

Кодирование информации в ЭВМ (компьютере)

Современные компьютеры могут обрабатывать следующие виды информации:

- Цифровую
- Текстовую
- Графическую
- Звуковую
- Видео
- И т.д.

Кодирование информации в ЭВМ (компьютере)

Компьютеры являются **цифровыми** устройствами.

Это значит, что компьютеры обрабатывают информацию, заданную в **цифровом, двоичном** (или **бинарном**) виде.

Поэтому, чтобы компьютеры могли обрабатывать различные виды информации, эту информацию надо **закодировать** в двоичном виде и **ввести** ее в компьютер.

Мы начинаем большой раздел нашего курса «Информатика», в котором рассмотрим вопросы кодирования **различных** видов информации в компьютере.

Типы данных, поддерживаемых процессором

Основными **типами данных** поддерживаемых аппаратно процессором являются:

1. Байт
2. Слово
3. Двойное слово
4. Учетверенное слово
5. И др.

Вообще память компьютера представляет совокупность **бит**. **Бит** - это **один двоичный** разряд, принимающий значение 0 или 1.

Однако биты в компьютере не адресуются, **не имеют адреса**.

Наименьшей **адресуемой** частью памяти является **байт**.

Байт – 8 последовательно расположенных бит и имеющий адрес.

8 бит байта нумеруются от 0 до 7 справа налево.

Номера бит	7	6	5	4	3	2	1	0
Байт								

Слово – два байта (16 бит), имеющих последовательные адреса.

Слово состоит из двух байтов: *младшего* и *старшего*.

Младший байт хранится по меньшему адресу.

Адресом слова является **адрес младшего байта**.

Двойное слово – четыре байта (32 бита), расположенных по последовательным адресам.

Двойное слово состоит из *младшего слова* и *старшего*.

Младшее слово хранится по меньшему адресу.

Адресом двойного слова является **адрес младшего слова**.

Учетверенное слово – восемь байт (64 бита),
расположенных по последовательным адресам.

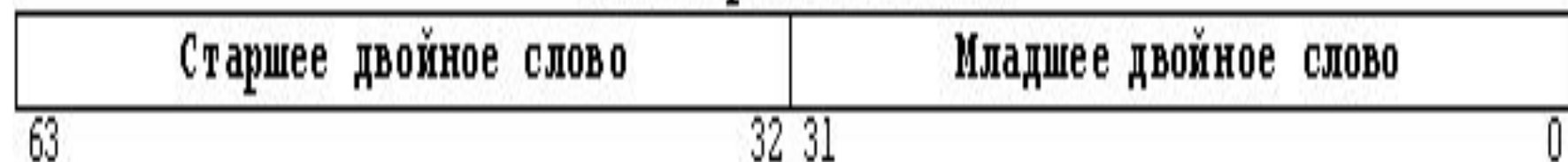
Учетверенное слово состоит из *младшего двойного слова* и *старшего двойного слова*.

Младшее двойное слово хранится по меньшему адресу.

Адресом учетверенного слова является *адрес младшего двойного слова*.



Учетверенное слово



Кодирование целых чисел без знака

Рассмотрим кодирование на ЭВМ целых **положительных**, или **натуральных** чисел.

Целые числа без знака в памяти ЭВМ могут иметь любой тип данных, т.е. записываться в **байт**, **слово**, **двойное слово** или **четверенное слово**.

Будем, в основном, рассматривать байтовое представление, т.е. кодирование в байт.

Код в **байтовом** представлении ВСЕГДА состоит из **8 двоичных** разрядов: **не больше и не меньше**.

Код в **двухбайтовом** представлении (**слово**) ВСЕГДА состоит из **16 двоичных** разрядов.

И.т.д.

Правило кодирования целых чисел без знака для любого типа данных:

- в младших разрядах кода размещается модуль целого числа без знака в двоичной системе счисления,
- старшие незанятые биты заполняются нулями.

модуль

Пример 1. Записать код целого 93_{10} числа без знака в **байтовом** представлении.

