2 — RCL r/m32,CL C1 /2 ib — RCL r/m32,imm8 |
| Описание | Циклический сдвиг влево через флаг переноса. |
| Действие | При циклическом сдвиге разрядов операнда влево его старший бит становится значением флага переноса CF. Старое содержимое CF вдвигается в операнд справа и становится значением его младшего бита. |
| Флаги | CF=r OF=?r SF=r ZF=r PF=r AF=? Флаг OF представляет интерес, если сдвиг осуществляется на один разряд. При сдвиге на несколько разрядов его значение не определено. По его значению можно судить об изменении знакового разряда операнда:   * если OF=1, то значение флага CF и значение старшего бита операнда после выполнения сдвига различны; * если OF=0, то значение флага CF и значение старшего бита операнда после выполнения сдвига совпадают. |

**RCR**

|  |  |
| --- | --- |
| Синтаксис | RCR операнд, количество\_сдвигов |
| Машинный код | D0 /3 — RCR r/m8,1 D2 /3 — RCR r/m8,CL С0 /3 ib — RCR r/m8,imm8 D1 /3 — RCR r/m16,1 D3 /3 — RCR r/m16,CL C1 /3 ib — RCR r/m16,imm8 D1 /3 — RCR r/m32,1 D3 /3 — RCR r/m32,CL C1 /3 ib — RCR r/m32,imm8 |
| Описание | Циклический сдвиг вправо через флаг переноса. |
| Действие | При циклическом сдвиге разрядов операнда вправо его младший бит становится значением флага переноса CF. Старое содержимое CF вдвигается в операнд слева и становится значением его старшего бита. |
| Флаги | CF=r OF=?r SF=r ZF=r PF=r AF=? Флаг CF содержит значение последнего выдвинутого из операнда бита. Флаг OF представляет интерес, если сдвиг осуществляется на один разряд. При сдвиге на несколько разрядов его значение не определено. По его значению можно судить об изменении знакового разряда операнда:   * если OF=1, то значение флага CF и значение старшего бита операнда после выполнения сдвига различны; * если OF=0, то значение флага CF и значение старшего бита операнда после выполнения сдвига совпадают. |

**RET/RETN/RETF**

|  |  |
| --- | --- |
| Синтаксис | RET число |
| Машинный код | СЗ — ret (близкий возврат) СВ — ret (дальний возврат) С2 i16 — ret i16 (близкий возврат с выталкиванием i16 байт из стека) СА i16 — ret i16 (дальний возврат с выталкиванием i16 байт из стека) |
| Описание | Возврат из процедуры. |
| Действие | Передача управления по адресу, расположенному на вершине стека. Этот адрес обычно помещается в стек командой CALL, его значение соответствует команде, следующей за командой CALL. Необязательный операнд *число* определяет количество байтов стека, которые будут вытолкнуты после выталкивания адреса возврата. Команда RET используется для выполнения трех вариантов возврата управления:   * Близкий возврат — возврат управления вызывающей процедуре в пределах текущего сегмента кода (внутрисегментный возврат). При этом из вершины стека выталкивается значение в регистр EIP/IP. Регистр CS не изменяется. Процессор продолжает выполнение кода в том же сегменте, но по другому смещению. * Дальний возврат — возврат управления вызывающей процедуре, расположенной в отличном от текущего сегменте кода (межсегментный возврат). При этом из вершины стека последовательно выталкиваются значения в регистры EIP/IP и CS. Процессор продолжает выполнение кода в другом сегменте. * Дальний возврат между уровнями привилегий — дальний возврат управления коду на уровне привилегий, отличном от текущего. Этот вид возврата может быть выполнен только в защищенном режиме. Его механизм подобен механизму дальнего возврата за исключением того, что процессор следит за уровнями привилегий и правами доступа к сегментам кода и стека, которым возвращается управление для определения возможности подобной передачи. Команда RET очищает регистры DS, ES, FS и GS, если они ссылаются на соответствующие сегменты, недоступные на новом уровне привилегий. Так как при межуровневом возврате управления производится переключение стека, то команда RET также производит загрузку этих регистров из стека. |
| Флаги | Изменяются только при переключении задачи. |

