

# Основные команды ассемблера

Пересылки  
данных

Арифметические

Логические

Передачи  
управления

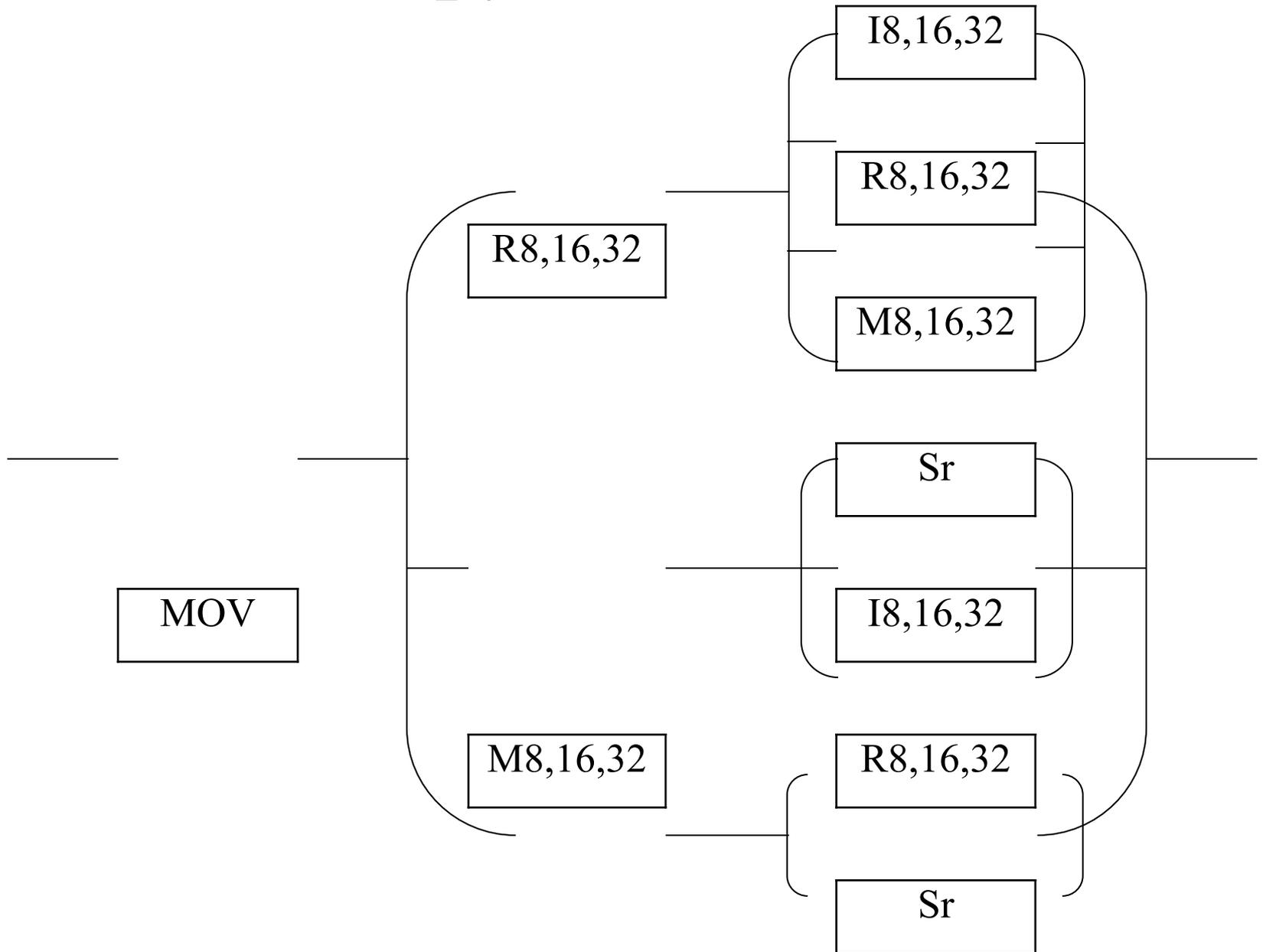
Обработки  
цепочек

Управления  
работой ЦП

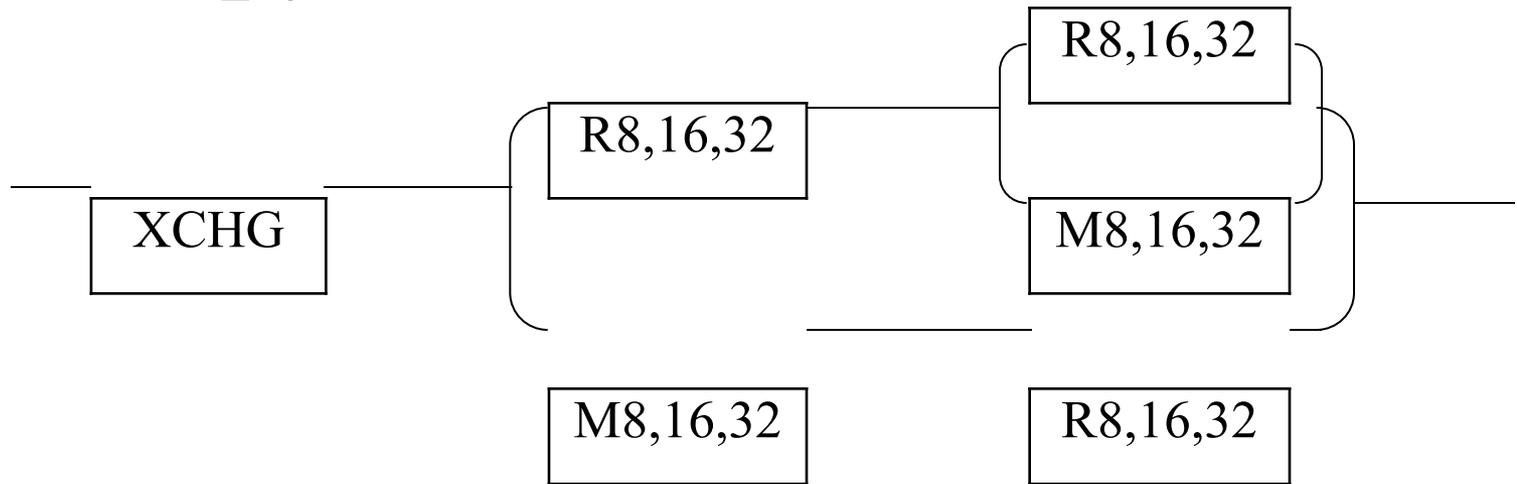
# Команды пересылки данных

Общего назначения	Работы с адресами	Работы со стеком	Преобразования данных
Mov	Lea	Puch	Xlat
Xchg	Lss	Pop	
	Lds	Pucha	
	Les	Popa	
	Lfs	Puchf	
	Lgs	Popf	