Переводим десятичное число в двоичное:

$$93_{10} = 101\ 1101_2$$

$$\boxed{0}\ \boxed{1}\ \boxed{0}\ \boxed{1}\ \boxed{1}\ \boxed{1}\ \boxed{0}\ \boxed{1} = 01011101_2 = 5D_{16}$$

Пример 2. Записать код целого 93_{10} числа без знака в **двухбайтовом** представлении, слово.

$$\boxed{0}\ \boxed{0}\ \boxed{0}\ \boxed{0}\ \boxed{0}\ \boxed{0}\ \boxed{0}\ \boxed{0}\ \boxed{0}\ \boxed{0}\ \boxed{1}\ \boxed{0}\ \boxed{1}\ \boxed{1}\ \boxed{1}\ \boxed{0}\ \boxed{1} = 005D_{16}$$

Рассмотрим пределы представления целых чисел без знака в разных представлениях.

Рассмотрим **байтовое** представление.

Минимальное целое число без знака равно $0_{10} = 0_2$.

В однобайтовом представлении код 0 будет:

$$\begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} = 00_{16}$$

В однобайтовом представлении **максимально возможное представимое целое число без знака** будет равно:

$$1111\ 1111_2 = 2^8 - 1 = 255_{10}:$$

$$\begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline \end{array} = FF_{16}$$

Итак, в **байтовом** представлении можно закодировать целые числа без знака в пределах от 0 до 255_{10} .

$$\text{Различных чисел} - 256 = 2^8$$

Рассмотрим **двухбайтовое** представление – тип данных **СЛОВО**.

Минимальное целое число без знака равно $0_{10} = 0_2 = 0_{16}$.

В двухбайтовом представлении код 0 будет:

 = 0000_{16}

В двухбайтовом представлении **максимально возможное представимое целое число без знака** будет равно:

$1111\ 1111\ 1111\ 1111_2 = 2^{16} - 1 = 65535_{10}$:

 = $FFFF_{16}$

Итак, в двухбайтовом представлении можно закодировать **целые числа без знака** в пределах от 0 до 65535_{10} .

Различных чисел – $65536 = 2^{16}$

Примеры:

$$68_{10} = 100\ 0100_2$$

Байт:  = 44_{16}

Слово:  = 0044_{16}

$$365_{10} = 1\ 0110\ 1101_2$$

Байт: **Однобайтового представления числа 365 нет.**

Слово:  = $016D_{16}$

Кодирование целых чисел со знаком

Для кодирования целых чисел со знаком в ЭВМ применяют:

- 1. прямой код**
- 2. обратный код**
- 3. дополнительный код**

1. *Прямой код* для целых чисел со знаком

В прямом коде **старший бит (бит знака)** всегда используется для обозначения **знака числа**.

Если число является положительным, то бит знака равен 0,

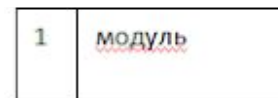
Если число - отрицательное, то бит знака равен 1.

В младших битах *прямого кода* располагается модуль числа в двоичной системе счисления.

Формат двоичного числа со знаком в прямом коде:



а) положительное
число



б) – отрицательное

ПРИМЕР. Рассмотрим *прямой код* целого числа со знаком в байтовом представлении (будем этот код кратко обозначать **ПК8**).

Возьмем число $28_{10} = 11100_2$

$$+28 = 1\ 1100_2 \quad \Rightarrow \text{ПК8}(+28) = \mathbf{0001\ 1100} = 1C_{16}$$

$$-28 = -1\ 1100_2 \quad \Rightarrow \text{ПК8}(-28) = \mathbf{1001\ 1100} = 9C_{16}$$

Красным цветом выделены значения старших битов байта:

- Для положительного числа (+28) бит знака равен **0**
- Для отрицательного числа (-28) бит знака равен **1**

Младшие биты байта для **ПРЯМОГО** кода чисел (+28) и (-28) одинаковые: это модуль числа $28_{10} = \mathbf{001\ 1100_2}$

Примеры.

