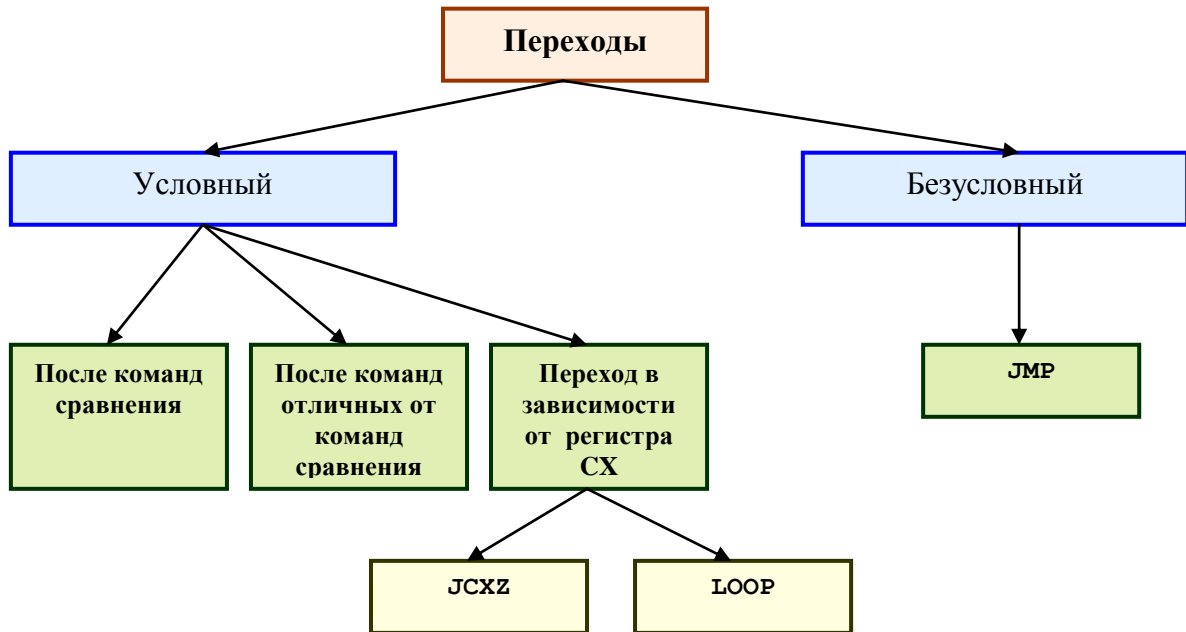


Лекция №11

Тема "Переходы. Циклы"



Безусловный переход

```
JMP <операнд>
```

операнд - указывает на адрес перехода

Безусловный переход изменяет адрес следующей исполняемой команды, который хранится в регистре IP (*указатель команд*). Поэтому сделать переход по адресу - значит записать данный адрес в регистр IP.

Особенность заключается в том, что в команде безусловного перехода указывается не адрес, а разность между ним и адресом команды перехода. И действие самой команды перехода заключается в прибавлении этой величины к текущему значению регистра IP.

Прямой переход

операнд - метка

Пример

```

    JMP L
    .
    .
    .
L:  mov ax, 0
  
```

Косвенный переход

операнд - r16, m16

Пример

```
MODEL small
STACK 256
DATA SEG
CODE SEG
```

```
A          dw  L
exCode      DB    0
Start:
           mov    ax, @data
           mov    ds, ax

           Jmp  A          ; mov    dx, A
           mov    ax, 0      ; Jmp    dx
L:         mov    ax, 1

Exit:
           mov    ah, 04Ch
           mov    al, [exCode]
           int    21h

           END    Start
```

Команды сравнения и условного перехода

Условный переход реализуется в два шага:

- сравниваются некоторые величины, в результате чего соответствующим образом формируются флаги (ZF, SF и т.д.)
- выполняется собственно условный переход в зависимости от значений флагов

Команда сравнения

СМР <операнд-1>, <операнд-2>

Эта команда эквивалентна команде SUB, за одним исключением: вычисленная разность никуда не записывается. Поэтому результат - это установка флагов, характеризующих полученную разность.

Команды условного перехода

Жхх <метка>

Условные обозначения, используемые в мнемокоде Жхх

Е	equal	равно	
N	not	не, отрицание	
G	greater	больше	для чисел со знаком
L	less	меньше	
A	above	выше, больше	для чисел без знака
B	below	ниже, меньше	

1 группа команд условного перехода

Команды условного перехода, используемые после команд сравнения.

Мнемокод	Условие для перехода после команды CMP	Состояние флагов для перехода
для любых чисел:		
JE	$op1 = op2$	$ZF = 1$
JNE	$op1 \neq op2$	$ZF = 0$
для чисел со знаком:		
JL / JNGE	$op1 < op2$	$SF \neq OF$
JLE / JNG	$op1 \leq op2$	$SF \neq OF$ или $ZF = 1$
JG / JNLE	$op1 > op2$	$SF = OF$ и $ZF = 0$
JGE / JNL	$op1 \geq op2$	$SF = OF$
для чисел без знака:		
JB / JNAE	$op1 < op2$	$CF = 1$
JBE / JNA	$op1 \leq op2$	$CF = 1$ или $ZF = 1$
JA / JNBE	$op1 > op2$	$CF = 0$ и $ZF = 0$
JAЕ / JNB	$op1 \geq op2$	$CF = 0$

Пример

Почему для знаковых чисел (JL) проверяется соотношение $OF \neq SF$ ($OF = 1$ и $SF = 0$ либо $OF = 0$ и $SF = 1$) ?

