МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ

БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

КАФЕДРА РАДИОЭЛЕКТРОННЫХ СРЕДСТВ

Лабораторная работа №1

по дисциплине «Цифровые устройства и микропроцессоры»

Вариант 11

«СИСТЕМА

КОМАНД МИКРОПРОЦЕССОРА X86»

Выполнил студент группы ИНБб-3301-02-00 \_\_\_\_\_\_\_\_\_\_/ К.А. Чашечников/

Проверил: преподаватель \_\_\_\_\_\_\_\_\_\_\_\_/М.А. Земцов/

Киров 2025

**1.** **Цель работы:**

Изучение системы команд и способов адресации

микропроцессоров с архитектурой x86.

**2. Ход работы:**

**Вариант задания:**



**Текст программы:**

.686

.model flat, stdcall

STD\_OUTPUT\_HANDLE EQU -11

GetStdHandle PROTO, nStdHandle: DWORD

WriteConsoleA PROTO, handle: DWORD, lpBuffer:PTR BYTE, nNumberOfBytesToWrite:DWORD, lpNumberOfBytesWritten:PTR DWORD, lpReserved:DWORD

ExitProcess PROTO, dwExitCode: DWORD

.stack 100h

.data

X DW -20 ; Инициализация переменных

Y DW 54

Z DW -5

B DW 15

N DW 65518

M DW ? ; Результат вычислений

buffer DB 6 DUP(?) ; Буфер для хранения строки

consoleOutHandle DD ?

bytesWritten DD ?

message DB "Answer: ", 13, 10

lmessage DD 13

.code

Start:

mov ax, B

xor Z, ax ; Инвертирование последних 4-х битов Z и Y для Z' Y'

xor Y, ax

mov ax, Y ; Начало вычисления выражения, перемещение в 16 битный регистр значения Y

AND ax, X ; Побитовое И

xor ax, Z ; Исключающее ИЛИ

sar ax, 1 ; Деление знакового числа на 2 сдвигом

mov M, ax ; Запись в переменную

; Преобразуем значение M в строку и сохраняем результат в buffer

movsx eax, word ptr M

lea esi, buffer+5

mov ecx, 5

mov ebx, 10

; Если число отрицательное, добавляем знак минус

test eax, eax

js negative

mov byte ptr [esi-5], '+'

jns next\_digit

negative:

mov byte ptr [esi-5], '-'

neg eax

next\_digit:

mov edx, 0

idiv ebx

add dl, '0'

