

Арифметка в 8-разрядном процессоре

Отрицательные числа представляются в допкоде (two's complement) aka коды положительного и отрицательного числа дополняют друг друга до  $2^n$   
 $A + [-A]_{\text{доп}} = 2^n$



Вычитание по сути - замена на сложение с дополнением вычитаемого

$$A - B = A + [-B]_{\text{доп}}$$

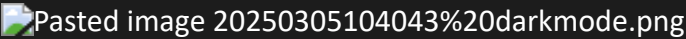
^ Буквально начальная школа лмао

Сложение/вычитание чисел со знаком и без знака в процессоре работает одинаково, что там по операндам и результатам - на разработчике

Способы перевода из допкода в десятичную систему:

- Первод модуля + добавить минус
- Сумма степеней 2, где старший бит имеет вес  $-2^n$

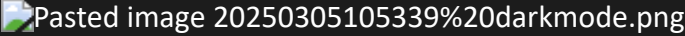
Сложение и вычитание 8-разрядных чисел на уровне процессора



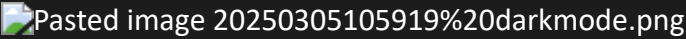
Флаги операций

Для чисел знаковых/беззнаковых:

- **Z (Zero)** - Результат = 0
- **C (Carry)** - Перенос при сложении, заём при вычитании
- **H (Half Carry)** - Межтетрадный перенос/заём  
Только для знаковых чисел:
- **N (Negative)** - Результат < 0
- **V (Two's complement Overflow)** - Переполнение допкода
- **S (Sign)** - Знак результата  $S = V \oplus N$   
Пример (ебал я его переписывать держи скрин):  
// Лол той же хернёй занимались лекцию назад, зачем ещё раз???



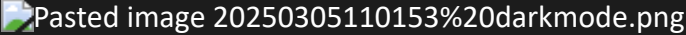
Сложение/вычитание многобайтовых операндов



Алгоритм умножения

```
; ch:c1 = a*b
mul:
    clr c1 ; C = 0
    clr ch
    ldi n,8
loop:
    clc
    sbrc b, 0
    add ch, a
    ror ch
    ror c1
    lsr b
    dec n
    brne loop
    ret
```

Алгоритм (идея считай та же что у умножения столбиком):



Деление с восстановлением остатка

16-разрядное делимое, 8-разрядные делитель и частное



$$C = [AH,AL] / B$$

Проверка на отсутствие переполнения результата:  $B > AH$

Пример (я всё ещё ебал переписывать):