А) Записать в прямом коде байтовом представлении следующие числа:

$$15_{10} = 1111_2 \quad \boxed{000\ 1111} \quad \text{ПК8} = 0F$$

$$-15_{10} = -1111_2 \quad \boxed{1000\ 1111} \quad \text{ПК8} = 8F$$

Б) Записать десятичное число, прямой код которых в байтовом представлении равен:

ПК8 =

$\boxed{0010\ 1100}$

$\boxed{1010\ 1100}$

$\boxed{0000\ 0000}$

$\boxed{1000\ 0000}$

44

-44

0

0

ВНИМАНИЕ: В ПРЯМОМ коде число 0 имеет **ДВА** кода.

Рассмотрим пределы кодирования целых чисел со знаком в ПК8 (байтовое представление).

Код максимального числа:

Это ПК числа: $+127_{10}$.

0111 1111

Код минимального числа:

Это ПК числа : -127 .

1111 1111

ВЫВОД: В ПК8 можно закодировать целые числа со знаком от -127_{10} до $+127_{10}$.

Различных чисел - 255

В ПК16 пределы кодирования целых чисел со знаком от -32767_{10} до $+32767_{10}$.

Правила сложения в **прямом** коде **целых чисел со знаком**, имеющих **одинаковые** знаки:

- **складываем** модули обоих чисел
- **присваиваем** результату **общий знак** чисел

Правила сложения в **прямом** коде **целых чисел со знаком**, имеющих **разные** знаки:

- определяем число, большее по модулю
- **вычитаем** из числа с **большим** модулем число с **меньшим** модулем
- **присваиваем** результату **знак** числа с **большим** модулем.

Недостатки ПК:

- два нуля (**0000 0000** и **1000 0000**)
- неудобство выполнения сложения чисел

2. Обратный код для целых чисел со знаком

В обратном коде (ОК), как и в прямом коде, для обозначения знака числа используется старший бит кода (бит знака):

Для положительного числа бит знака = 0

Для отрицательного числа бит знака = 1

ОК двоичного числа образуется по следующему правилу.

ОК положительных целых чисел со знаком совпадает с ПК.

ОК отрицательного числа:

- Знак бита = 1
- модуль числа заменяется на инверсный модуль (т.е. нули заменяются единицами, а единицы – нулями).

0	модуль
---	--------

1	Инверсн. модуль
---	-----------------

а) положительное число

б) –

отрицательное

Пределы представления чисел те же, что и ПК.

Пример: Получить ОК8 целых чисел со знаком.

$$+28_{10} = 11100_2: \quad \mathbf{0001\ 1100} \quad \text{ПК8} = 1\text{C}_{16} \quad \text{ОК8}(+\gg^*) = \text{ПК8} = 1\text{C}_{16}$$

$$-28_{10} = -11100_2: \quad \mathbf{1001\ 1100} \quad \text{ПК8}(-28) = 9\text{C}_{16}$$

$$\mathbf{1110\ 0011} \quad \text{ОК8}(-28) = \text{E3}_{16}$$

Примеры.

Записать десятичное число, **обратный** код которых в **байтовом** представлении равен

0011 0110

1011 0110

0000 0000

11111111

54

-73

0

0

ВНИМАНИЕ: В ОБРАТНОМ коде число 0 имеет **ДВА** кода.

3. Сложение целых чисел со знаком в обратном коде

Алгоритм сложения в ОК следующий:

- сложение кодов, включая знаковый разряд;
- прибавление (возможной) единицы переноса к младшему разряду суммы.
- Результат есть ОК суммы чисел.

Рассмотрим разные случаи.

1. А и В положительные.

$$A=3_{10} = 11_2$$

$$\text{ПК8}=0000\ 0011=\text{OK8}$$

$$B=7_{10} = 111_2$$

$$\text{ПК8}=0000\ 0111=\text{OK8}$$

$$A+B=10_{10}$$

Биты	8	7	6	5	4	3	2	1	0
перенос						1	1	1	
1 слаг.		0	0	0	0	0	0	1	1
2 слаг.		0	0	0	0	0	1	1	1
Сумма		0	0	0	0	1	0	1	0

Результат: $\text{OK8}(A+B)=0000\ 1010$.