mov [esi], dl

dec esi

loop next\_digit

INVOKE GetStdHandle, STD\_OUTPUT\_HANDLE

mov consoleOutHandle, eax

mov edx, offset message

pushad

mov eax, lmessage

INVOKE WriteConsoleA, consoleOutHandle, edx, eax, offset bytesWritten, 0

popad

; Выводим содержимое буфера

mov edx, offset buffer

pushad

mov eax, 6

INVOKE WriteConsoleA, consoleOutHandle, edx, eax, offset bytesWritten, 0

popad

INVOKE ExitProcess, 0

end Start

**Верификация программы:**

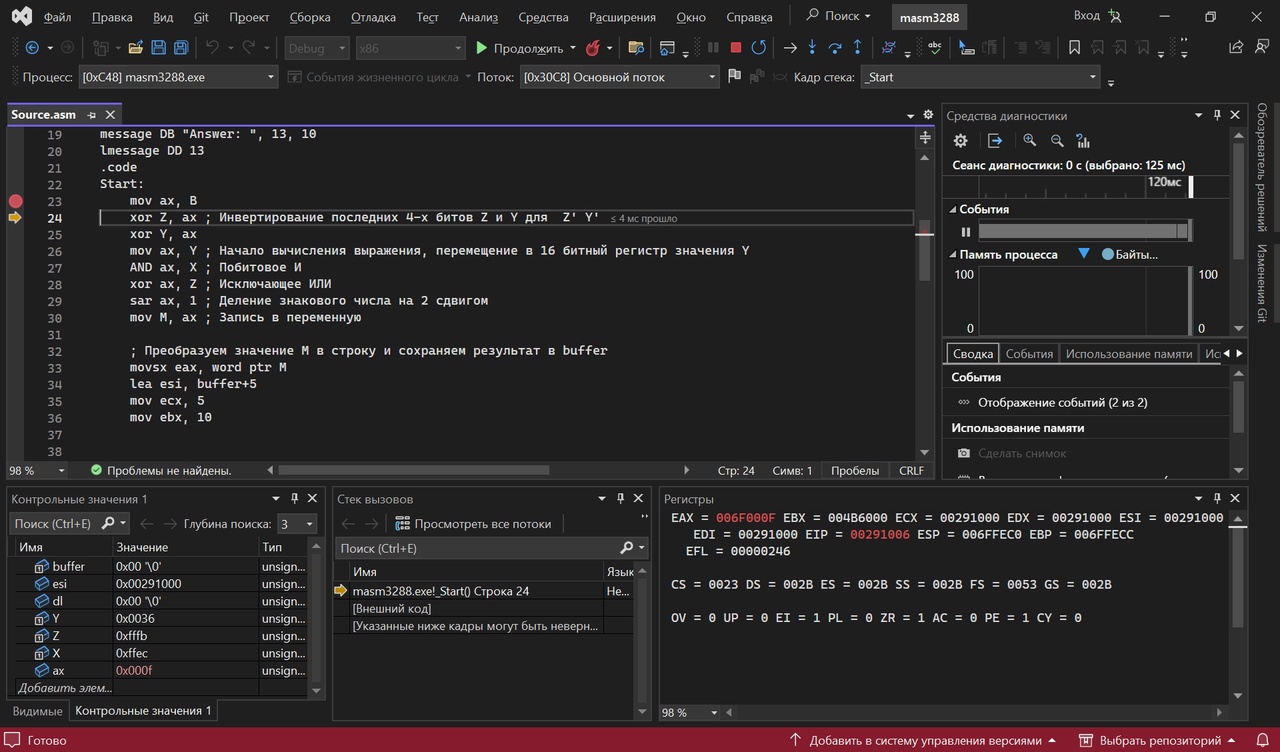


Рисунок 1 – Содержимое регистра AX после первого шага программы

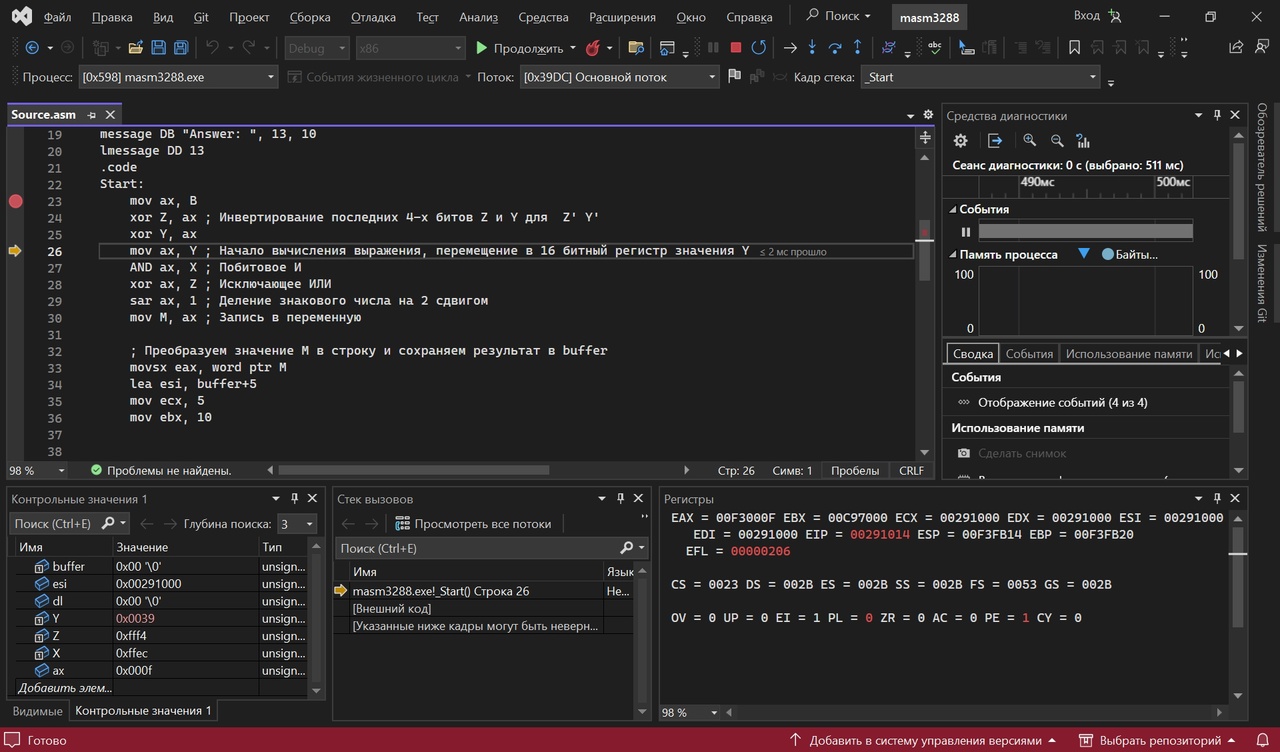


Рисунок 2 – Содержимое переменных Y и Z после инверсии младших 4 бит

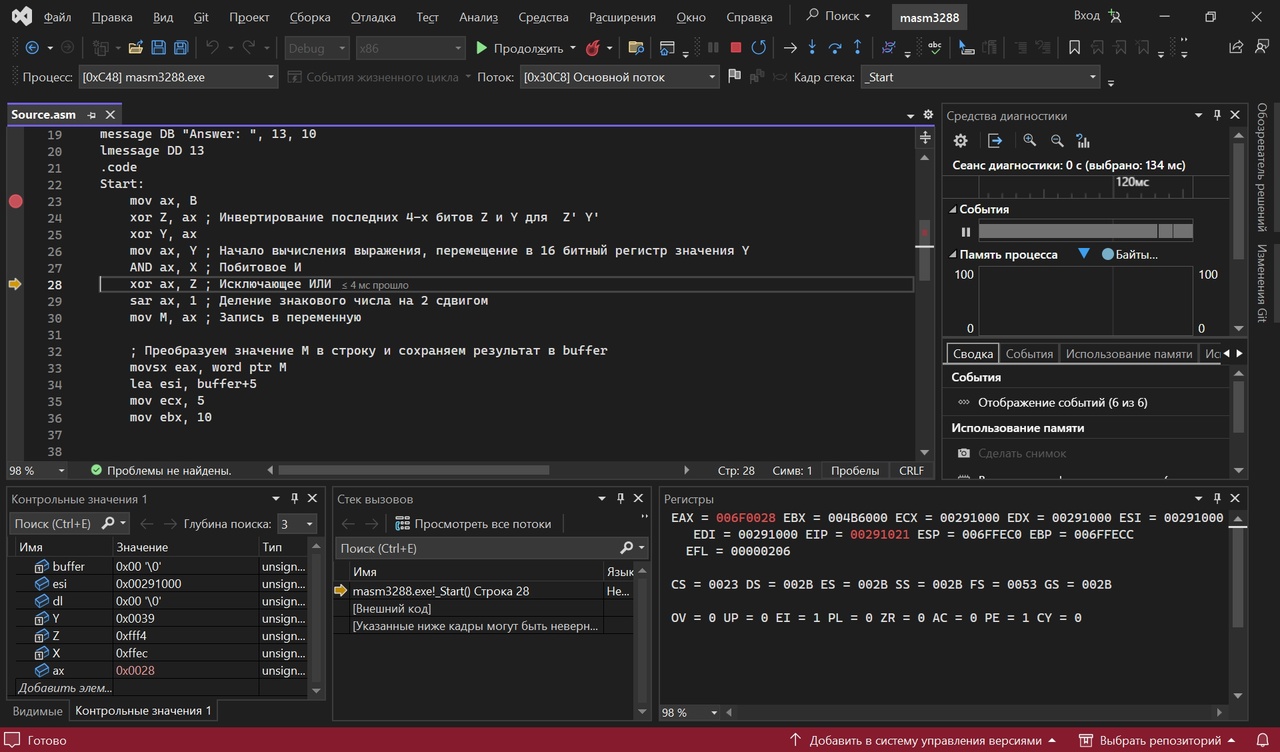


Рисунок 3 – Содержимое регистра EAX после побитового И X&Y’