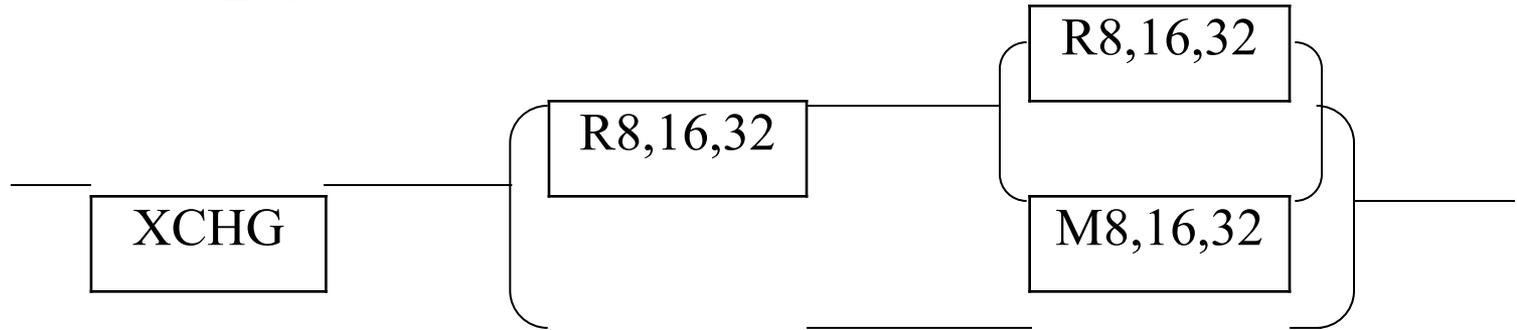
# Инструкция MOV



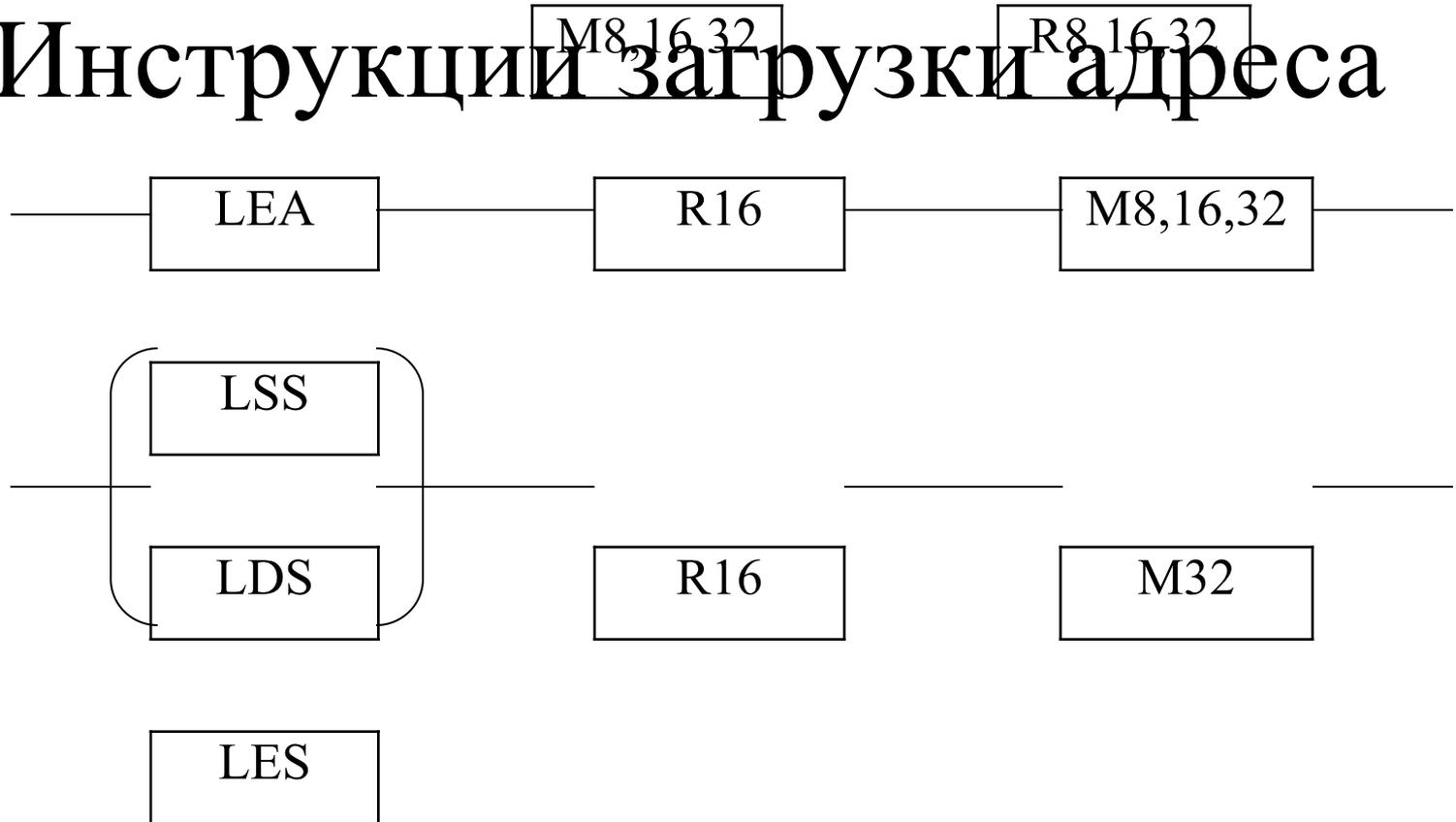
# Инструкция обмена данными



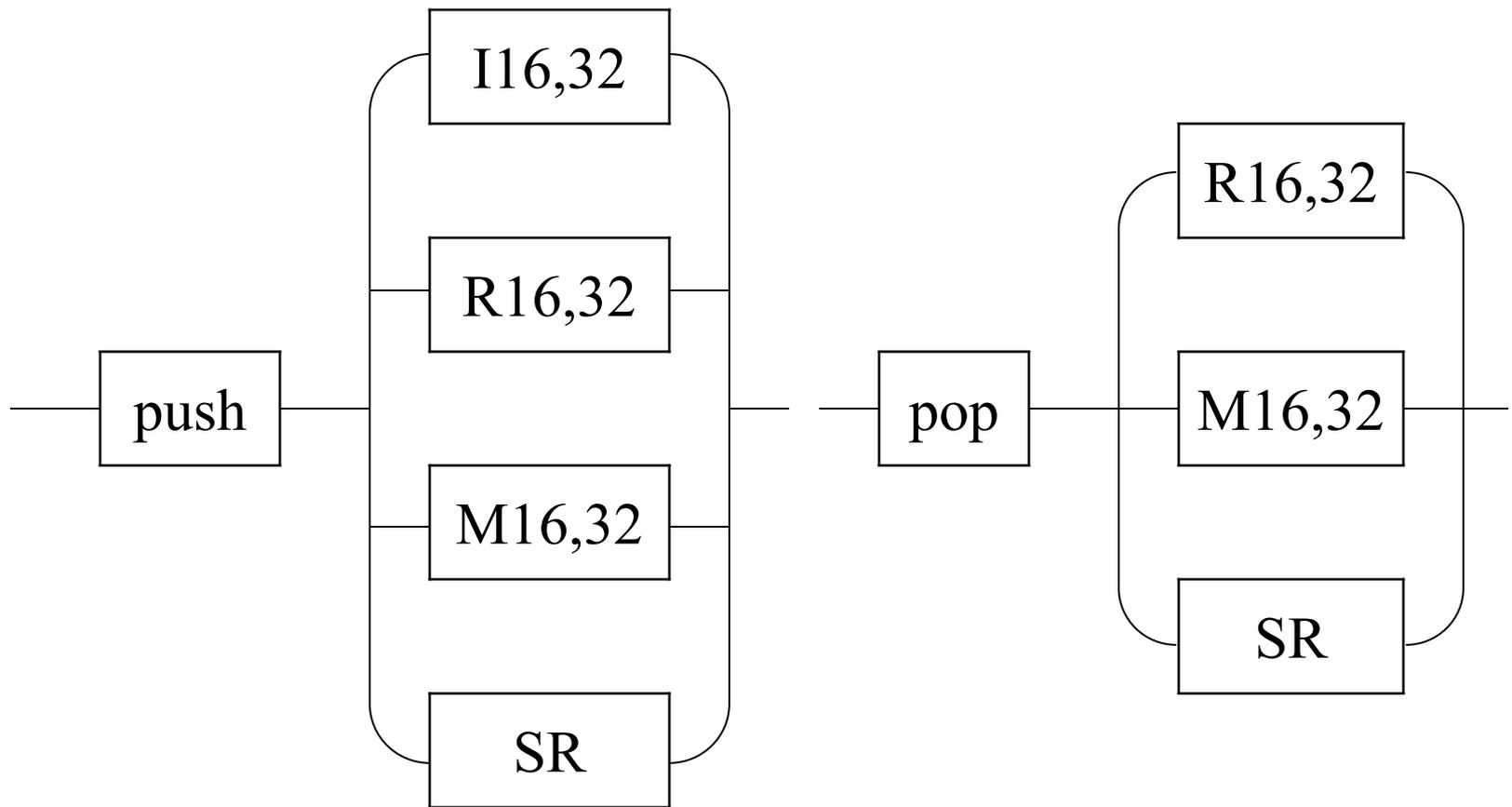
# Инструкция обмена данными



# Инструкции загрузки адреса



# Инструкции работы со стеком



# Инструкция перекодировки xlat

Таблица перекодировки

BX

AL

	0
	1
	2
	3
	4
	...
...	12
	13
	14
	15

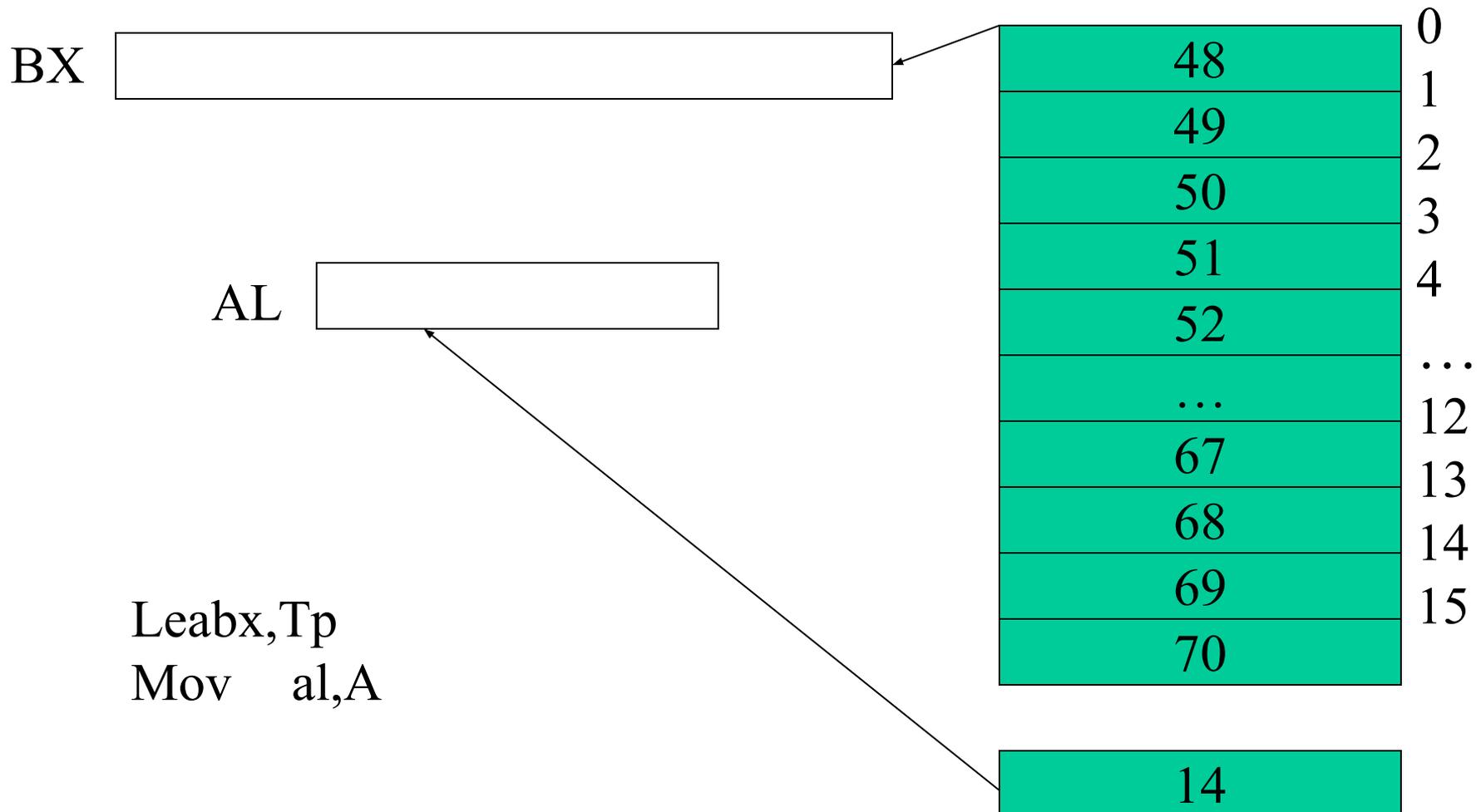
Tr db '0123456789ABCDEF'

A db 14

16-е число

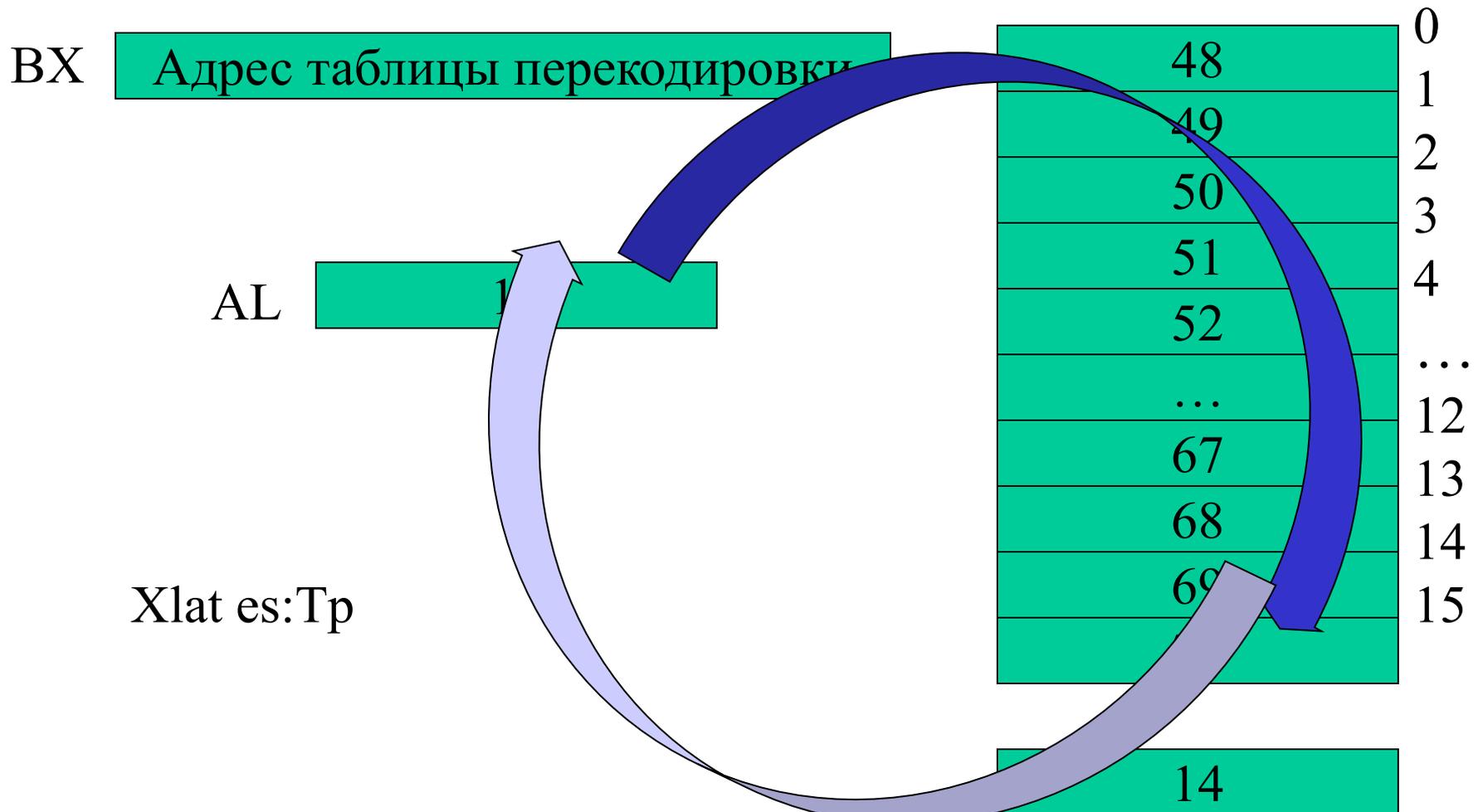
