Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Дисциплина: «Организация ЭВМ и систем»

Профиль: «Программная инженерия»

ОТЧЕТ

по лабораторной работе №1

РАЗРАБОТКА АЛГОРИТМА И МИКРОПРОГРАММЫ

АРИФМЕТИЧЕСКОЙ ОПЕРАЦИИ

Вариант №13

Выполнил: студент группы РИС-19-1б

Миннахметов Э.Ю.

Проверил: доцент кафедры ИТАС

Погудин А.Л. \_\_\_\_\_\_\_\_\_\_\_

Пермь, 2021

**Постановка задачи**

Разработать алгоритм сложения 8-разрядных двоичных целых чисел из табл. 1.1 и реализовать его в виде микропрограммы для операционных автоматов арифметико-логических устройств ALU-1 и ALU-R.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Вариант** | **Коды представления** | | | | **Примеры чисел** | |
| **операнда *A*** | **операнда *B*** | **операции** | **результата** | ***A*10** | ***B*10** |
| 13 | прямой | обратный | обратный | прямой | 40 | 38 |

**Граф-схема алгоритма решения задачи**

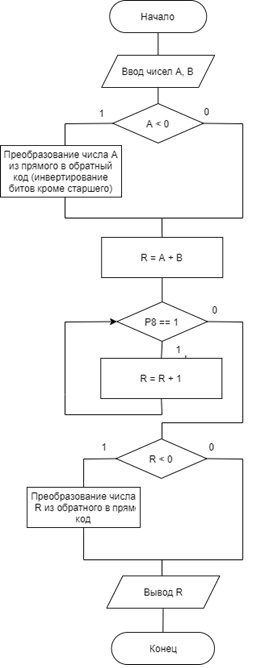
****

Рисунок 1 – Алгоритм работы программы

Запись положительного числа в прямом == обратном == прямом коде.

**Составленная программа с комментариями**

|  |  |  |
| --- | --- | --- |
| **Номер** | **Команда** | **Комментарий** |
| 1 | y14 | P0 = 0 |
| 2 | y0, y2 | Ввод числа Ax в А |
| 3 | y4, y32, y18, y24; x2, 1, nodop1 | Если число А положительное (1 бит = 0), то переход к nodop1, если отрицательное, то идем дальше |
| 4 | y26, y2 | Сдвиг числа А влево на 1 бит |
| 5 | y28 | Сдвиг вправо на 1 бит и замена первого бита на 0 |
| 6 | y4, y32, y17, y24 |
| 7 | y26, y2 |
| 8 | y5, y32, y16, y24 | Инвертирование всех битов числа А (прямой переходит в обратный код первоначального числа А) |
| 9 | y26, y2 |
| 10 | nodop1: y1, y3 | Ввод числа Bx в B |
| 11 | y14 | P0 = 0 (обнуление) |
| 12 | y4, y6, y8, y16, y24; x1, 1, skipadd | Сумма чисел A и B в обратном коде. Если нет переноса старшего бита, то идем к skipadd |
| 13 | repadd: y26, y3 | Если есть перенос старшего бита, то прибавляем единицу до тех пор, пока переноса не будет. |
| 14 | y15 |
| 15 | y6, y8, y16, y24; x1, 0, repadd |
| 16 | skipadd: y26, y2 | Запись результата в С |
| 17 | y4, y32, y18; x2, 1, nodop2 | Если число С положительное (1 бит = 0), то переход к nodop3, если отрицательное, то идем дальше |
| 18 | y5, y32, y16, y24 | Инвертирование всех битов числа С (обратный переходит в прямой код результирующего значения) |
| 19 | nodop2: y21 | Конец |

**Результаты работы программы**

На входе Ax: 10000011 (-3 в прямом коде)

На входе Bx: 11111100 (9 в обратном коде)

На выходе С: 00000110 (6 в прямом коде)

