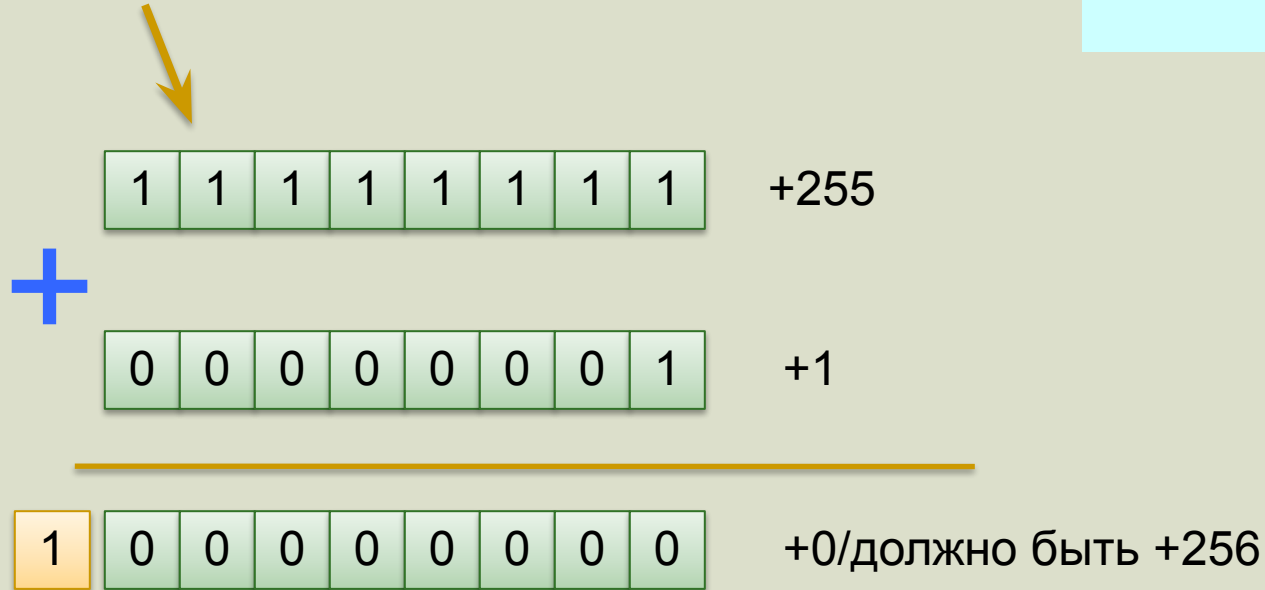

Переполнение

Переполнение

целые типы

беззнаковые

unsigned char



Для **беззнаковых** типов переполнение определяется по факту переноса бита за пределы разрядной сетки/занятия бита за пределами разрядной сетки