# Инструкция перекодировки xlat

Таблица перекодировки



# Инструкция перекодировки xlat

Таблица перекодировки



# Инструкция перекодировки xlat

Таблица перекодировки

ВХ

Адрес таблицы перекодировки

AL

‘E’

48	0
49	1
50	2
51	3
52	4
...	...
67	12
68	13
69	14
70	15

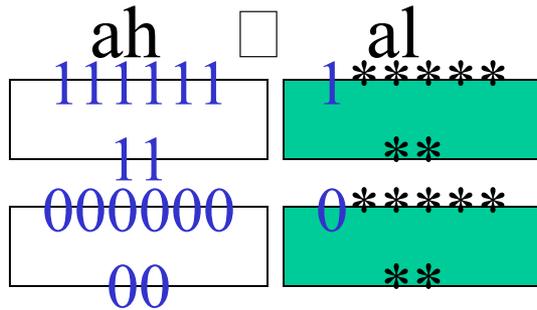
14

# Арифметические команды

Преобразования типов	Двоичной арифметики		Десятичной арифметики	Прочие
Cbw	Add	Imul	Aaa	Cmp
Cwd	Adc	Mul	Daа	Setcc
Cwde	Inc	Idiv	Aas	
Cdq	Sub	Div	Das	
Movsx	Sbb	Neg	Aam	
Movzx	Dec		Aad	