Знаковый бит – 0: число положительное. Сумма двух положительных чисел есть число положительное. Верно.

$$\text{OK8}(A+B)=\text{ПК8}(A+B)=0000\ 1010 \Rightarrow \text{Модуль} = 10_{10}.$$

Вывод: Получен результат: +10, результат верный.

2. А положительное, В отрицательное и по абсолютной величине больше, чем А.

$$A = 3_{10} = 11_2$$

$$\text{ПК8}(A) = \text{ОК8} = 0000\ 0011$$

$$B = -10_{10} = -1010_2$$

$$\text{ПК8}(B) = 1000\ 1010$$

$$\text{ОК8}(B) = 1111\ 0101$$

$$A + B = -7_{10}$$

Биты	8	7	6	5	4	3	2	1	0
перенос						1	1	1	
1 слаг.		0	0	0	0	0	0	1	1
2 слаг.		1	1	1	1	0	1	0	1
Сумма		1	1	1	1	1	0	0	0

Результат: ОК8(A+B)=1111 1000.

Знаковый бит – 1: число отрицательное. Сумма (A+B) есть число отрицательное. Верно.

$$\text{ПК8}(A+B) = 1000\ 0111$$

Модуль числа=7

Вывод: Получен результат: -7, результат верный.

3. А положительное, В отрицательное и по абсолютной величине меньше, чем А.

$$A = 10_{10} = 1010_2$$

$$\text{ПК8}(A) = \text{ОК8}(A) = 0000 \ 1010$$

$$B = -3_{10} = -11_2$$

$$\text{ПК8}(B) = 1000 \ 0011$$

$$\text{ОК8}(B) = 1111 \ 1100$$

$$A+B = +7_{10}$$

Биты	8	7	6	5	4	3	2	1	0
перенос	1	1	1	1					
1 слаг.		0	0	0	0	1	0	1	0
2 слаг.		1	1	1	1	1	1	0	0
	1	0	0	0	0	0	1	1	0
									1
Сумма		0	0	0	0	0	1	1	1

Результат: $\text{ОК8}(A+B) = 0000 \ 0111$.

Знаковый бит – 0: число положительное. Сумма (A+B) есть число положительное. Верно.

$\text{ПК8}(A+B) = \text{ОК8}(A+B) = 0000 \ 0111 \Rightarrow \text{модуль} = 7$

Вывод: Получен результат: +7, результат верный.

4. А и В отрицательные.

$$A = -3_{10} = -11_2$$

$$\text{ПК8}(A) = 1000\ 0011$$

$$\text{ОК8}(A) = 1111\ 1100$$

$$B = -7_{10} = -111_2$$

$$\text{ПК8}(B) = 1000\ 0111$$

$$\text{ОК8}(B) = 1111\ 1000$$

$$A + B = -10_{10}$$

Биты	8	7	6	5	4	3	2	1	0
перенос	1	1	1	1	1				
1 слаг.		1	1	1	1	1	1	0	0
2 слаг.		1	1	1	1	1	0	0	0
	1	1	1	1	1	0	1	0	0
									1
Сумма		1	1	1	1	0	1	0	1

Результат: $\text{ОК8}(A+B) = 1111\ 0101$.

Знаковый бит – 1: число отрицательное. Сумма двух отрицательных чисел есть число отрицательное. Верно.

$\text{ПК8}(A+B) = 1000\ 1010 \Rightarrow$ модуль=10

Вывод: Получен результат: -10, результат верный.

5. А и В положительные, сумма $A+B > 127$.

$$A = 65_{10} = 1000001_2$$

$$\text{ПК8}(A) = 0100\ 0001 = \text{ОК8}(A)$$

$$B = 97_{10} = 1100001_2$$

$$\text{ПК8}(B) = 0110\ 0001 = \text{ОК8}(B)$$

$$A+B = +162_{10}$$

Биты	8	7	6	5	4	3	2	1	0
перенос		1							
1 слаг.		0	1	0	0	0	0	0	1
2 слаг.		0	1	1	0	0	0	0	0
Сумма		1	0	1	0	0	0	0	1

Результат: $\text{ОК8}(A+B) = 1010\ 0001$.