Не хватило разрядов.
Перенос за пределы
разрядной сетки

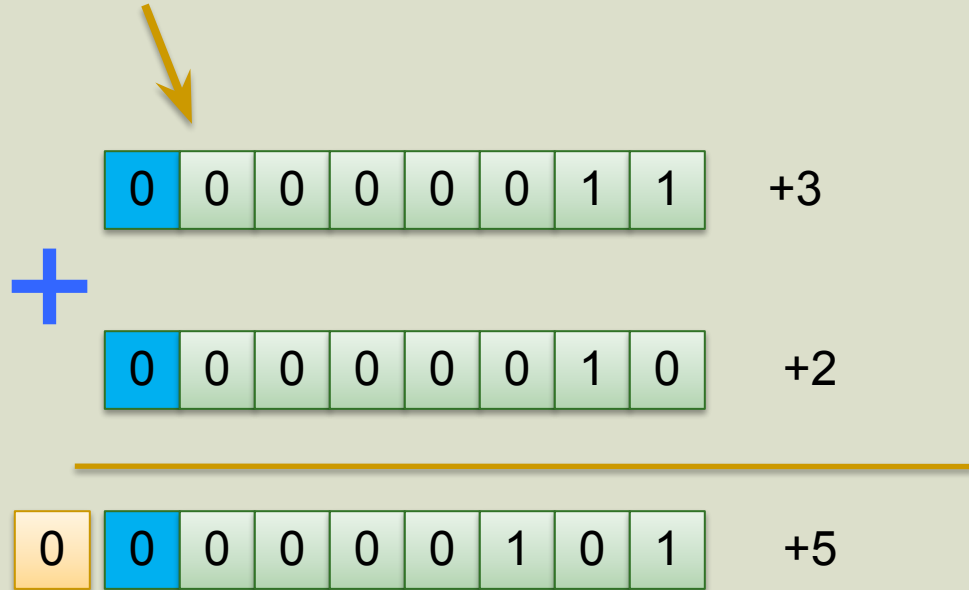
переполнение

Переполнение

целые типы

числа со знаком

signed char



Переноса за пределы разрядной сетки нет

Переноса в знаковый разряд нет

Переполнения
НЕТ

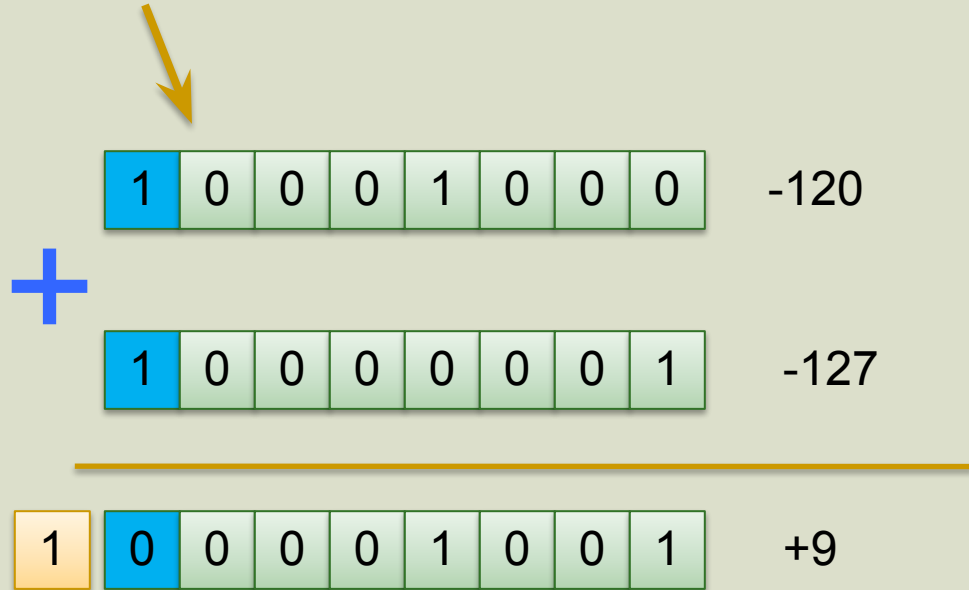
Перенос за пределы разрядной сетки	Перенос в знаковый разряд	Переполнение
-	-	-
+	-	+
-	+	+
+	+	-

Переполнение

целые типы

числа со знаком

signed char



Перенос за пределы разрядной сетки

Переноса в знаковый разряд нет

переполнение

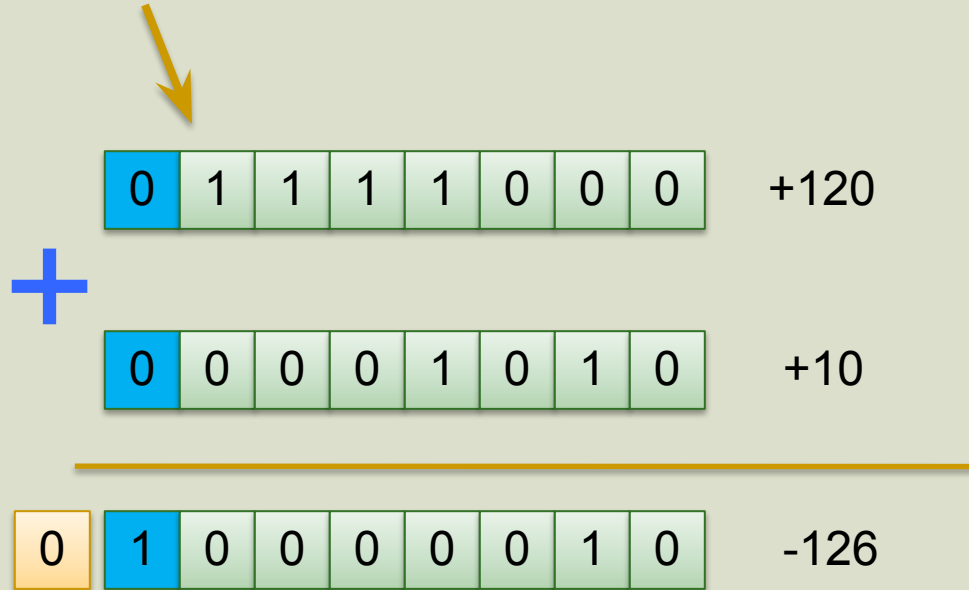
Перенос за пределы разрядной сетки	Перенос в знаковый разряд	Переполнение
-	-	-
+	-	+
-	+	+
+	+	-