# Преобразование

## ❖ Байта в слово



cbw

## ❖ Слова в двойное слово

– Cwd: ax □ dx

– Cwde: ax □ eax

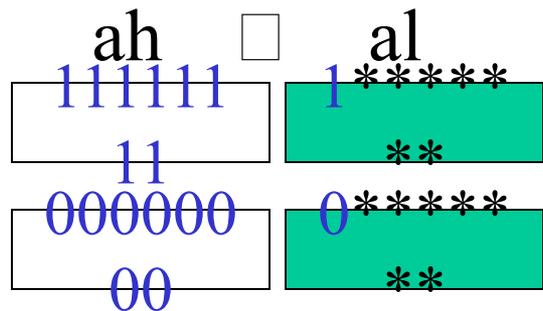
## ❖ Двойного слова в учетверенное

– Cdq: eax □ edx

# Пересылка

# Преобразование

## ❖ Байта в слово



`cbw`

## ❖ Слова в двойное слово

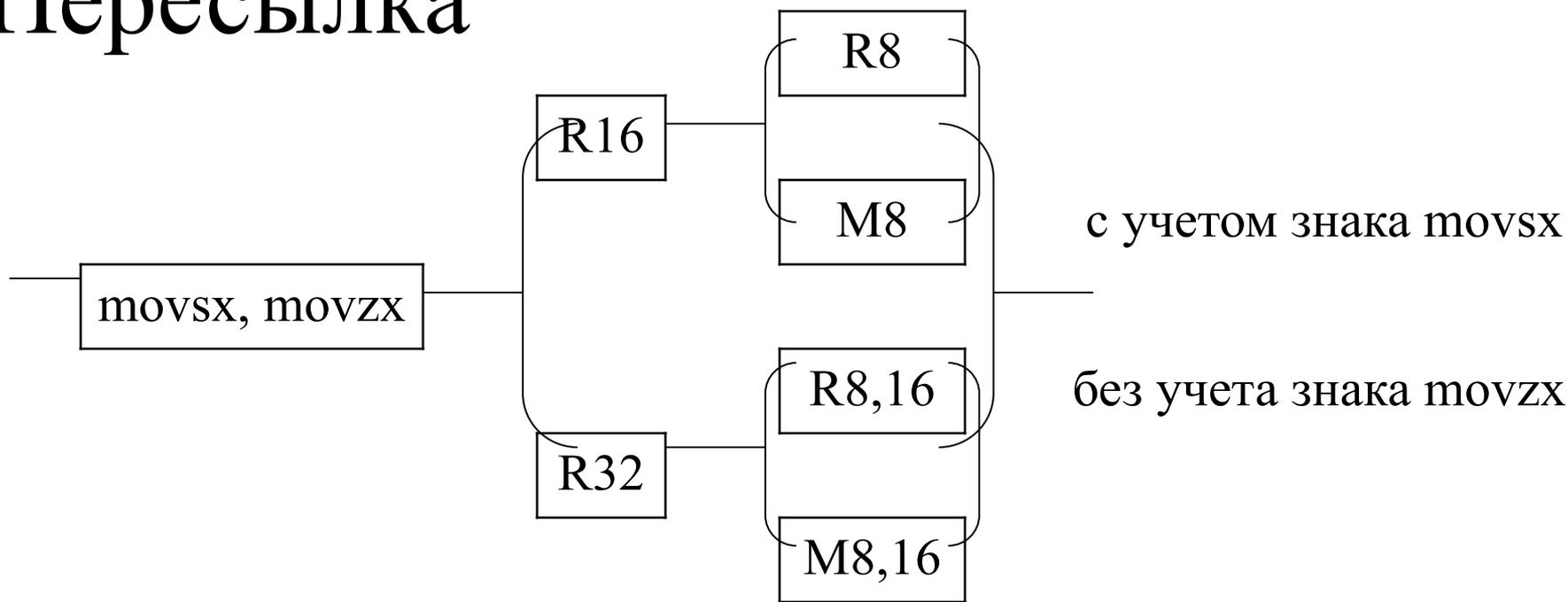
– `Cwd:` `ax` □ `dx`

– `Cwde:` `ax` □ `eax`

## ❖ Двойного слова в учетверенное

– `Cdq:` `eax` □ `edx`

# Пересылка



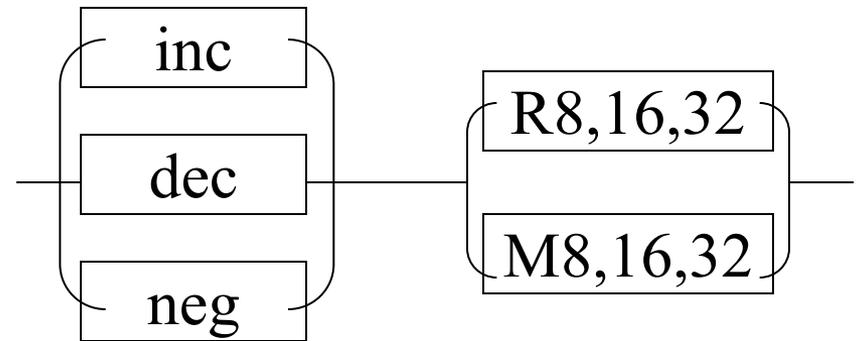
# Двоичная арифметика

Inc – увеличение на 1<sup>\*)</sup>

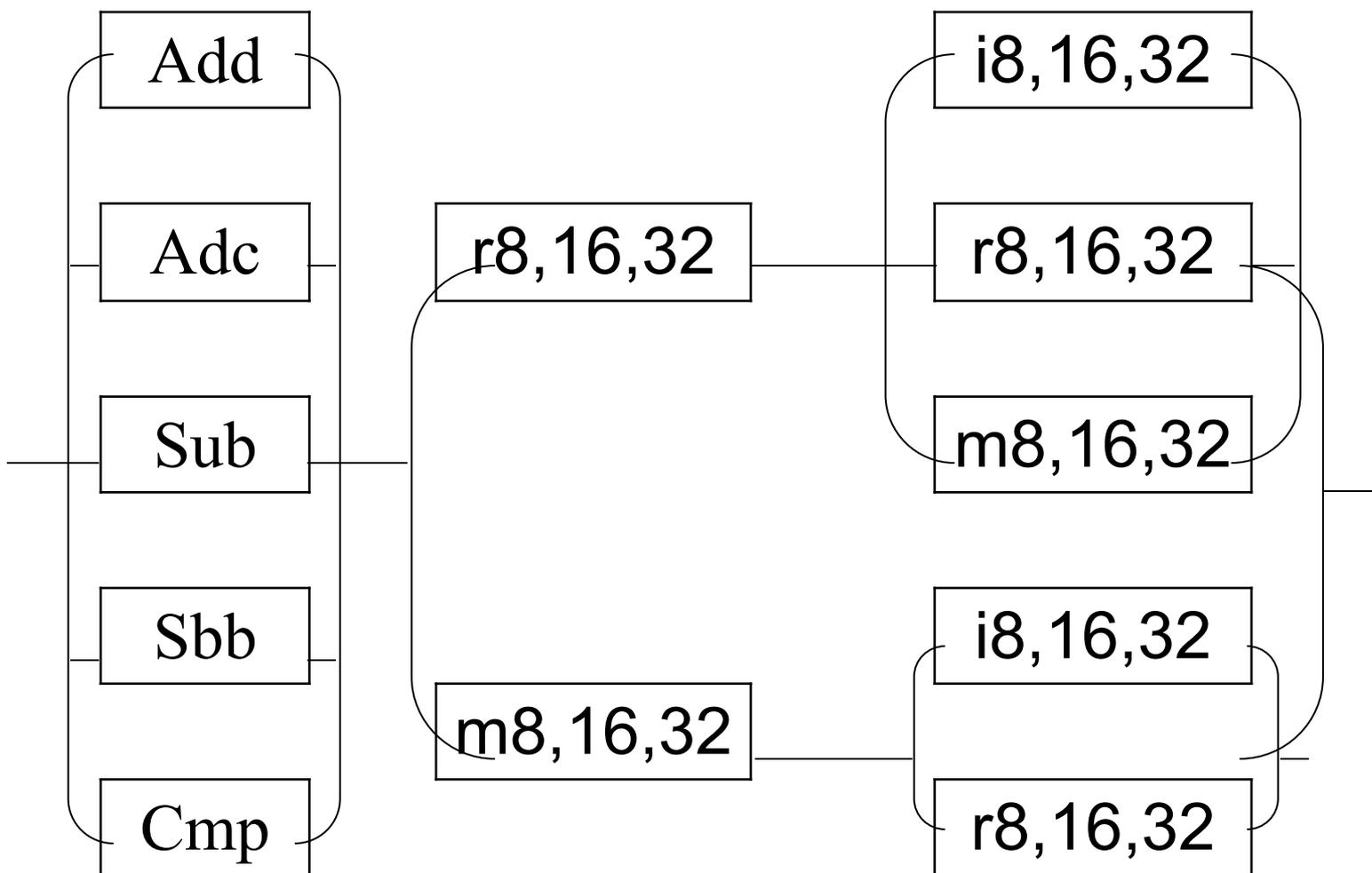
Dec – уменьшение на 1<sup>\*)</sup>

Neg – смена знака

<sup>\*)</sup> Не изменяет флага cf.