Знаковый бит – 1: число отрицательное.

Сумма двух **положительных** чисел **не может** быть **отрицательным** числом.

ВЫВОД: результат **неверный**.

Причина: сумма $A+B = +162_{10} > 127_{10}$ не может быть представлена в ОК8.

6. А и В отрицательные, сумма $A+B < -127$.

$$A = -65_{10} = -1000001_2$$

$$\text{ПК8}(A) = 1100\ 0001$$

$$\text{ОК8}(A) = 1011\ 1110$$

$$B = -97_{10} = -1100001_2$$

$$\text{ПК8}(B) = 1110\ 0001$$

$$\text{ОК8}(B) = 1001\ 1110$$

$$A+B = -162_{10}$$

Биты	8	7	6	5	4	3	2	1	0
перенос	1		1	1	1	1	1		
1 слаг.		1	0	1	1	1	1	1	0
2 слаг.		1	0	0	1	1	1	1	0
Сумма	1	0	1	0	1	1	1	0	0
									1
		0	1	0	1	1	1	0	1

Результат: $\text{ОК8}(A+B) = 0101\ 1101$.

Знаковый бит – 0: число **положительное**.

Сумма **двух отрицательных** чисел **не может** быть **положительным** числом

ВЫВОД: результат **неверный**.

Причина: сумма $A+B = -162_{10} < -127_{10}$ не может быть представлена в ОК8.

Пример.

Вычислить в ОК $63+(-34)$

Достоинства ОК:

Операция сложения выполняется одинаково, независимо от знаков и соотношения слагаемых.

Недостатки ОК:

- возникают два нуля: +0 и -0,

- (+0) = 

- (-0) = 

- в операции сложения требуется дополнительная операция по прибавлению бита переноса в младший разряд суммы.

4. *Дополнительный код* для целых чисел со знаком

Дополнительный код (ДК) строится следующим образом:

Для положительных чисел: $ДК=ОК=ПК$

Для отрицательных чисел:

$$ДК=ОК+1 \text{ (к младшему разряду)}$$

Схема преобразований отрицательного числа из ПК в ДК:

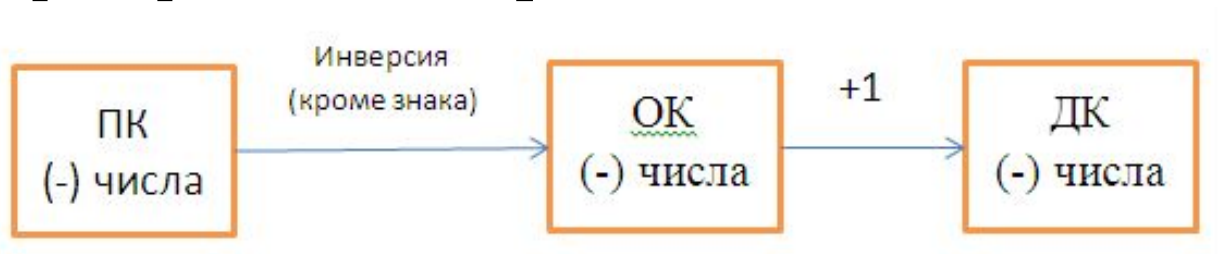
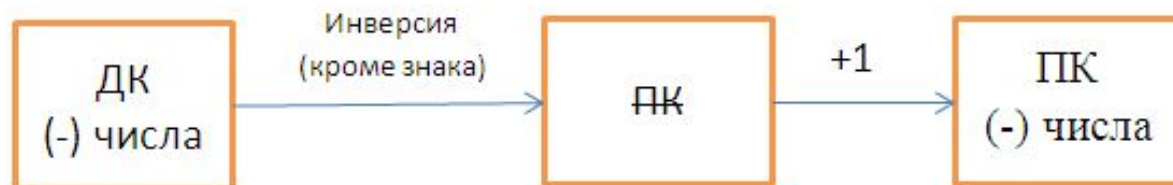


Схема преобразований отрицательного числа из ДК в ПК:



Пример.