Переполнение

целые типы

числа со знаком

signed char



Переноса за пределы разрядной сетки нет

Перенос в знаковый разряд

переполнение

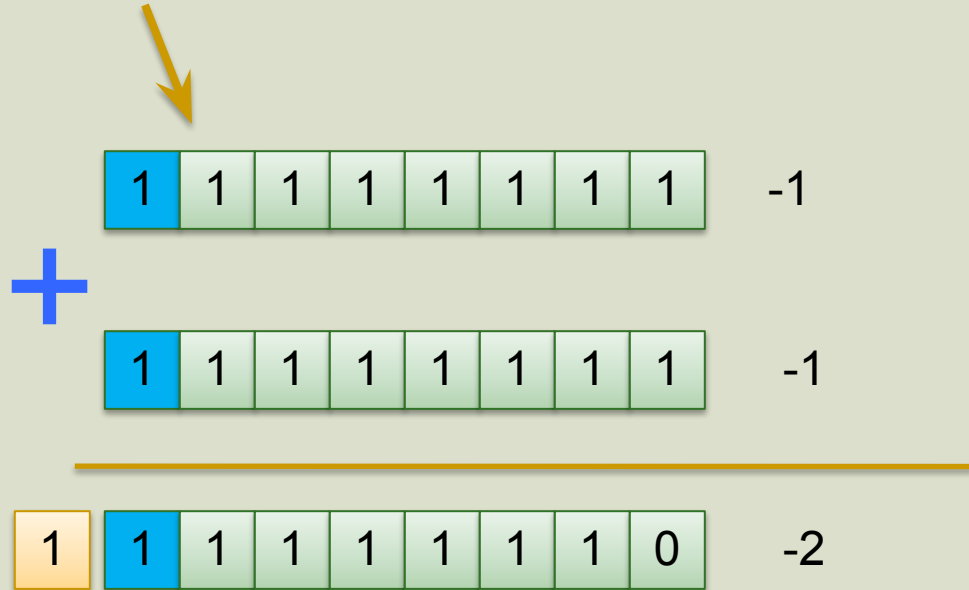
Перенос за пределы разрядной сетки	Перенос в знаковый разряд	Переполнение
-	-	-
+	-	+
-	+	+
+	+	-

Переполнение

целые типы

числа со знаком

signed char



Перенос за пределы разрядной сетки

Перенос в знаковый разряд

Переполнения
НЕТ

Перенос за пределы разрядной сетки	Перенос в знаковый разряд	Переполнение
-	-	-
+	-	+
-	+	+
+	+	-

Переполнение

Знаковые и беззнаковые типы

Полностью эквивалентные команды

`mov al, -1`

`mov al, 255`

Команды `mov`, `add`, `sub` не различают знаковые и беззнаковые типы

signed char



-1

unsigned char



255

Переполнение

Примеры

```
#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
    char isOverflow;
    __asm
    {
        mov al, -1
        mov bl, -1
        add al, bl
        mov isOverflow, 1
        jo exit1; Считаem, что числа со знаком
        mov isOverflow, 0
    }
    exit1:
    if (isOverflow) printf("Overflow\n");
    if (!isOverflow) printf("Not overflow\n");
    return 0;
}
```

```
#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
    char isOverflow;
    __asm
    {
        mov al, 255
        mov bl, 255
        add al, bl
        mov isOverflow, 1
        jc exit1; Считаem, что числа без знака
        mov isOverflow, 0
    }
    exit1:
    if (isOverflow) printf("Overflow\n");
    if (!isOverflow) printf("Not overflow\n");
    return 0;
}
```

jo – команда условного перехода. Переход в случае знакового переполнения

jc – команда условного перехода. Переход в случае переноса за пределы разрядной сетки