**ROL**

|  |  |
| --- | --- |
| Синтаксис | ROL операнд, количество\_сдвигов |
| Машинный код | D0 /0 — ROL r/m8,1 D2 /0 — ROL r/m8,CL С0 /0 ib — ROL r/m8,imm8 D1 /0 — ROL r/m16,1 D3 /0 — ROL r/m16,CL C1 /0 ib — ROL r/m16,imm8 D1 /0 — ROL r/m32,1 D3 /0 — ROL r/m32,CL C1 /0 ib — ROL r/m32,imm8 |
| Описание | Циклический сдвиг влево. |
| Действие | При циклическом сдвиге разрядов операнда влево его старший выдвигаемый бит вдвигается в операнд справа и становится одновременно значением младшего бита операнда и флага переноса CF. |
| Флаги | CF=r OF=?r SF=r ZF=r PF=r AF=? Флаг OF представляет интерес, если сдвиг осуществляется на один разряд. При сдвиге на несколько разрядов его значение не определено. По его значению можно судить об изменении знакового разряда операнда:   * если OF=1, то значение флага CF и значение старшего бита операнда после выполнения сдвига различны; * если OF=0, то значение флага CF и значение старшего бита операнда после выполнения сдвига совпадают. |

**ROR**

|  |  |
| --- | --- |
| Синтаксис | ROR операнд, количество\_сдвигов |
| Машинный код | D0 /1 — ROR r/m8,1 D2 /1 — ROR r/m8,CL С0 /1 ib — ROR r/m8,imm8 D1 /1 — ROR r/m16,1 D3 /1 — ROR r/m16,CL C1 /1 ib — ROR r/m16,imm8 D1 /1 — ROR r/m32,1 D3 /1 — ROR r/m32,CL C1 /1 ib — ROR r/m32,imm8 |
| Описание | Циклический сдвиг вправо. |
| Действие | При циклическом сдвиге разрядов операнда вправо его младший выдвигаемый бит вдвигается в операнд слева и становится одновременно значением старшего бита операнда и флага переноса CF. |
| Флаги | CF=r OF=?r SF=r ZF=r PF=r AF=? Флаг OF представляет интерес, если сдвиг осуществляется на один разряд. При сдвиге на несколько разрядов его значение не определено. По его значению можно судить об изменении знакового разряда операнда:   * если OF=1, то значение флага CF и значение старшего бита операнда после выполнения сдвига различны; * если OF=0, то значение флага CF и значение старшего бита операнда после выполнения сдвига совпадают. |

**SAHF**

|  |  |
| --- | --- |
| Синтаксис | SAHF |
| Машинный код | 9E |
| Описание | Запись содержимого AH в младший байт регистра флагов. |
| Действие | Команда загружает значение из регистра АН в младший байт регистра EFLAGS/FLAGS. При этом флаги SF, ZF, AF, PF и CF инициализируются битами 7, 6, 4, 2, и 0 регистра АН. Биты 1, 3 и 5 регистра EFLAGS/FLAGS не изменяются, то есть остаются равными значениям 1, 0 и 0 соответственно. |
| Флаги | SF=r ZF=r AF=r PF=r CF=r. |

**SHL/SAL**

|  |  |
| --- | --- |
| Синтаксис | SHL/SAL операнд, количество\_сдвигов |
| Машинный код | D0 /4 — SHL r/m8,1 D2 /4 — SHL r/m8,CL С0 /4 ib — SHL r/m8,imm8 D1 /4 — SHL r/m16,1 D3 /4 — SHL r/m16,CL C1 /4 ib — SHL r/m16,imm8 D1 /4 — SHL r/m32,1 D3 /4 — SHL r/m32,CL C1 /4 ib — SHL r/m32,imm8 |
| Описание | Сдвиг влево. |
| Действие | Сдвиг всех битов операнда влево на количество разрядов, указанное операндом количество\_сдвигов, при этом выдвигаемый слева бит становится значением флага переноса CF. Одновременно справа в операнд вдвигается нулевой бит. |
| Флаги | CF=r OF=?r SF=r ZF=r PF=r AF=? Флаг CF содержит значение последнего выдвинутого из операнда бита. Флаг OF представляет интерес, если сдвиг осуществляется на один разряд. При сдвиге на несколько разрядов его значение не определено. По его значению можно судить об изменении знакового разряда операнда:   * если OF=1, то значение флага CF и значение старшего бита операнда после выполнения сдвига различны; * если OF=0, то значение флага CF и значение старшего бита операнда после выполнения сдвига совпадают. |

