Краткий список изучаемых команд

Требования к операндам

1) одинаковый размер
2) два операнда из ОП не допускаются

Везде: Получатель (OP1) – r, m
Источник (OP2) – r, m, i

Пересылки

```
MOV    OP1, OP2     OP1:=OP2
XCHG   OP1, OP2     OP1↔OP2

MOVZX  OP1, OP2     OP1:= OP2, беззнаковое
MOVSX  OP1, OP2     OP1:= OP2, знаковое
```

Операнды OP1–r, OP2–r, m меньшего размера

Загрузка исп. адреса

```
LEA OP1, OP2
```

Операнды OP1–r32, OP2–m
Пример: `LEA EAX, [ESP+4*EBX+5]`

Расширение знака в AX, DX, EAX, EDX

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

Арифметические

Флаги CF, OF, SF, ZF

```
ADD  OP1, OP2     OP1:= OP1 + OP2
ADC  OP1, OP2     OP1:= OP1 + OP2 + CF
SUB  OP1, OP2     OP1:= OP1 – OP2
SBB  OP1, OP2     OP1:= OP1 – OP2 – CF
CMP  OP1, OP2     OP1 – OP2
NEG  OP1          OP1 :=0 – OP1
```

Флаги OF, SF, ZF

```
INC OP1           OP1 := OP1 + 1
DEC OP1           OP1 := OP1 – 1
```

Умножение/деление

Операнд OP1 - не i

	Зависит от размера OP1	
MUL OP1	*byte*	*AX:= AL*OP1*
IMUL OP1	*word*	*DX:AX := AX*OP1*
	dword	*EDX:EAX := EAX*OP1*

	Зависит от размера OP1
	byte
DIV OP1	*AX / OP1, div →AL, mod →AH*
IDIV OP1	*word*
	DX:AX / OP1, div →AX, mod →DX
	dword
	EDX:EAX / OP1, div →EAX, mod →EDX

Побитовые

Логические команды

Флаг ZF

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

Не устанавливает флаги

```
NOT  OP1          OP1:= not OP1
```

Сдвиги

Флаг CF

SHR OP1, OP2	*логические*
SHL OP1, OP2	
ROR OP1, OP2	*циклические*
ROL OP1, OP2	
RCR OP1, OP2	*циклические через CF*
RCL OP1, OP2	

Операнд OP2 – i, CL
CF= последний выдвинутый бит

Команды перехода

Безусловные

```
JMP метка      прямой
JMP r32/m32    косвенный
```

Условные (все прямые)

```
  JE  метка      =
  JNE метка      ≠
```

числа со знаком		числа без знака
JL метка	<	JB метка
JLE метка	≤	JBE метка
JG метка	>	JA метка
JGE метка	≥	JAE метка

LOOP метка	*1. ECX:=ECX-1* *2. if ECX ≠0 goto метка*
JECXZ метка	*ECX=0*

Оба перехода КОРОТКИЕ

```
  JC  метка      CF=1
  JNC метка      CF=0
  JZ  метка      ZF=1
  JNZ метка      ZF=0
```

Синонимичные названия,
используются с побитовыми командами

Работа со стеком

```
PUSH OP2       в стек
POP  OP1       из стека
```

Операнды 32 разряда

```
PUSHFD         флаги в стек
POPFD          восстановить
```

Процедуры

```
CALL метка
RET
RET OP         операнд OP – i
```

Строковые команды

src – [ESI], *dst* – [EDI]
acc – AL, AX, EAX
Три вида команды:
<ст.ком>B, <ст.ком>W, <ст.ком>D
Действие: 1. опер.; 2. сдвиг указат.
Направление сдвига: DF

LODS	*acc:= src*	
STOS	*dst:= acc*	
MOVS	*dst:= src*	
SCAS	*acc – dst*	*CF, OF, SF, ZF*
CMPS	*src – dst*	*CF, OF, SF, ZF*

Управление DF

CLD	*DF:=0*
STD	*DF:=1*

Префиксы

REP	*1. ECX=0? Да - goto конец* *2. операция, сдвиг указат.* *3. ECX:=ECX-1* *4. goto 1*
REPE REPNE	*1. ECX=0? Да, goto конец* *2. операция, сдвиг указат.* *3. ECX:=ECX-1* *4. '=' ('≠')? Нет - goto конец* *5. goto 1*

Адресация (забыть про сегм. регистры)

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

Определение данных

DB, DW, DD, DQ
спецификатор **5 dup (?)** для повторения