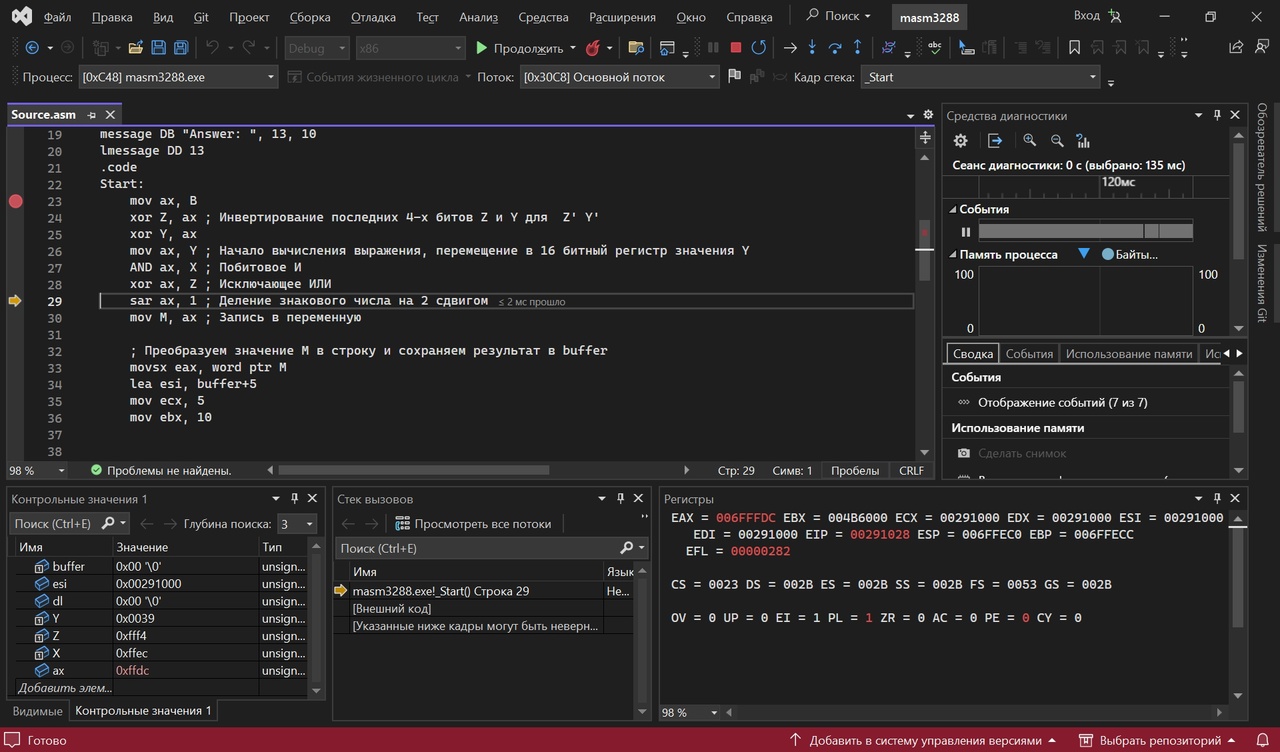


Рисунок 4 – Содержимое регистра EAX после исключающего ИЛИ

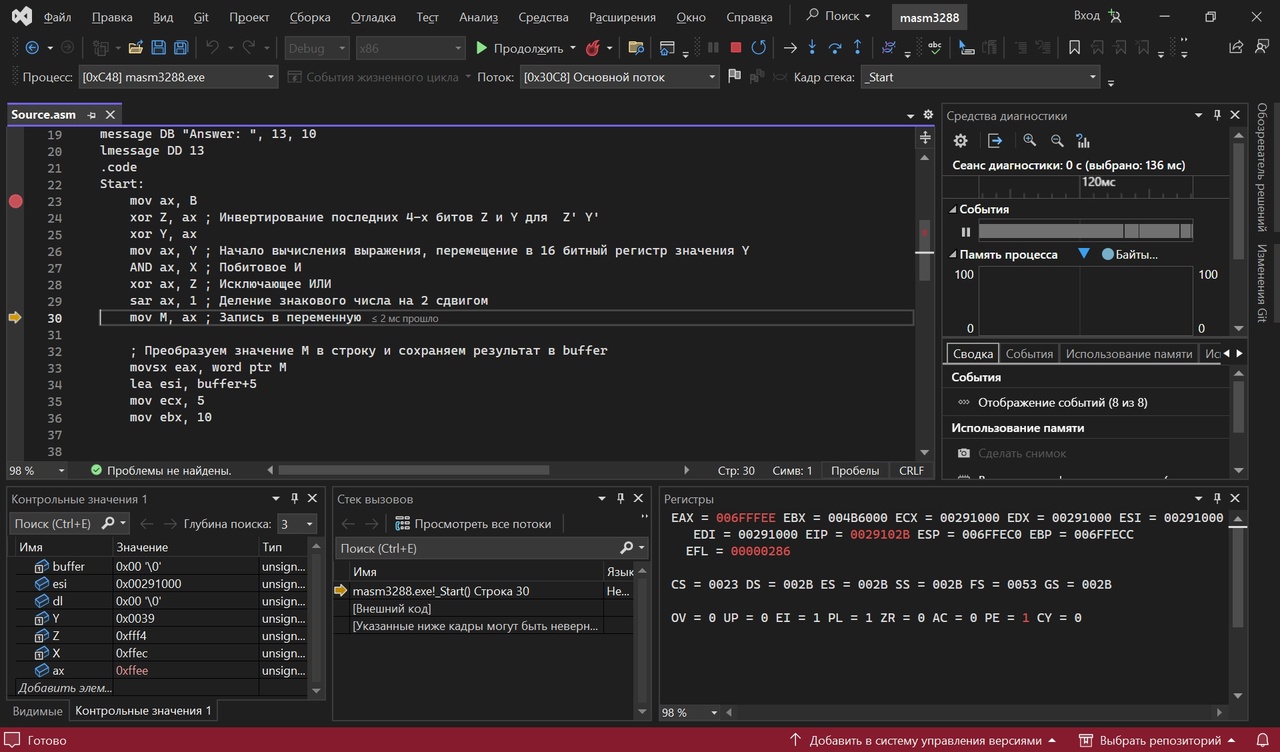


Рисунок 5 – Содержимое регистра EAX деления знакового числа на 2