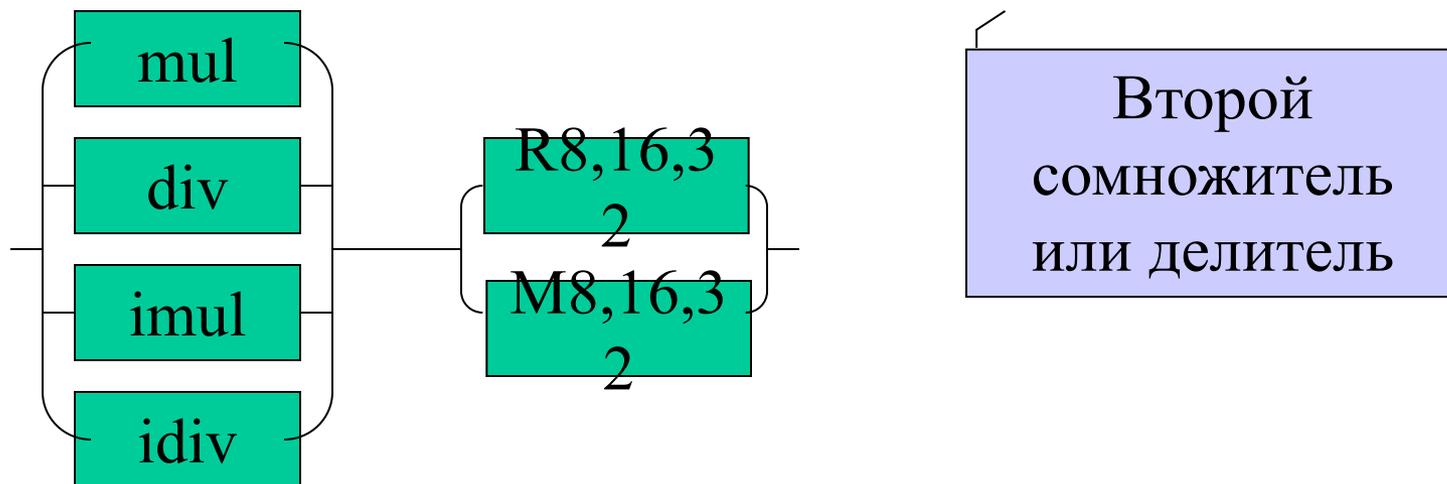
# Сложение, вычитание сравнение



# Пример

	cf	ah	al
Mov ax,128			
		00000000	10000000
		0	00
Add al,128	1	00000000	00000000
		0	00
Adc ah,128	0	10000000	00000000
		1	00
Adc ah,128	1	00000000	00000000
		1	00

# Умножение, деление



Тип операнда	Первый сомножитель	Результат	Делимое	Результат	
				Частное	Остаток
8	al	ax	ax	al	ah
16	ax	dx:ax	dx:ax	ax	dx
32	eax	edx:eax	edx:eax	eax	edx

Делитель 0 или частное велико – исключительная ситуация

# Десятичная арифметика

Имя	Содержание
Для неупакованных BCD чисел в регистре al	
Aaa	ASCII-коррекция после сложения
Aas	ASCII-коррекция после вычитания
Aam	ASCII-коррекция после умножения
Aad	ASCII-коррекция перед делением
Для упакованных BCD чисел в регистре al	
Daа	Десятичная коррекция после сложения
Das	Десятичная коррекция после вычитания

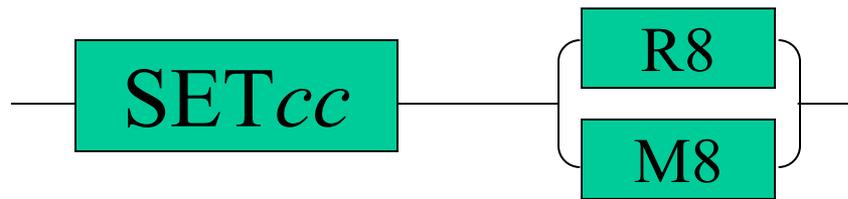


# AAM и AAD: примеры

- `mov al,9`                   \*\* 09
- `mov bl,9`                   \*\* 09
- `mul bl`                    00 51
- `aam`                      08 01
- `mov al,99`                00 63
- `aam`                      09 09
- `mov ax,0703h`            07 03
- `aad`                      00 49
- `mov bl,9`                00 09
- `div bl`                   01 08
- `mov ax,'99'`             39 39
- `aad`                      00 73
- `sub al,10h`             00 10
- 00 63

# ДAA и DAS: примеры

- mov ax,44h      00 44  
  add al,37h      00 37  
  daa            00 7B     $al_1 > 9, af=0$   
                 00 81     $af = 1$   
  
                 00 57
- Mov ax,57h      00 57  
  sub al,19h      00 19  
  das            00 3E     $af = 1$   
  
                 00 38
- Mov ax,88h      00 88  
  add al,12h      00 12  
  daa            00 9A     $al_1 > 9$   
                 00 00     $af = cf = 1$   
  
                 00 88
- Mov ax,88h      00 88  
  sub al,99h      00 99  
  das            00 EF     $af = cf = 1$   
  