**SHLD**

|  |  |
| --- | --- |
| Синтаксис | SHLD приёмник, источник, количество\_сдвигов |
| Машинный код | 0F А4 — SHLD r/m16,r16,imm8 0F A5 — SHLD r/m16,r16,CL 0F A4 — SHLD r/m32,r32,imm8 0F A5 — SHLD r/m32,r32,CL |
| Описание | Сдвиг двойной точности влево. |
| Действие | Команда сдвигает операнд приемник влево на число битов, указанных операндом количество\_сдвигов. Операнд источник обеспечивает биты, которые вдвигаются в приемник справа (начиная с бита 0 приемника). Операнд количество\_сдвигов — целое число без знака, которое может быть непосредственным 8-разрядным значением или содержимым регистра CL. |
| Флаги | CF=r OF=?r SF=r ZF=r PF=r AF=? Флаг CF заполняется последним битом, сдвинутым из операнда приемник. Флаг OF устанавливается в 1 для одноразрядного сдвига, если изменился знаковый разряд приемника, иначе он равен нулю. Если операнд количество\_сдвигов равен нулю, то флаги не изменяются. |

**SHR**

|  |  |
| --- | --- |
| Синтаксис | SHR операнд, количество\_сдвигов |
| Машинный код | D0 /5 — SHR r/m8,1 D2 /5 — SHR r/m8,CL С0 /5 ib — SHR r/m8,imm8 D1 /5 — SHR r/m16,1 D3 /5 — SHR r/m16,CL C1 /5 ib — SHR r/m16,imm8 D1 /5 — SHR r/m32,1 D3 /5 — SHR r/m32,CL C1 /5 ib — SHR r/m32,imm8 |
| Описание | Логический сдвиг вправо. |
| Действие | Сдвиг всех битов операнда вправо на количество разрядов, указанное операндом количество\_сдвигов, при этом выдвигаемый справа бит становится значением флага переноса CF. Одновременно слева в операнд вдвигается нулевой бит. |
| Флаги | CF=r OF=?r SF=r ZF=r PF=r AF=? Флаг CF содержит значение последнего выдвинутого из операнда бита. При сдвиге на 1 бит флаг OF устанавливается равным старшему значащему биту исходного операнда. Если количество\_сдвигов больше 1, то OF=0. |

**SHRD**

|  |  |
| --- | --- |
| Синтаксис | SHRD приёмник, источник, количество\_сдвигов |
| Машинный код | 0F АC — SHRD r/m16,r16,imm8 0F AD — SHRD r/m16,r16,CL 0F AC — SHRD r/m32,r32,imm8 0F AD — SHRD r/m32,r32,CL |
| Описание | Сдвиг двойной точности вправо. |
| Действие | Команда сдвигает операнд приемник вправо на число битов, указанных операндом количество\_сдвигов. Операнд источник обеспечивает биты, которые вдвигаются в приемник слева (начиная со старшего бита приемника). Операнд количество\_сдвигов — целое число без знака, которое может быть непосредственным 8-разрядным значением или содержимым регистра CL. |
| Флаги | CF=r OF=?r SF=r ZF=r PF=r AF=? Флаг CF заполняется последним битом, сдвинутым из операнда приемник. Флаг OF устанавливается в 1 для одноразрядного сдвига, если изменился знаковый разряд приемника, иначе он равен нулю. Если операнд количество\_сдвигов равен нулю, то флаги не изменяются. |