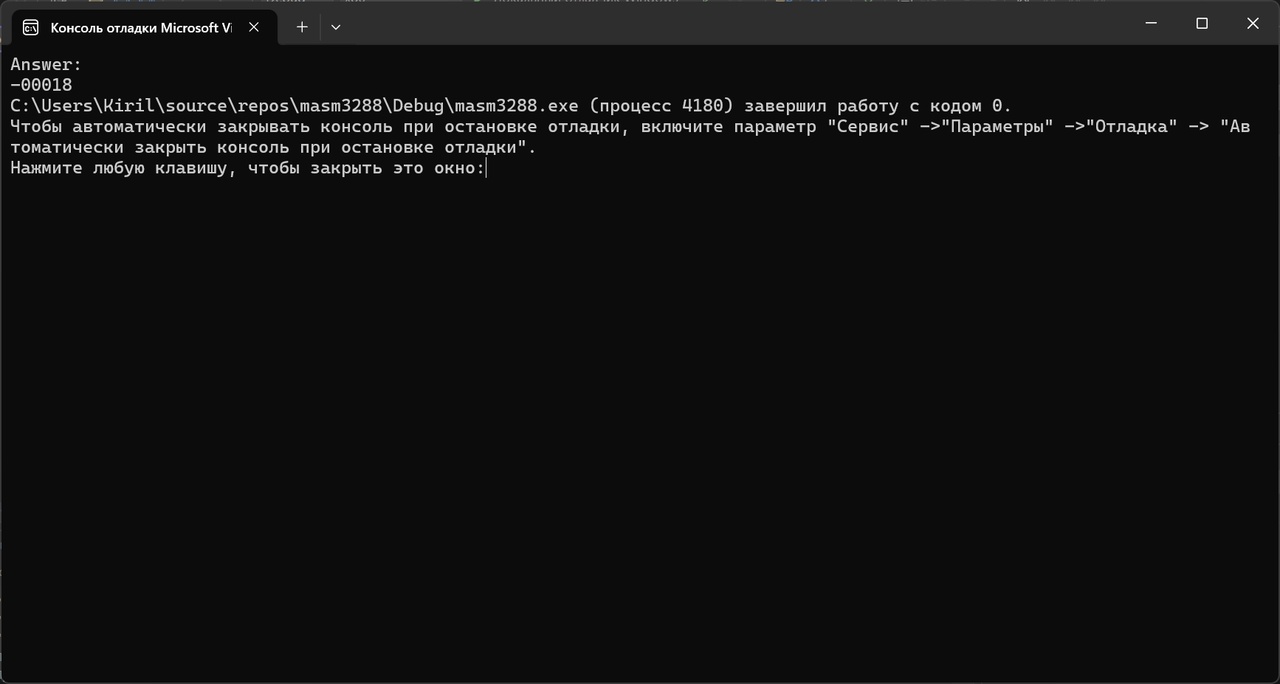


Рисунок 6 – Вывод в консоль результата вычисления

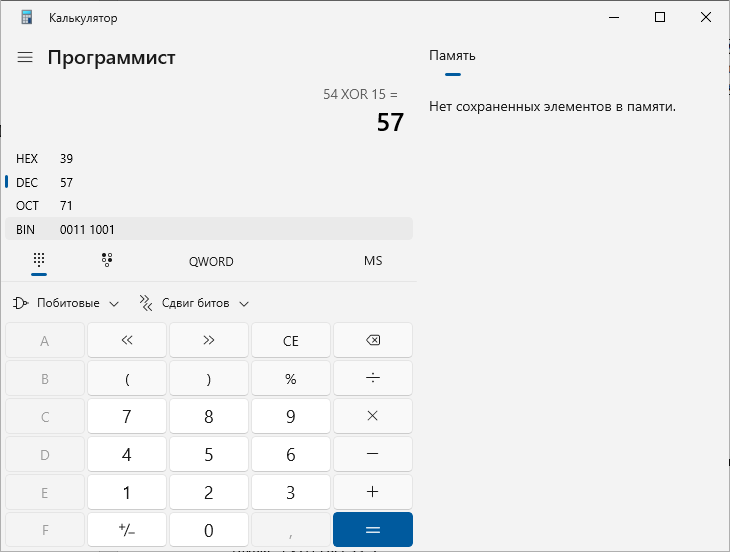


Рисунок 7 – инвертирование последних 4 битов Y