Для числа (-51_{10}) получить ДК8:

$$-51_{10} = -11\ 0011_2$$

$$\text{ПК8}(-51) = 1011\ 0011$$

$$\text{ОК8}(-51) = 1100\ 1100 \quad (\text{инвертируем младшие биты})$$

$$\text{ДК8}(-51) = 1100\ 1101 \quad (\text{Прибавляем 1 к младшему разряду})$$

Пример.

Определить число, для которого ДК8=1100 1101

$$\text{ОК8} = 1011\ 0010 \quad (\text{инвертируем младшие разряды})$$

$$\text{ПК8} = 1011\ 0011 \quad (\text{Прибавляем 1 к младшему разряду})$$

Знаковый бит =1, число отрицательное

$$\text{Модуль} = 011\ 0011_2 = 51_{10}$$

Вывод: искомое число $= -51_{10}$

5. Сложение чисел в дополнительном коде

Алгоритм сложения чисел в ДК следующий:

- сложение кодов, включая знаковый разряд;
- отбрасывание (возможной) единицы переноса.
- Результат есть ДК суммы чисел.

Пределы представления чисел в ДК.

В ДК8 можно закодировать числа от -128 до +127.

Рассмотрим этот вопрос подробно.

Дополнительные коды *положительных* чисел со знаком в байтовом представлении (ДК8):

$$0_{10} = \text{ПК8} = \text{ОК8} = \text{ДК8} = 00000000_2 = 00_{16}$$

$$1_{10} = \text{ПК8} = \text{ОК8} = \text{ДК8} = 00000001_2 = 01_{16}$$

$$2_{10} = \text{ПК8} = \text{ОК8} = \text{ДК8} = 00000010_2 = 02_{16}$$

...

$$126_{10} = \text{ПК8} = \text{ОК8} = \text{ДК8} = 01111110_2 = 7E_{16}$$

$$127_{10} = \text{ПК8} = \text{ОК8} = \text{ДК8} = 01111111_2 = 7F_{16}$$

Вывод: 128 различных положительных чисел, включая 0.

Дополнительные коды *отрицательных* чисел со знаком в байтовом представлении (ДК8):

$-1_{10} =$	ПК8=10000001;	ОК8=11111110;	ДК8=11111111=FF ₁₆
$-2_{10} =$	ПК8=10000010;	ОК8=11111101;	ДК8=11111110=FE ₁₆
$-3_{10} =$	ПК8=10000011;	ОК8=11111100;	ДК8=11111101=FD ₁₆
...			
$-125_{10} =$	ПК8=11111101;	ОК8=10000010;	ДК8=10000011 =83 ₁₆
$-126_{10} =$	ПК8=11111110;	ОК8=10000001;	ДК8=10000010=82 ₁₆
$-127_{10} =$	ПК8=11111111;	ОК8=10000000;	ДК8=10000001=81 ₁₆

Обратим **ВНИМАНИЕ** на то, что не использован код
ДК8=10000000=80₁₆

Давайте выясним: какому числу соответствует дополнительный код ДК8=10000000

Давайте проверим

Вычислим: $A+B=(-127)+(-1)=(-128)$

Биты	8	7	6	5	4	3	2	1	0
перенос	1	1	1	1	1	1	1	1	
1 слаг.(-127)		1	0	0	0	0	0	0	1
2 слаг.(-1)		1	1	1	1	1	1	1	1
Сумма(-128)	±	1	0	0	0	0	0	0	0

Вычислим: $A+B=(-128)+(+1)=(-127)$

Биты	8	7	6	5	4	3	2	1	0
перенос									
1 слаг.(-128)		1	0	0	0	0	0	0	0
2 слаг.(+1)		0	0	0	0	0	0	0	1
Сумма(-1)		1	0	0	0	0	0	0	1

ДК8=1000 0001; ~~НК8~~=1111 1110; ПК8=1111 1111;

$A+B=-1111111_2=-127$

ВЫВОД: дополнительный код ДК8=10000000 ведет себя как **дополнительный код** числа (-128).