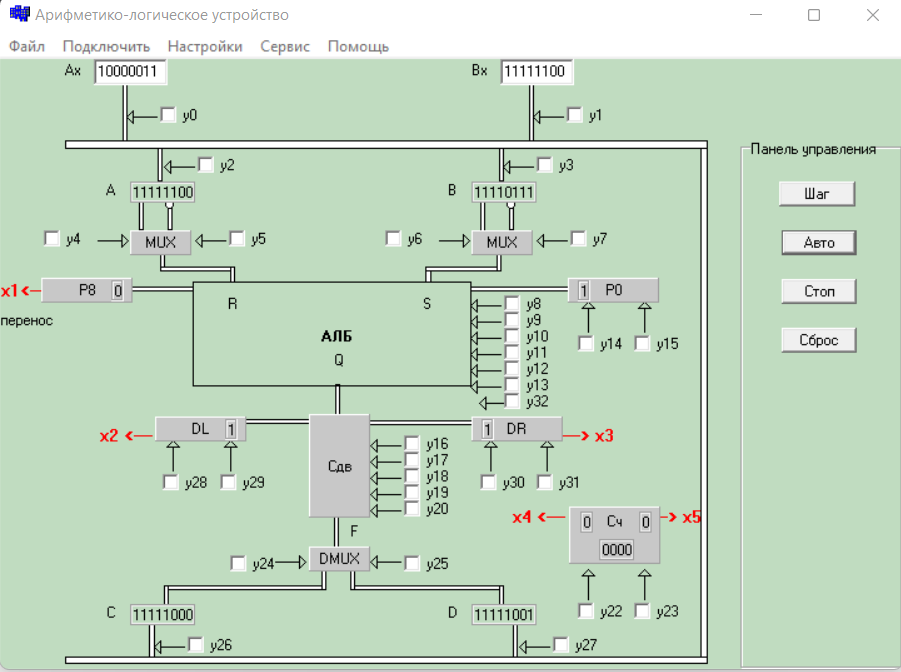
****

Рисунок 2 – Пример выполнения программы 1

На входе Ax: 00001011 (11 в прямом коде)

На входе Bx: 00010100 (2 в обратном коде)

На выходе С: 00001101 (13 в прямом коде)

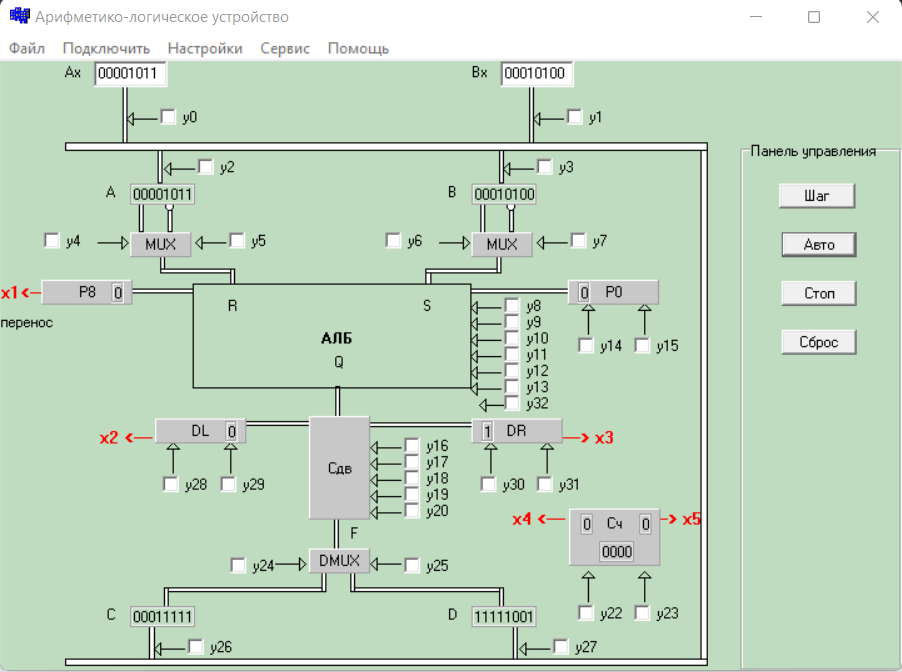
****

Рисунок 3 – Пример выполнения программы 2

На входе Ax: 10001011 (-11 в прямом коде)

На входе Bx: 00010100 (20 в обратном коде)

На выходе С: 00001001 (9 в прямом коде)

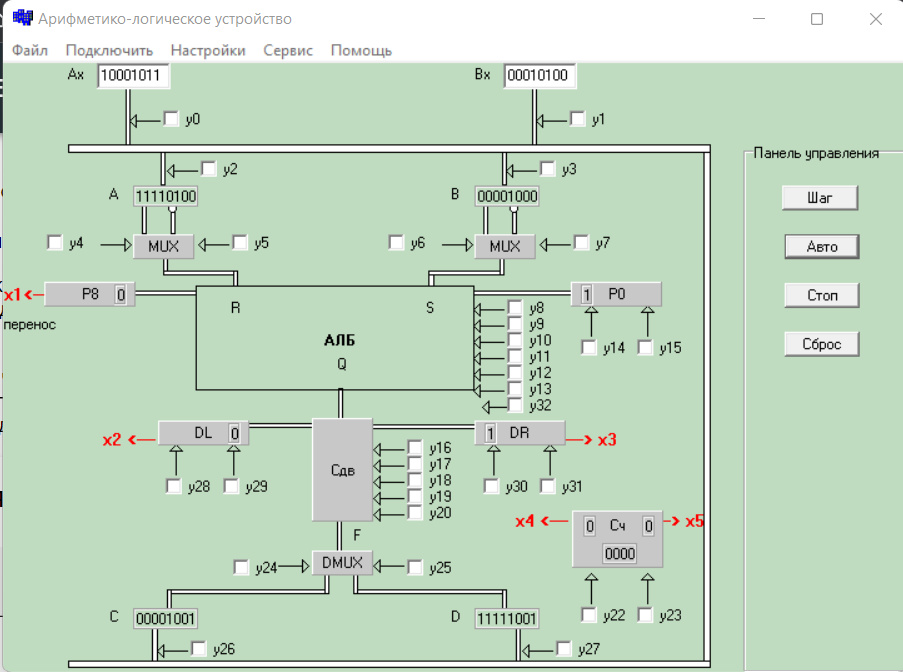
****

Рисунок 4 – Пример выполнения программы 3

На входе Ax: 00001011 (11 в прямом коде)