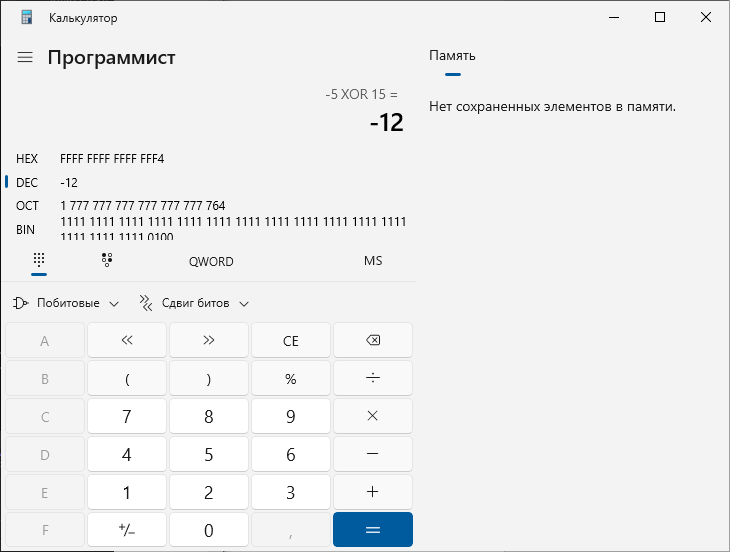


Рисунок 8 – инвертирование последних 4 битов Z

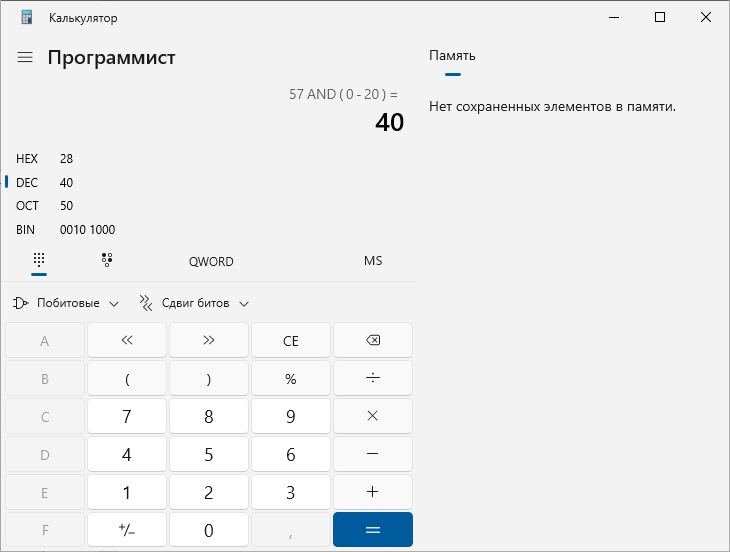


Рисунок 9 – Побитовое И Y’ и X

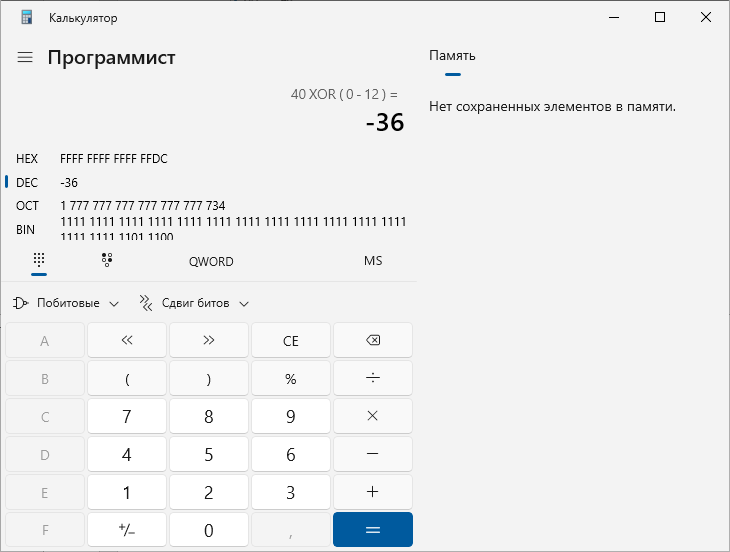


