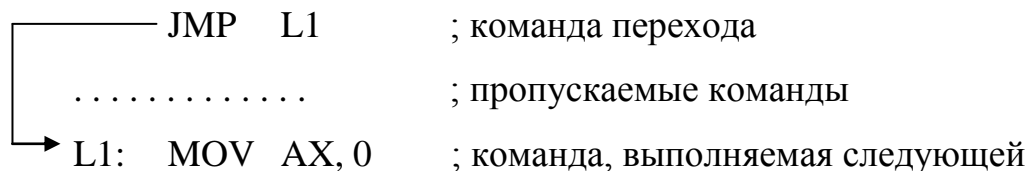


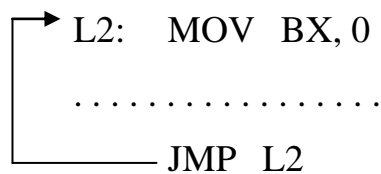
7. Команды передачи управления (перехода)

Назначение – изменение естественного последовательного порядка выполнения потока команд. Причины достаточно очевидны - необходимость разветвления вычислительного процесса или организация повторений. Существуют два типа переходов - условный и безусловный.

Более простым является безусловный переход, который задается командой JMP (от Jump, прыжок). Единственный операнд команды задает адрес той команды, которая должна выполняться следующей после текущей. Этот адрес может задаваться напрямую или косвенно. В первом случае операндом команды JMP является метка команды, на которую надо выполнить переход:



Здесь переход выполняется вперед, но можно – и назад, на одну из предыдущих команд:



Замена метки-операнда адресом перехода выполняется ассемблирующей программой при обработке исходного текста (более подробно этот процесс рассматривается в теме ____). Фактически команда JMP просто заносит в регистр IP новое значение – адрес команды, на которую выполняется переход. При этом адрес задается не абсолютный, а относительный, т.е. в машинную команду перехода заносится разность между адресом команды JMP и адресом целевой команды (значением метки перехода). Эта разность называется смещением и рассматривается, как знаковое целое: переход вперед – положительное смещение, переход назад – отрицательное. При этом для хранения смещения в машинной команде можно отвести либо 1 байт (короткий переход, от -128 до +127 байтов т.е.

примерно на 30-40 команд), либо 1 слово (длинный переход, от -32 768 до +32 767 байтов, т.е. примерно на 8 тысяч команд).

Большинство ссылок в обычных программах – короткие (так называемый принцип локальности ссылок), что позволяет создавать и более короткие машинные программы. Ассемблер при обработке исходной программы сам решает, какой тип перехода использовать, вычисляя смещение и анализируя его величину. Здесь возможна проблема при использовании ссылок вперед, когда при формировании текущей машинной команды перехода адрес метки перехода еще не известен, и ассемблеру ничего другого не остается, как предполагать длинный переход. Можно явно указать короткий переход с помощью директивы **SHORT**:

```
JMP SHORT L3
```

```
.....
```

```
L3: ADD AX, 1
```

Если на самом деле переход не короткий, то ассемблер выдаст ошибку.

Особенностью косвенного перехода является то, что операнд команды **JMP** задает имя регистра или имя области памяти, где находится адрес перехода.

Перейдем к рассмотрению условных переходов. Как следует из названия, условный переход выполняется только при выполнении заданного условия. Условие перехода задается одним или несколькими флагами регистра **Flags**, значения которых устанавливаются предыдущими командами. Очень часто перед командой перехода используется команда сравнения двух операндов с мнемоникой **CMR** (от *Compare*, сравнивать), которая всего лишь устанавливает соответствующие значения флагов:

```
CMR op1, op2 ; сравнить два операнда
```

Здесь операнды могут задаваться в любых комбинациях, кроме “память” – “память”. После команды сравнения можно поставить одну из следующих команд условного перехода:

```
JE метка ; выполнить переход, если op1 = op2
```

JNE метка ; выполнить переход, если op1 и op2 не равны

Кроме того, при сравнении знаковых чисел можно использовать следующие команды условных переходов:

JG метка ; выполнить переход, если op1 > op2

JGE метка ; выполнить переход, если op1 >= op2

JL метка ; выполнить переход, если op1 < op2

JLE метка ; выполнить переход, если op1 <= op2

Аналогичные команды введены для беззнаковых чисел, только вместо символов G и L (т.е. Greater и Less) используются A и B (от Above и Below).