На входе Bx: 00010100 (-20 в обратном коде)

На выходе С: 11110111 (-9 в прямом коде)

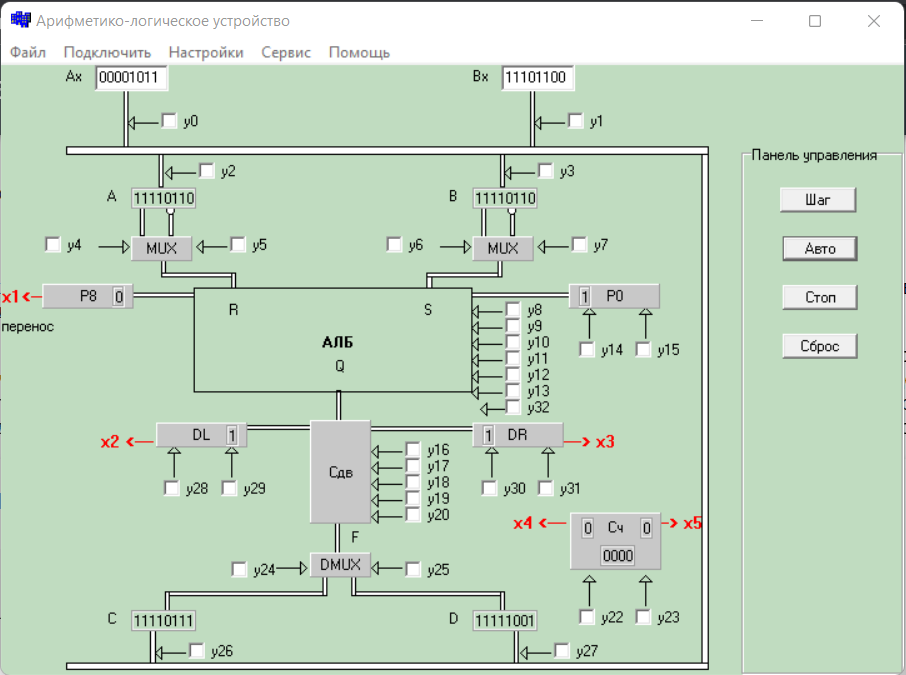
****

Рисунок 5 – Пример выполнения программы 4

На входе Ax: 01010110 (86 в прямом коде)

На входе Bx: 00001010 (10 в обратном коде)

На выходе С: 01100000 (96 в прямом коде)

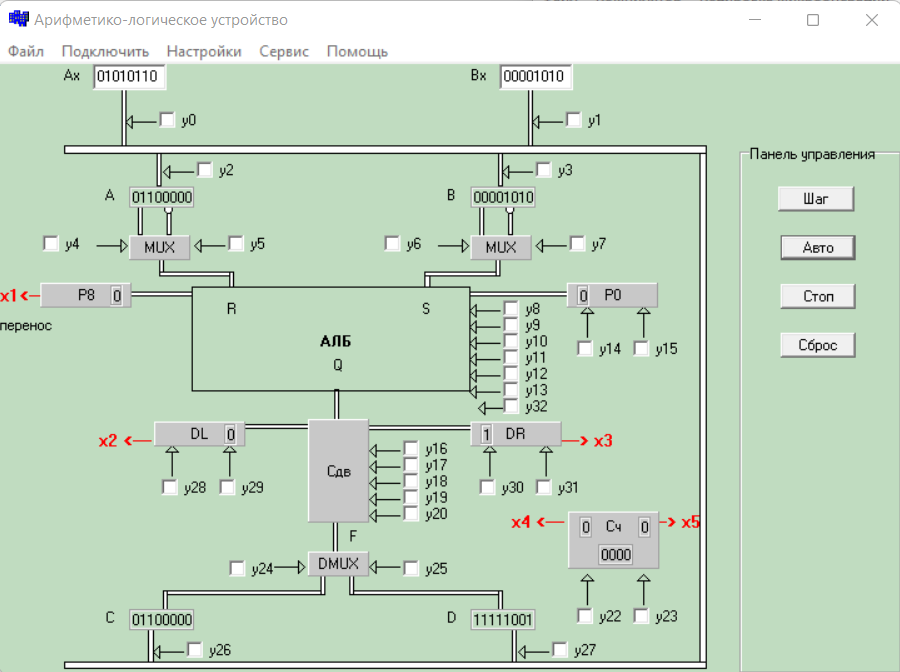
****

Рисунок 6 – Пример выполнения программы 5