Примеры сложения чисел в дополнительном коде

Также рассмотрим шесть случаев:

1. А и В положительные.

$$A = 3_{10} = 11_2 = \text{ДК8}(A) = 0000 \ 0011$$

$$B = 7_{10} = 111_2 = \text{ДК8}(B) = 0000 \ 0111$$

$$A+B=10_{10}$$

Биты	8	7	6	5	4	3	2	1	0
перенос						1	1	1	
1 слаг.		0	0	0	0	0	0	1	1
2 слаг.		0	0	0	0	0	1	1	1
Сумма		<u>0</u>	0	0	0	1	0	1	0

Результат: ДК(A+B)=0000 1010.

Знаковый бит – 0: сумма положительная.

Верно: Сумма двух положительных чисел есть число положительное.

$$\text{ДК}(A+B) = \text{ОК}(A+B) = \text{ПК}(A+B) = 0000 \ 1010$$

$$\text{Модуль } 1010_2 = 10_{10}.$$

Вывод: Получен результат: $A+B = +10$, результат верный.

2. А положительное, В отрицательное и по абсолютной величине больше, чем А.

$$A = 3_{10} = 11_2; \quad \text{ДК8}(A) = 0000 \ 0011$$

$$B = -10_{10} = -1010_2; \quad \text{ПК8}(B) = 1000 \ 1010; \quad \text{ОК8}(B) = 11110101;$$

$$\text{ДК8}(B) = 1111 \ 0110$$

$$A+B = -7_{10}$$

Биты	8	7	6	5	4	3	2	1	0
перенос						1	1		
A		0	0	0	0	0	0	1	1
B		1	1	1	1	0	1	1	0
A+B		1	1	1	1	1	0	0	1

Результат: ДК8(A+B)=1111 1001 .

Знаковый бит – 1: число отрицательное.

Возможно: Сумма положительного и отрицательного числа может быть отрицательной.

$$\text{НК8}(A+B) = 1000 \ 0110; \quad \text{ПК8}(A+B) = 1000 \ 0111$$

$$\text{Модуль } 111_2 = 7_{10}.$$

Вывод: Получен результат: A+B=-7, результат верный.

3. А положительное, В отрицательное и по абсолютной величине меньше, чем А.

$A=10_{10}=1010_2; ПК8(A)=ОК8(A)=ДК8(A)=0000 1010$

$B = -3_{10} = -11_2; ПК8(B)=1000 0011; ОК8(B)=1111 1100;$

$ДК8(B)=1111 1101$

$A+B=7_{10}$

Биты	8	7	6	5	4	3	2	1	0
перенос	±	1	1	1	1				
A=10		0	0	0	0	1	0	1	0
B=-3		1	1	1	1	1	1	0	1
A+B=7		0	0	0	0	0	1	1	1

Результат: $ДК8(A+B)= 0000 0111 .$

Знаковый бит – 0: число положительное.

Возможно: Сумма положительного и отрицательного числа может быть отрицательной.

$ДК(A+B)=ОК(A+B)=ПК(A+B)=0000 0111$

Модуль $111_2=7_{10}$.

Вывод: Получен результат: $A+B= +7$, результат верный.

4. А и В отрицательные..

$A = -7_{10} = -111_2$; ПК8(A)=1000 0111; ОК8(A)=1111 1000;

ДК8(A)=1111 1001

$B = -3_{10} = -11_2$; ПК8(B)=1000 0011; ОК8(B)=1111 1100;

ДК8(B)=1111 1101

$A+B=-10_{10}$

Биты	8	7	6	5	4	3	2	1	0
перенос	±	1	1	1	1			1	
A=-7		1	1	1	1	1	0	0	1
B=-3		1	1	1	1	1	1	0	1
A+B=-10		1	1	1	1	0	1	1	0

Результат: ДК8(A+B)=1111 0110 .