Рисунок 10 – Побитовое ИЛИ результата и Z’

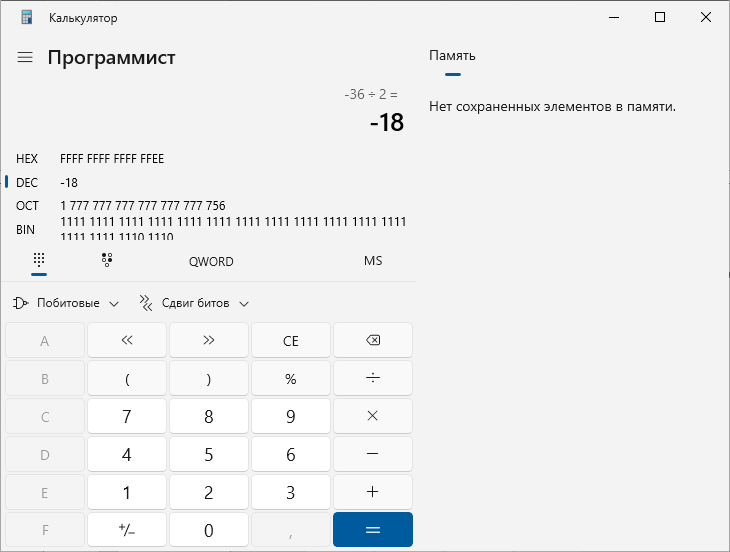


Рисунок 11 – Деление на 2

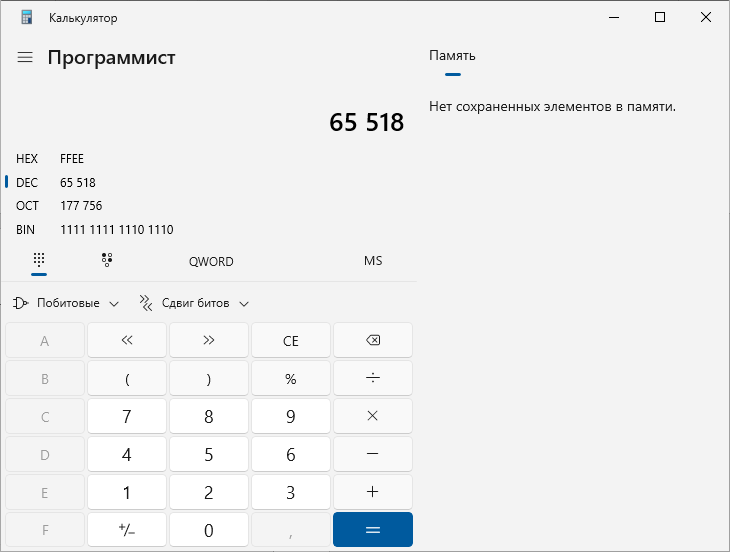


Рисунок 12 – Полученное в программе число, переведенное в другие системы