                 00 89     $af = cf = 1$

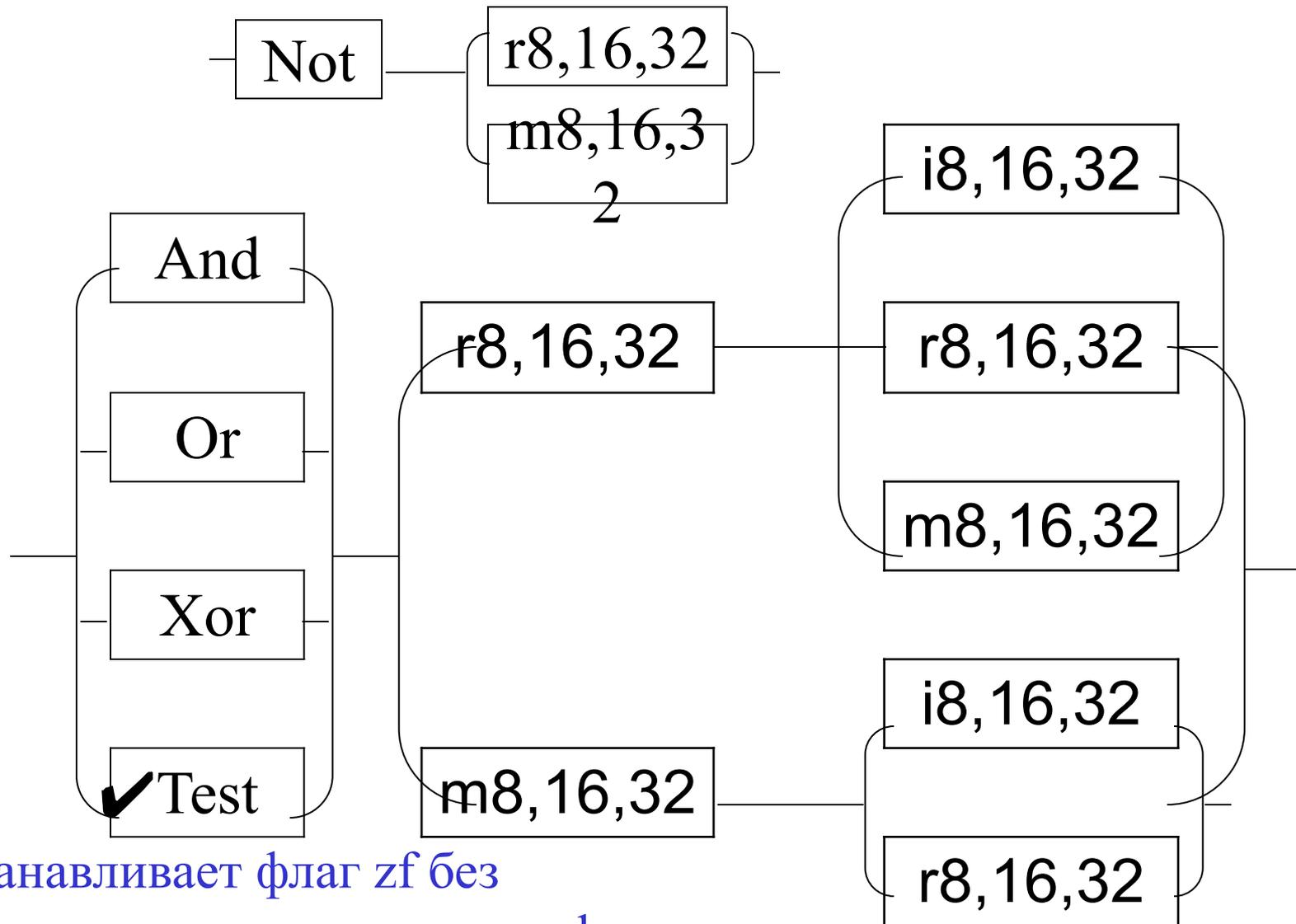


Команда	Условие	Команда	Условие
seta/setnbe	cf=0 и zf=0	setle/setng	zf=1 или sf≠of
setae/setnb	cf=0	setnc	cf=0
setb/setnae	cf=1	setne/setnz	zf=0
setbe/set	cf=1 или zf=1	setno	of=0
setc	cf=1	setnp/setpo	pf=0
sete/setz	zf=1	setns	sf=0
setg/setnle	zf=0 или sf=of	seto	of=1
setge/setnl	sf=of	setp/setpe	pf=1
setl/setnge	sf≠of	sets	sf=1

# Логические команды

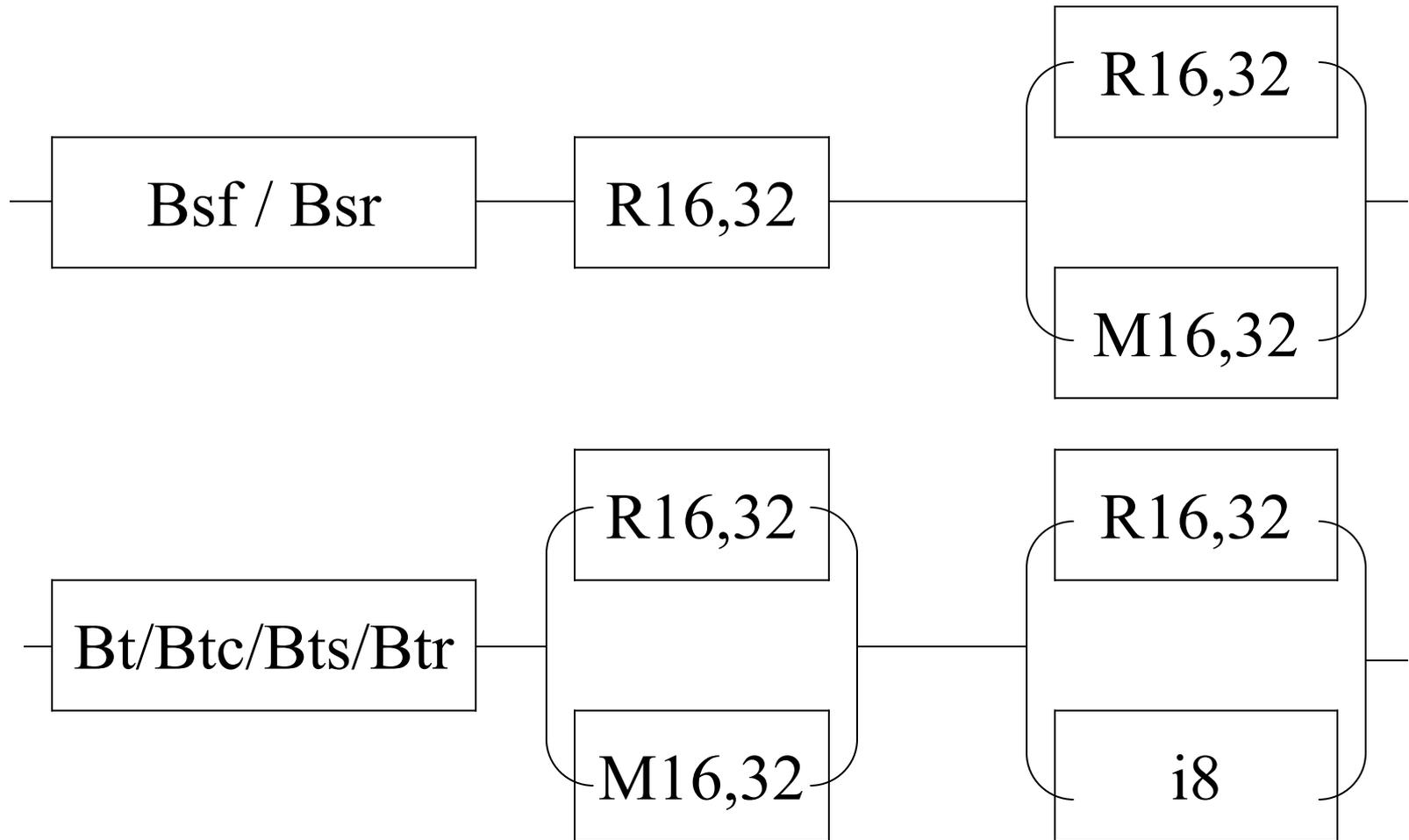
Логические	Обработки бит	Сдвига
And	Bsf	Sar
Or	Bsr	Sal
Xor	Bt	Shl
Not	Btc	Shr
Test	Btr	Shld
	Bts	Shrd
		Rcl
		Rcr
		Rol
		Ror

# Побитовые булевские операции



✓ устанавливает флаг zf без формирования результата and

# Операции с битами



# Примеры

- A dw0000h  
B dw0110h  
...