Знаковый бит – 1: число отрицательное.

Верно: Сумма двух отрицательных чисел есть число отрицательное.

~~ПК8~~(A+B)=1000 1001; ПК8(A+B)=1000 1010

Модуль $1010_2=10_{10}$.

Вывод: Получен результат: $A+B= -10$, результат верный.

5. А и В положительные, сумма $A+B > 127$.

$A=65_{10} = 1000001_2$; ПК8(A)=ОК8(A)=ДК8(A)= 0100 0001

$B=97_{10} = 1100001_2$; ПК8(B)=ОК8(B)=ДК8(B)= 0110 0001

$A+B=162_{10}$

Биты	8	7	6	5	4	3	2	1	0
перенос		1						1	
A=65		0	1	0	0	0	0	0	1
B=97		0	1	1	0	0	0	0	1
A+B=?		1	0	1	0	0	0	1	0

Результат: 1010 0010 .

Знаковый бит – 1: число отрицательное.

Невозможно: Сумма двух **положительных** чисел не может быть **отрицательным** числом.

Вывод: результат **неверный**.

Причина: Сумма $A+B=162_{10} > 127$ не может быть представлена в ДК8.

6. А и В отрицательные, сумма < -128

$V = -65_{10} = -100001_2$; ПК8(A)=11000001; ОК8(A)=10111110;

ДК8(B)=10111111

$V = -65_{10} = -110001_2$; ПК8(B)=11100001; ОК8(B)=10011110;

ДК8(B)=10011111 $A+B = -162_{10}$

Биты	8	7	6	5	4	3	2	1	0
перенос	±		1	1	1	1	1	1	
A=-65		1	0	1	1	1	1	1	1
B=-97		1	0	0	1	1	1	1	1
A+B=		0	1	0	1	1	1	1	0

Результат: ДК8=0101 1110 .

Знаковый бит – 0: число положительное.

Невозможно: Сумма двух **отрицательных** чисел не может быть **положительным** числом.

Вывод: результат **неверный**.

Причина: Сумма $A+B = -162_{10} < -128$ не может быть представлена в ДК8.

Пример.

Вычислить алгебраическую сумму 58-23.

$$58_{10} = 11\ 1010_2 = \boxed{0011\ 1010} \quad (\text{ПК8})$$

$$\begin{aligned} -23_{10} &= -1\ 0111_2 = \boxed{1001\ 0111} \quad (\text{ПК8}) \\ &= \boxed{1110\ 1000} \\ &= \boxed{1110\ 1001} \end{aligned}$$

$$\begin{array}{r} + \boxed{0011\ 1010} \\ \boxed{1110\ 1001} \\ \hline \boxed{\oplus 0010\ 0011} \end{array}$$

$$\text{ДК8} = \boxed{0010\ 0011} = 35$$

Пример.

Вычислить алгебраическую сумму $26+(-34)$

$$26_{10} = 11011_2 = \boxed{0001\ 1010} \text{ ДК8(+26)}$$

$$\begin{aligned} -34_{10} &= -10010_2 = \boxed{1010\ 0010} \text{ ПК8(-34)} \\ &= \boxed{1101\ 1101} \text{ СК8(-34)} \\ &= \boxed{1101\ 1110} \text{ ДК8(-34)} \end{aligned}$$

$$\begin{array}{r} + \boxed{0001\ 1010} \\ \boxed{1101\ 1110} \\ \hline \boxed{1111\ 1000} \text{ (ДК)} \\ \boxed{1000\ 0111} \text{ ~~СК8~~: Инверсия} \\ \boxed{1000\ 1000} \text{ ПК8: Прибавление 1} \end{array}$$

Сумма=-8

Пример:

Провести вычисления в дополнительном коде (ДК)
чисел:

$$73 + (-39) =$$

$$45 + (-87) =$$