Возможны две комбинации флагов, соответствующие условию $op1 < op2$:

1. Если при вычитании не было переполнения мантиссы ($OF = 0$), тогда флаг SF фиксирует настоящий знак разности и поэтому $op1 - op2 < 0$, то $SF = 1$.
2. Если при вычитании произошло переполнение мантиссы ($OF = 1$), тогда результат команды будет число с противоположным знаком, чем у настоящей разности. Поэтому $SF = 0$ означает, что у искаженного результата знак положителен, а значит у настоящей разности знак отрицателен, т.е. $op1 < op2$.

Пример

Пусть X, Y, Z - переменные размером в слово. Требуется записать в Z максимальное из чисел X и Y.

Числа со знаком	Числа без знака
<pre> mov ax, X cmp ax, Y ; x = y ? JGE M ; x >= y -> M mov ax, Y M: mov Z, ax </pre>	<pre> mov ax, X cmp ax, Y ; x = y ? JAЕ M ; x >= y -> M mov ax, Y M: mov Z, ax </pre>

2 группа команд условного перехода

Команды ставятся после команд отличных от команд сравнения.

Мнемокод	Условный переход	Мнемокод	Условный переход
JZ	ZF = 1	JNZ	ZF = 0
JS	SF = 1	JNS	SF = 0
JC	CF = 1	JNC	CF = 0
JO	OF = 1	JNO	OF = 0
JP	PF = 1	JNP	PF = 0

Пример

Пусть A, B, C - беззнаковые байтовые переменные.

Требуется вычислить

$C = A * A + B$, но если ответ превосходит размер байта, тогда надо передать управление на метку ERROR.

```

mov  al, A
mul  al
JC   ERROR      ; A * A > 255 (CF = 1) -> ERROR
add  al, B
JC   ERROR      ; перенос (CF = 1) -> ERROR
mov  C, al
...
ERROR:

```

3 группа команд условного перехода

Команда условного перехода проверяет значение регистра CX.

JCXZ <метка>

Действие команды можно описать:

IF CX = 0 THEN goto <метка>

Особенность команд условного перехода

Эти команды осуществляют только **«короткий»** переход, т.е. управление можно передать не далее чем на 127-128 байтов вперед и назад (примерно 30-40 команд).

КОП i8 → IP := IP + i8

Это можно избежать, если использовать команду безусловного перехода JMP, которая позволяет делать **«далекий»** переход (i16).

Например, при «далекой» метке M оператор

IF AX = BX THEN goto M

следует реализовать так:

<pre>IF AX <> BX THEN goto L; { короткий переход} ... L: goto M; ... { длинный переход} M:</pre>	<pre> CMP ax, bx JNE L . . . L: JMP M . . . M:</pre>
--	---

Задачи

1. Даны два беззнаковых числа A и B. Найти значение числа C = A / B. Если B=0, то C=0.
2. Даны три числа A, B, C. Записать в A максимальное число, в B - минимальное число, а в C - сумму абсолютных значений A и B.

Команды управления циклом

С помощью команд перехода можно реализовать любые разветвления и циклы.

<pre>If X > 0 then S1 else S2</pre>	<pre>CMP X, 0 JLE L2 S1 JMP FIN L2: S2 FIN:</pre>
--	---

<pre>While X > 0 do S</pre>	<pre>BEG: CMP X, 0 JLE FIN S ; тело цикла JMP BEG FIN:</pre>
--------------------------------	--

<pre>Repeat S until X > 0</pre>	<pre>BEG: S ; тело цикла CMP X, 0 JG BEG</pre>
------------------------------------	--

<pre>For I := 1 to N do S</pre>	<pre>BEG: mov cx, N ; cx - счетчик цикла - ; число повторений L: S ; тело цикла dec cx ; cx := cx - 1 CMP cx, 0 ; cx = 0 ? JNE L ; cx <> 0 --> goto L</pre>
---------------------------------	---

Команда LOOP

В конце циклов (для FOR) всегда применяется одна и та же тройка команд, которую заменили **LOOP** <метка>

Действие этой команды можно описать так:

```
cx := cx - 1; IF cx <> 0 THEN goto <метка>
```

Тогда цикл будет выглядеть так:

```
      mov  cx, N
L:    ...           ; тело цикла

      LOOP L
```

Особенности команды LOOP:

1. В качестве счетчика цикла используется регистр CX.
2. Начальное значение (количество повторений цикла) для CX должно быть присвоено до цикла.
3. Тело цикла будет выполнено хотя бы раз.

Пример

N может быть числом от 0 до 8. Вычислить N! Результат записать в регистр AX.

```
      mov  ax, 1      ; ax := 0!
      mov  cl, N
      mov  ch, 0      ; cx := N как слово (счетчик циклов)
      JCXZ F1         ; при N = 0 обойти цикл
      mov  si, 1      ; i := 1
F:    mul  si         ; (dx,ax) := ax * i (dx=0)
      inc  si         ; i := i + 1
      LOOP F
F1:
```