.386

bsf ax,A

zf=1

bsf ax,B

ax=4, zf=0

bsr ax,B

ax=8, zf=0

bt b,ax

cf=1

btc b,ax

cf=1, b=0010h

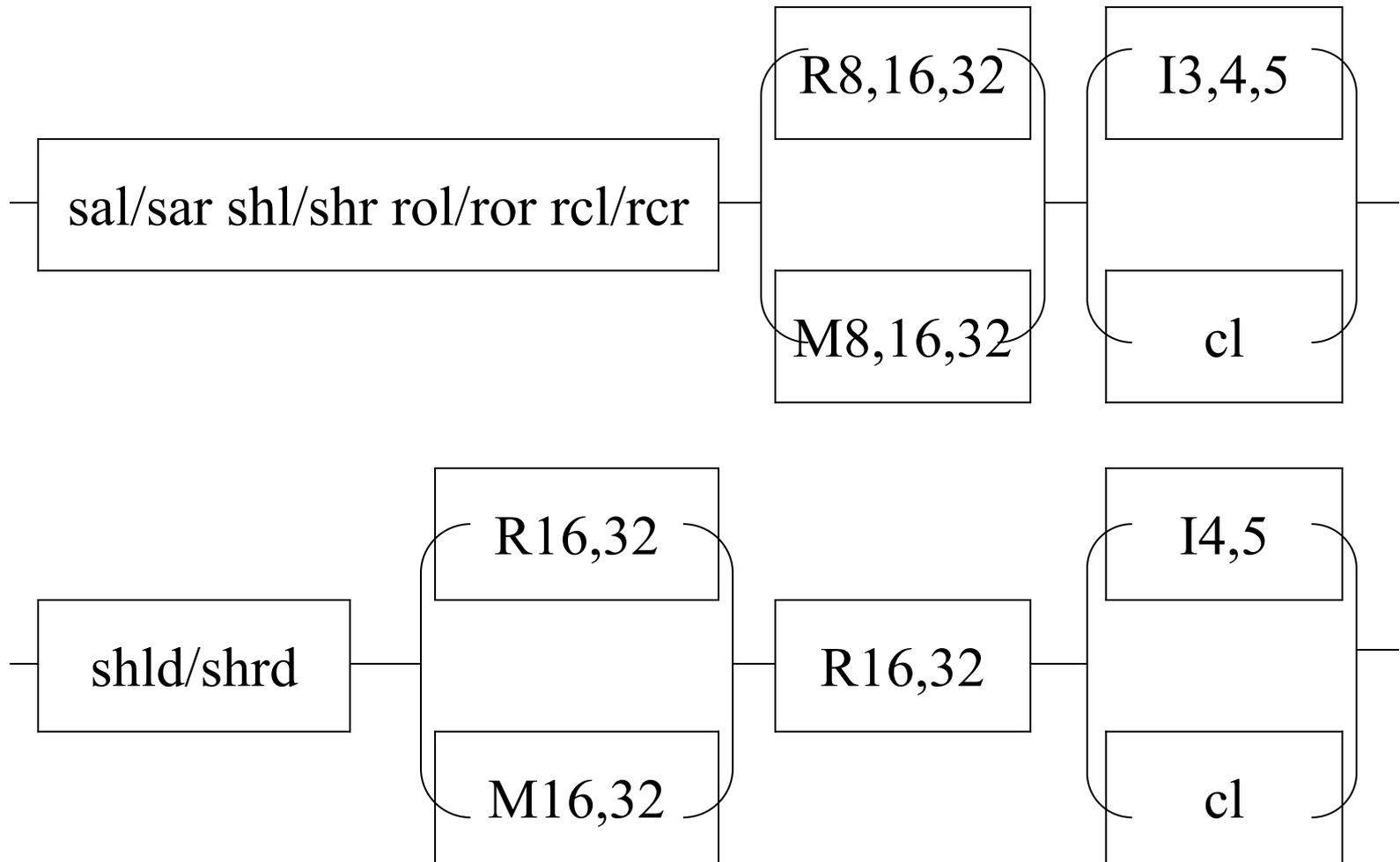
btr b,4

cf=1, b=0000h

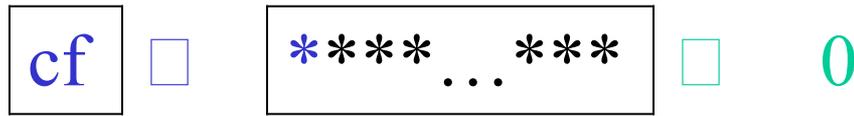
bts b,1

cf=0, b=0002h

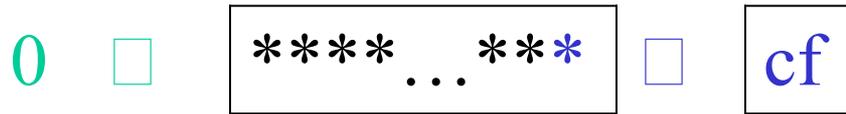
# Операции сдвига



sal,shl



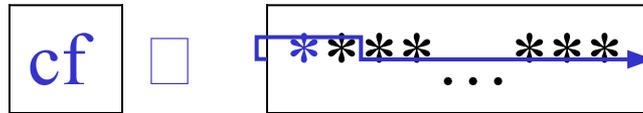
shr



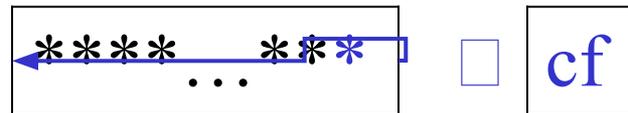
sar



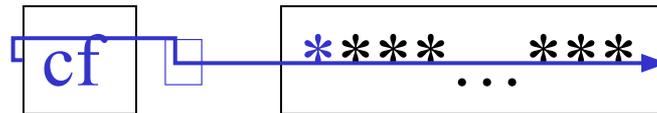
rol



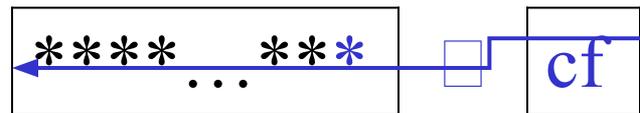
ror



rcl



rcr



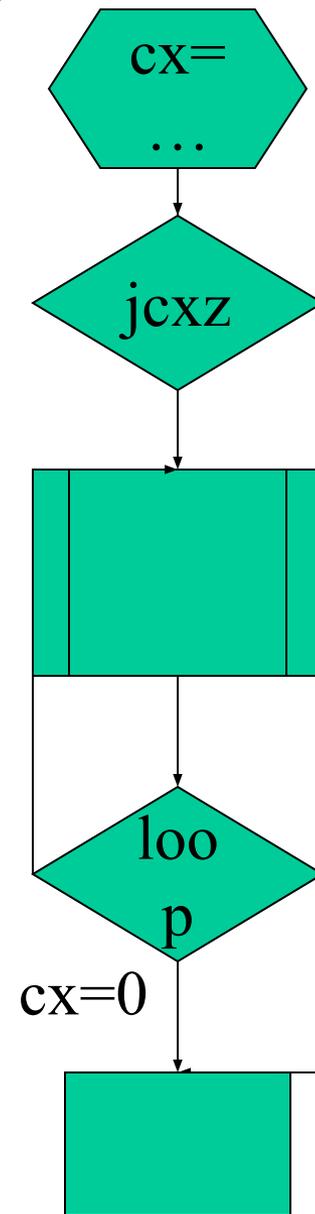
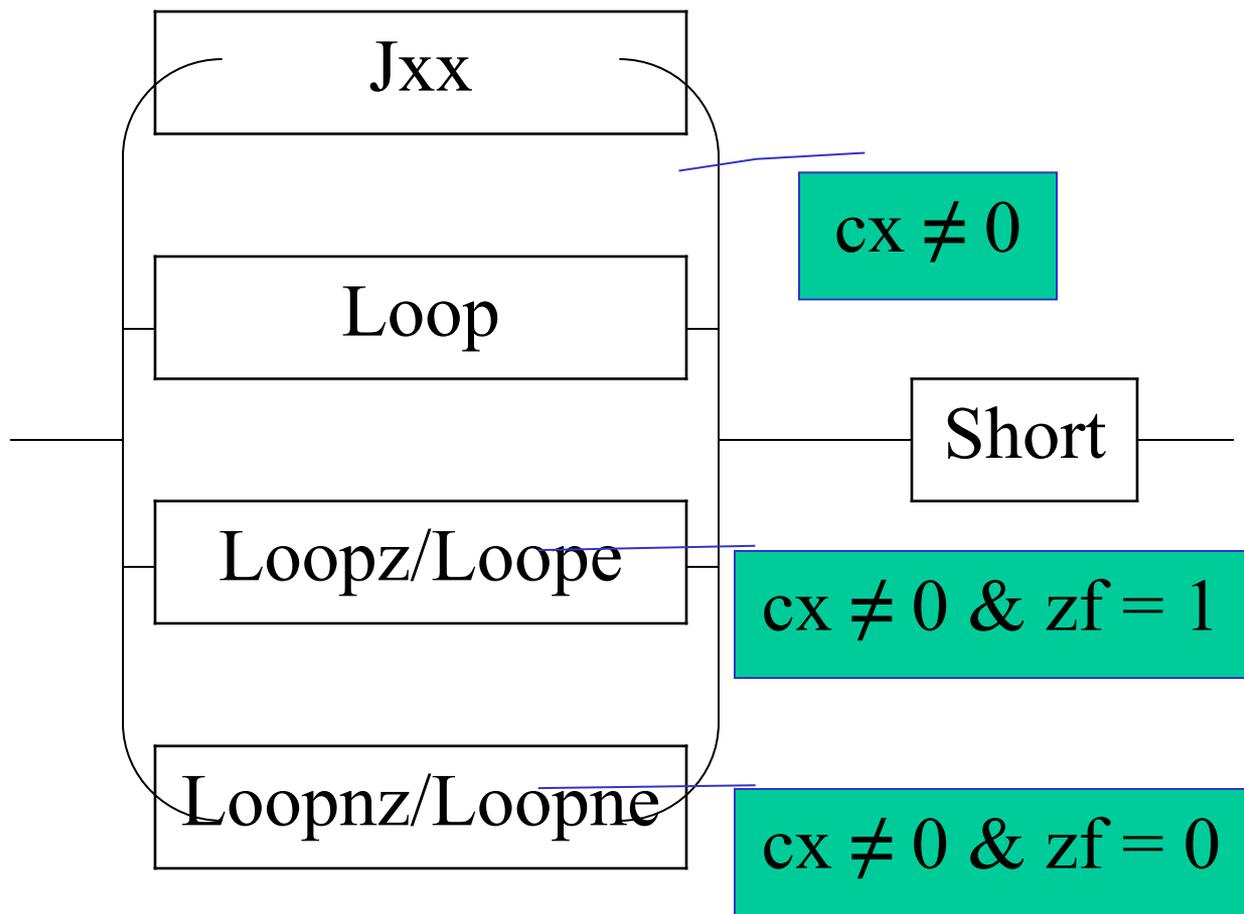
# Команды передачи управления

