

# 15. Команда трансляции по таблице.

## Команда XLAT <адрес> / XLATB

Помещает в AL байт из таблицы по адресу DS:BX со смещением относительно начала таблицы, равным AL. Адрес, указанный в исходном коде, не обрабатывается компилятором и служит в качестве комментария. Если в адресе явно указан сегментный регистр, он будет использоваться вместо DS.

Короче говоря, XLATB -> `AL = DS:[(E)BX + AL]`

**Таблица перекодировки** - строка байтов (область памяти размером от 1 до 255 байтов)

## Как понять какой байт взять из таблицы памяти?

Адрес байта в таблице определен суммой BX + AL, то есть номер байта, расположен в AL. А потом после выполнения команды в AL находится уже не номер байта, а значение этого байта в таблице. В BX предварительно должен быть загружен адрес строки байтов.

## Отличие XLAT от XLATB

```
mov bx, 0      ; поместит в bx адрес таблицы
mov al, 6      ; поместит в al номер байта в таблице
xlat ds:[bx+al] ; провели трансляцию - результат в al
```

Если мы не хотим указывать сегментный регистр, то можно использовать команду XLATB:

```
mov bx, 0      ; поместит в bx адрес таблицы
mov al, 6      ; поместит в al номер байта в таблице
xlatb          ; провели трансляцию - результат в al
```

## Применение

Команда трансляции по таблице является мощным средством перекодировки данных. Например, при переводе нижнего регистра в верхний.

**Пример:** перевод десятичного числа (от 0 до 15) в соответствующую цифру в 16 с/с.

```
; сегмент данных
table db '0123456789ABCDEF' ; таблица перекодировки

; сегмент кода
lea bx, table                ; поместить в bx адрес таблицы
mov al, decimal_digit        ; поместить в al десятичную цифру
xlatb                        ; провести трансляцию
```

Теперь в AL находится ASCII-код соответствующей цифры в 16 с/с.