Расширение знака в AX, DX, EAX,EDX

CBW	AX:=AL
CWD	DX:AX:=AX
CWDE	EAX := AX
CDQ	EDX:EAX := EAX

Команды пересылки MOV OP1, OP2

OP1	OP2
r8 r16 r32	r8, m8, i8 r16, m16, i16 r32, m32, i32
m8 m16 m32	r8, i8 r16, i16 r32, i32

Команда обмена XCHG OP1, OP2

OP1	OP2
r8 r16 r32	r8, m8 r16, m16 r32, m32
m8 m16 m32	r8 r16 r32

Пересылка с расширением размера

доп. нулями **MOVZX OP1, OP2;** доп. знаковым битом **MOVSX OP1, OP2;**

OP1	OP2
r16	r8, m8
r32	r8, m8, r16, m16

Загрузка исп. адреса LEA OP1, OP2
Например, LEA EAX,[ESP+4*EBX+5]

Арифметические команды КОП OP1,OP2:
ADD OP1, OP2 - OP1 := OP1 + OP2
SUB OP1, OP2 - OP1 := OP1 – OP2
CMP OP1, OP2 - только для флагов
ADC OP1, OP2 - OP1 := OP1 + OP2 + CF
SBB OP1, OP2 - OP1 := OP1 – OP2 – CF

OP1	OP2
r8 r16 r32	r8, m8, i8 r16, m16, i16 r32, m32, i32
m8 m16 m32	r8, i8 r16, i16 r32, i32

Операции меняют ZF, SF, CF, OF

Деление и умножение без знака (MUL, DIV)
Деление и умножение знаковое (IMUL, IDIV)

MUL OP2 **IMUL OP2**	AX:=AL*OP2 DX:AX := AX*OP2 EDX:EAX := EAX*OP2
DIV OP2 **IDIV OP2**	AL:=AX/OP2; AH:=AX%OP2 AX := DX:AX / OP2; DX – ост EAX := EDX:EAX / OP2; EDX := EDX:EAX % OP2
OP1	OP2
Не указывается	r8, r16, r32 m8, m16, m32

CF=OF=0 если результат одинарный
CF=OF=1 если результат двойной точности
Внимание, OP2 – не может быть i8, i16, i32 !

IMUL OP1, OP2

OP1	OP2
r16	r16, m16, i16
r32	r32, m32, i32

Только знаковые, результат одинарной точности

IMUL OP1, OP2, OP3

OP1	OP2	OP3
r16	r16, m16	i16
r32	r32, m32	i32

Только знаковые, результат одинарной точности

Однооперандные команды

NEG OP1	OP1 := –OP1
INC OP1	OP1 := OP1 + 1
DEC OP1	OP1 := OP1 – 1
NOT OP1	побитовая инверсия
	OP1
	r8, m8, r16, m16, r32, m32

INC и DEC не меняют CF

Логические команды (побитовые)

AND OP1, OP2	OP1 := OP1 & OP2
TEST OP1, OP2	OP1 & OP2 → флаги
OR OP1, OP2	OP1 := OP1 | OP2
XOR OP1, OP2	OP1 := OP1 ^ OP2

Операнды и флаги такие же как и в арифметических командах

Команды сдвига: КОП OP1,OP2

SHR OP1, OP2 - простой сдвиг,
SHL OP1, OP2 вдвигается 0

SAR OP1, OP2 - арифметический сдвиг,
SAL OP1, OP2 при SAR вдвигается знак

ROR OP1, OP2 - циклический сдвиг
ROL OP1, OP2

RCR OP1, OP2 - циклический сдвиг
RCL OP1, OP2 через CF

OP1	OP2
r8, m8	CL, i8
r16, m16	CL, i8
r32, m32	CL, i8

CF = значение последнего сдвинутого бита

SHRD OP1, OP2, OP3 - двойной сдвиг
SHLD OP1, OP2 OP3
сдвигается OP1:OP2 как единый операнд

OP1	OP2	OP3
r16	r16, m16	CL, i8
r32	r32, m32	CL, i8

В операции CL берется по модулю 32.

Адресация (забыть про сегм. регистры)

<имя переменной>, ds:[i32]
[BASE] + [i*INDEX] + [i32]
BASE – любой 32-бит. регистр (EAX .. ESP)
INDEX – любой 32-бит. регистр кроме ESP
i = 1, 2, 4, 8. MOV EAX, [EAX+4*EDX+20]

Команды перехода

Безусловный переход

JMP i32 переход по метке
JMP r32 переход по регистру
JMP m32 косвенный переход

Условный переход: J Метка**

JE, JZ =
JNE, JNZ <>
JG, JNLE > знаковый
JGE, JNL >= знаковый
JL, JNGE < знаковый
JLE, JNG <= знаковый
JA, JNBE > без знака
JAE, JNB, JNC >= без знака
JB, JNAE, JC < без знака
JBE, JNA <= без знака
JS по минусу
JNS по плюсу
JO по OF
JNO по not OF
JCXZ по CX=0
JECXZ по ECX=0

Организация цикла
LOOP / LOOPE / LOOPNE Метка
Внимание, если изначально ECX=0, то цикл выполнится 2^{32} раз. Использовать JECXZ до цикла!

CF: CLC, STC, CMC CF:=0, CF:=1, CF:= not CF;

Процедуры
CALL i32 вызов по имени процедуры
CALL r32 вызов по регистру
CALL m32 косвенный вызов
RET
RET n
PUSH r/m/16/32, PUSHF, PUSHA, PUSHAD
POP r/m/16/32 , POPF, POPA, POPAD

Строковые команды
MOVSB, MOVSW, MOVSD пересылка
LODSB, LODSW, LODSD загрузка
STOSB, STOSW, STOSD запись
SCASB, SCASW, SCASD сканирование
CMPSB, CMPSW, CMPSD сравнение
CLD, STD – установка флага Direction

префиксы перед строковыми командами
REP, REPE, REPNE (REPZ, REPNZ)

Определение данных
DB, DW, DD, DQ
спецификатор **5 dup (?)** лля повторения

ДОПОЛНИТЕЛЬНЫЕ КОМАНДЫ
(полезно знать об их существовании)
LAHF – AH = Flags – загрузка флагов на AH
SAHF – Flags = AH – пересылка флагов из AH
XLAT – AL := mem[EBX+AL]

Пересылка при выполнении условия
CMOVcc (только 16/32 битовые)
SETcc r/m8– условная установка байта
Например, CMOVLE EAX,[EBX]; SETG CH

BSWAP r16/ r32 – реверс байтов

Операции с отдельными битами
BSF OP1,OP2 – номер первого единичного бита
BSR OP1,OP2– номер последнего единичного бита
OP1 = r16/ r32; OP2=r16/ r32/ m16/ m32

BT OP1,OP2 – тестирование бита
BTC OP1,OP2 – тестирование и очистка
BTS OP1,OP2 – тестирование и установка единицы
BTR OP1,OP2 – тестирование и инвертирование
OP1=r16/ m16/ r32/ m32; OP2 = r16/ r32/ i8

XADD OP1, OP2 = XCHG + ADD
CMPXCHG – сравнить и обменять (удобно для реализации семафоров)