**STC**

|  |  |
| --- | --- |
| Синтаксис | STC |
| Машинный код | F9 |
| Описание | Установка флага переноса. |
| Флаги | CF = 1. |

**STD**

|  |  |
| --- | --- |
| Синтаксис | STD |
| Машинный код | FD |
| Описание | Установка флага направления. |
| Флаги | DF = 1. |

**STI**

|  |  |
| --- | --- |
| Синтаксис | STI |
| Машинный код | FB |
| Описание | Установка флага прерывания. |
| Флаги | IF = 1. |

**SUB**

|  |  |
| --- | --- |
| Синтаксис | SUB приемник,источник |
| Машинный код | 2C ib — SUB AL,imm8 2D iw — SUB AX,imm16 2D id — SUB EAX,imm32 80 /5 ib — SUB r/m8,imm8 81 /5 iw — SUB r/m16,imm16 81 /5 id — SUB r/m32,imm32 83 /5 ib — SUB r/m16,imm8 83 /5 ib — SUB r/m32,imm8 28 /r — SUB r/m8,r8 29 /r — SUB r/m16,r16 29 /r — SUB r/m32,r32 2A /r — SUB r8,r/m8 2B /r — SUB r16,r/m16 2B /r — SUB r32,r/m32 |
| Описание | Вычитание. |
| Действие | Вычитание двух целочисленных двоичных операндов. приёмник = приёмник — источник |
| Флаги | OF=r SF=r ZF=r AF=r PF=r CF=r |

**TEST**

|  |  |
| --- | --- |
| Синтаксис | TEST приемник,источник |
| Машинный код | A8 ib — TEST AL,imm8 A9 iw — TEST AX,imm16 A9 id — TEST EAX,imm32 F6 /0 ib — TEST r/m8,imm8 F7 /0 iw — TEST r/m16,imm16 F7 /0 id — TEST r/m32,imm32 84 /r — TEST r/m8,r8 85 /r — TEST r/m16,r16 85 /r — TEST r/m32,r32 |
| Описание | Логическое сравнение. |
| Действие | Команда выполняет операцию логического И над операндами. Сами операнды не изменяются. В зависимости от результата устанавливаются флаги SF, ZF и PF, флаги OF и CF сбрасываются. |
| Флаги | OF=0 SF=r ZF=r AF=? PF=r CF=0 |

**XCHG**

|  |  |
| --- | --- |
| Синтаксис | XCHG операнд1,операнд2 |
| Машинный код | 90+rw — XCHG AX,r16 90+rw — XCHG r16,АХ 90+rd XCHG EAX,r32 90+rd XCHG r32,EAX 86 /r — XCHG r/m8,r8 86 /r — XCHG r8,r/m8 87 /r — XCHG r/m16,r16 87 /r — XCHG r16,r/m16 87 /r — XCHG r/m32,r32 87 /r — XCHG r32,r/m32 |
| Описание | Обмен. |
| Действие | Обмен значениями между операндами. |
| Флаги | Не изменяются. |

**XOR**

|  |  |
| --- | --- |
| Синтаксис | XOR приемник,источник |
| Машинный код | 34 ib — XOR AL,imm8 35 iw — XOR AX,imm16 35 id — XOR EAX,imm32 80 /6 ib — XOR r/m8,imm8 81 /6 iw — XOR r/m16,imm16 81 /6 id — XOR r/m32,imm32 83 /6 ib — XOR r/m16,imm8 83 /6 ib — XOR r/m32,imm8 30 /r — XOR r/m8,r8 31 /r — XOR r/m16,r16 31 /r — XOR r/m32,r32 32 /r — XOR r8,r/m8 33 /r — XOR r16,r/m16 33 /r — XOR r32,r/m32 |
| Описание | Логическое исключающее ИЛИ. |
| Действие | Команда выполняет операцию логического исключающего ИЛИ над соответствующими парами битов операндов. приёмник = приёмник XOR источник |
| Флаги | OF=0 SF=r ZF=r AF=? PF=r CF=0 |