Безусловная	Взаимодействия с процедурами	Условные			Циклы
Jmp	Call	Jl=Jnge	Jc	Jnc	Loop
	Ret	Jle=Jng	Jp	Jnp	Loope
		Jg=Jnle	Jz	Jnz	Loopz
	Int	Jge=Jnl	Js	Jns	Loopne
	Iret	Jb=Jnae	Jo	Jno	Loopnz
		Jbe=Jna			
		Ja=Jnbe	Jcxz		
		Jae=Jnb	Jecxz		

«выше» - «ниже» - для чисел без знака

«больше» - «меньше» - для чисел со знаком

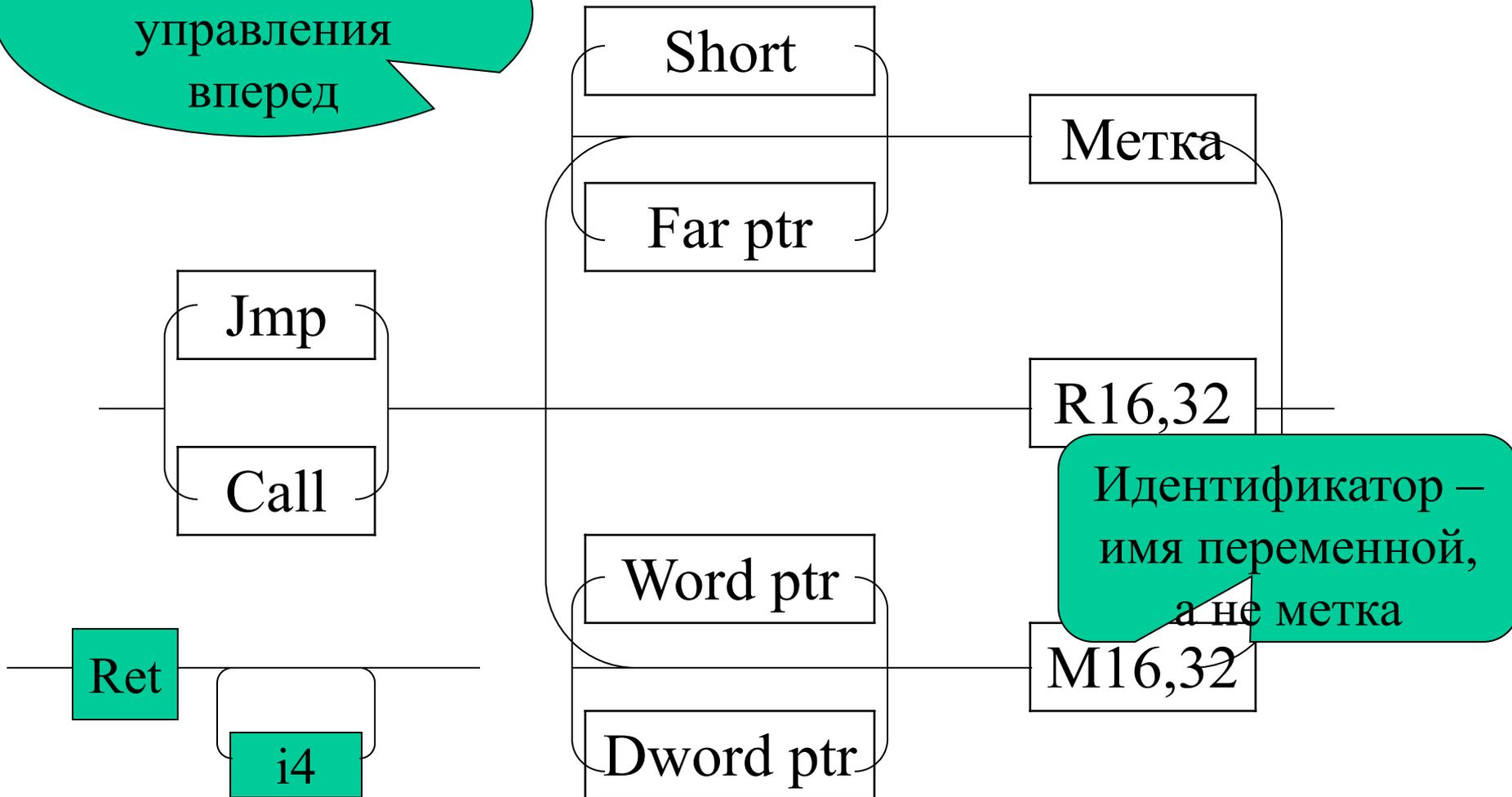
# Операторы перехода по условию



# Инструкции передачи

## управления

При передаче  
управления  
вперед



Идентификатор –  
имя переменной,  
а не метка

# Инструкции работы с прерываниями

- `Int i8` – вызов процедуры обслуживания прерывания с номером, заданным операндом команды:

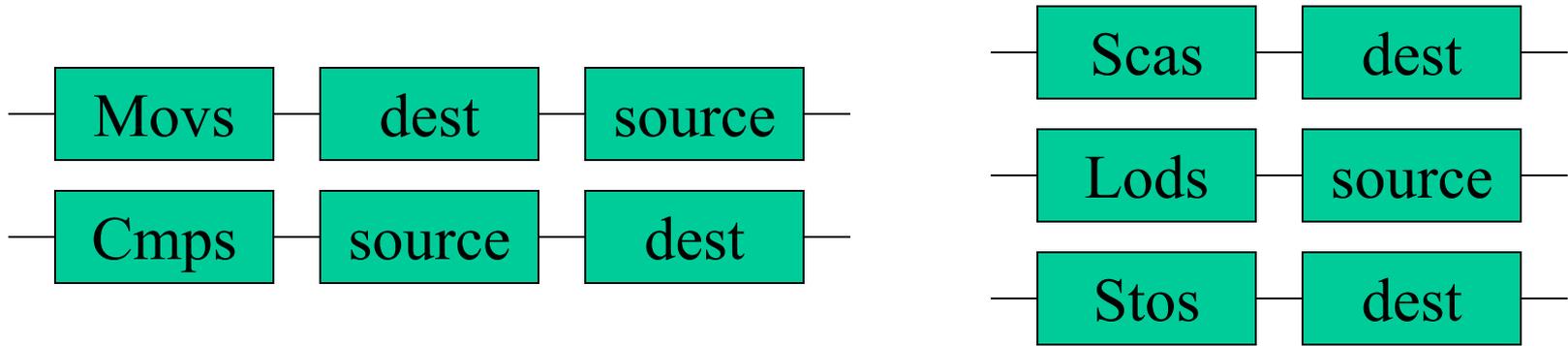
```
pushf          cli  
push cs        jmp ...  
push ip
```

- `Iret` – возврат из программы обработки прерывания в прерванную программу:

```
pop ip         popf  
pop cs        jmp ...
```

# Команды работы с цепочками

Пересылка	Сравнение	Сканирование	Загрузка	Сохранение
Movs	Cmps	Scas	Lods	Stos
Movsb	Cmpsb	Scasb	Lodsb	Stosb
Movsw	Cmpsw	Scasw	Lodsw	Stosw
Movsd	Cmpsd	Scasd	Lodsd	Stosd
Rep	Repz	Repe		
	Repnz	Repne		



	Источник	Приемник	Результат
Cmps	Ds:si	Es:di	Cf, zf, si, di $\pm$ 1
Lods	Ds:si	Eax/ax/al	Ds:[si] $\square$ al, si $\pm$ 1
	Приемник	Источник	Результат
Movs	Es:di	Ds:si	[si] $\square$ [di], si, di $\pm$ 1
Scas	Es:di	Eax/ax/al	Zf, di $\pm$ 1
Stos	Es:di	Eax/ax/al	al $\square$ [di], di $\pm$ 1

```
.model small
```

```
.stack 256
```

```
.data
```

```
    a    db '1234567890','$'
```

```
    b    db '0987654321','$'
```

```
.code
```

```
main proc
```

```
    assume es:@data
```

```
    mov ax,@data
```

```
    mov ds,ax
```

```
    mov es,ax
```

```
    mov ah,9
```

```
    lea dx,b
```

```
    int 21h
```

```
    lea si,a
```

```
    lea di,b
```

```
    mov cx,10
```

```
    rep movsb
```

```
    mov ah,9
```

```
    lea dx,b
```

```
    int 21h
```

```
    .exit 0
```

```
main endp
```

```
    end    main
```

Результат:

```
09876543211234567890
```

# Команды управления ЦП

Флаг переноса	Флаг направления	Флаг прерываний	Регистр флагов
Stc	Std	Sti	Lahf
Clc	Cld	Cli	Sahf
Cmc			

# Загрузка регистра флагов

- **LAHF** – в регистр AH
- **SAHF** – из регистра AH