Во всех командах условных переходов при невыполнении условия будет выполняться следующая по порядку команда, т.е. переход не происходит.

Альтернативную группу команд условных переходов составляют команды, в которых не используется операция сравнения, а сразу анализируются соответствующие флаги:

JZ или JNZ (переход, если флаг Zero =1 или 0)

JS или JNS (переход, если флаг Sign =1 или 0)

JC или JNC (переход, если флаг Carry =1 или 0)

JO или JNO (переход, если флаг Overflow =1 или 0)

Наконец, есть одна специальная команда, которая проверяет значение регистра CX и имеет мнемонику JCXZ (то есть Jump if CX is Zero).

Рассмотрим два простых примера.

Пример 1. Пусть X и Y – знаковые переменные размером в слово.

$$\text{Найти } Z = \max(X, Y) = \begin{cases} X, \text{ если } X > Y \\ Y, \text{ если } X < Y \end{cases}$$

Будем считать, что для размещения в памяти значений этих переменных объявлены двухбайтовые области памяти с именами X, Y и Z.

Фрагмент программы:

MOV AX, X ; (AX) = X

```

CMP  AX, Y    ; (AX) сравнить с Y
JGE  L1       ; перейти на метку L1, если X >= Y
MOV  AX, Y    ; если X < Y, то (AX) = Y
L1:  MOV  Z, AX ; Z = (AX)

```

Пример 2. Моделирование условного оператора языка высокого уровня: **if** ($X > 0$) **then** группа1 **else** группа2;

Необходимые команды:

```

CMP  X, 0    ; сравнить X с 0
JLE  L1      ; если X <= 0, то переход на L1,
                ; иначе (если X > 0) - выполнение следующей команды
.....
команды группы 1
.....
JMP  L2      ; безусловный переход на первую команду после
                ; условного оператора
L1:  первая команда группы 2
.....
      последняя команда группы 2
L2:  первая команда после условного оператора

```

Практические задания к теме №7.

Задание 1. Пусть M, L, N – переменные размером в слово. Требуется записать в L максимальное из чисел M и N . Листинг программы:

```

assume  cs:cod, ds:data
data   SEGMENT
      N   dw  5
      M   dw  4
      L   dw  ?
data   ENDS

```

```

cod    SEGMENT
main:
        mov  ax, data
        mov  ds, ax
        mov  ax, N
        cmp  ax, M
        jae  Lab
        mov  ax, M
Lab:    mov  L, ax
cod    ENDS
END main

```

Задание 2. Определить значение функции $F(X)$, принимающей различные значения на интервалах:

$$f(x) = \begin{cases} x^2, & \text{если } x > L \\ \frac{x}{2}, & \text{если } x \leq L \end{cases}$$

Фрагмент программы:

```

Z dw L
....
    cmp  ax, Z ; в ax находится значение x
    jg   L2 ; выполняется ветка возведения в квадрат, если ax > L
    idiv 2 ; выполняется деление содержимого ax пополам
L2: mul  ax

```

(самост)

Задание 3. Написать программу нахождения максимума из 3-х байтовых чисел

Задание 4. Нахождение минимума из пяти чисел размером в слово

Задание 5. Определение значения функции, заданной уравнением:

$$F(x) = \begin{cases} x^3 + \frac{1}{z}, & -1 < x < 1 \\ (x+z)^2, & 1 > x > 2 \\ (x+z) * x, & x = 0 \end{cases}$$