МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»**

|  |
| --- |
|  |

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**Кафедра информационной безопасности**

**Арифметико-логические основы цифровых вычислительных машин**

**Курсовая работа**

на тему «**Проектирование Арифметико-логического устройства**»

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Выполнил студент гр. Б8216а | | |  | Гусев М.Д. | |  | | | Проверил ст. преподаватель кафедры ИСУ | | |  | Капецкий И.О. | | дата | | |  | | | зачтено/не зачтено | | |

г. Владивосток

2018 г.

**Содержание**

1. Введение\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_стр. 3

* Цели и задачи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ стр. 5

1. Проектирование АЛУ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ стр. 6

* Общая структура АЛУ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ стр. 6
* Требуемая функциональность\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ стр. 8
* Входные и выходные данные\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ стр. 9
* Примеры работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ стр. 14

1. Заключение\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ стр. 17

**Введение**

В настоящее время степень удовлетворения жизненных по­требностей каждого человека определяет развитие техники и технологий. Чем более они совершенны, тем более качественную и более надежную продукцию можно изготавливать, тем луч­шим является развитие всех отраслей промышленности, тем бо­лее состоятельно живет каждый человек и государство в целом. Поэтому основной задачей на нынешнем этапе развития обще­ства является совершенствование техники и технологий.

За время развития человеческого общества техника и техноло­гии прошли сложный путь от примитивных орудий труда до слож­ных устройств, которые заменяют физический труд человека ма­шинным, от отдельных несложных операций до современных способов обработки материалов и изготовления изделий.

Мир сейчас находится на пороге информационного общества. Началом такого перехода стало внедрение в различные сферы деятельности человека современных средств обработки и передачи информации. Переход от индустриального общества к информационному осуществляется благодаря информатизации общества – процессу, при котором создаются условия, удовлетворяющие потребности любого человека в получении необходимой информации. Основную роль, в информационном обществе, будет играть система распространения, хранения и обработки информации, образуя информационную среду, которая может обеспечить любому человеку доступ ко всей информации. Трудно назвать другую сферу человеческой деятельности, которая развивалась бы столь стремительно и порождала бы такое разнообразие проблем, как информатизация и компьютеризация общества. История развития информационных технологий характеризуется быстрым изменением концептуальных представлений, технических средств, методов и сфер применения.

Еще десять лет назад бесспорным казался лозунг «Программирование - вторая грамотность», под которым подразумевалось умение каждого образованного человека создавать алгоритмы и программировать в своей предметной области на языках FORTRAN, BASIC и т.д. В современных реалиях весьма актуальным для большинства людей стало не столько программирование (в старом смысле слова), сколько умение пользоваться информационными технологиями. Проникновение компьютеров во все сферы жизни общества убеждает в том, что культура общения с компьютером становится частью общей культуры человека - термины «Word», «Excel», «Internet» стали такими же обыденными, как «телефон» или «шахматы» Информационные технологии имеют свои фундаментальные разделы: архитектура персонального компьютера, операционные системы, теоретическое программирование и др.

Мир меняется так быстро, что многие не успевают приспосабливаться к переменам. Люди, жившие в начале ХХ века, вряд ли могли себе представить нынешние технологические достижения, воспринимаемые нами как должное: самолеты, роботы, спутниковые телефоны, телевизоры и т.д. Технологии действительно могут изменять общество глубже и быстрее, чем можно себе вообразить. Всякая деятельность осуществляется по технологии, определяемой целью, предметом, средствами, характером операций и результатами.

Информационные технологии - использование компьютеров для хранения, обработки и передачи знаний, а также способы создания, фиксации, переработки и распространения информации. В постмодернистских обществах использование информационных технологий является центральным фактором экономики, основой информационной революции, в рамках которой возможность переработки информации сама по себе является богатством.

Как и всякая технология, информационные технологии, предполагающие технологическое применение вычислительной техники, других технических средств, включает определенный набор материальных средств (носители информации, технические средства измерения ее состояний, обработки, передачи и т.д.) и способы их взаимодействия, специалистов и совокупность определенных методов организации работы. Целью создания и широкого распространения информационных технологий является решение проблемы информатизации общества (внедрения комплекса мер, направленных на обеспечение полного и своевременного использования достоверной информации, обобщенных знаний во всех социально значимых видах человеческой деятельности).

Информатизация является реакцией общества на существенный рост информационных ресурсов и потребность в значительном увеличении производительности труда в информационном секторе общественного производства. Как показывает практика промышленно развитых стран (США, Англии, Японии), решение проблемы информатизации общества является глобальной целью развития и связывается с выходом страны в третьем тысячелетии на новый уровень цивилизации.

В процессе развития технологии избавляют людей от тяжелого физического труда и выводят их на более качественный уровень. Считается, что широкомасштабное внедрение новых информационных технологий повысит результативность решений, принимаемых на всех уровнях управления. Это обеспечит, в свою очередь, не только рост экономических показателей развития народного хозяйства, но и достижение качественно нового уровня открытий в фундаментальных и прикладных науках, направленных на развитие производства, создание новых рабочих мест, повышение жизненного уровня населения, защиту окружающей среды.

В данной сфере технологии развиваются последние восемьдесят лет. И неизвестно что было бы, если Алан Тьюринг в 1936 году не изобрел бы дискретный автомат, способный реализовать любой алгоритм. Но это было лишь начало развития всего разнообразия технологий, которые мы имеем и в современном мире.

Одним из самых больших прорывов в этой сфере мы обязаны венгеро-американскому математику Джону фон Нейману. В 1945 году он опубликовал первые научные работы по новому компьютеру, названному Electronic Discrete Variable Automatic Computer(EDVAC). Уже через год он со своими коллегами приступил к разработке нового компьютера в Принстонском институте перспективных исследований (IAS). Архитектура этого компьютера позже стала прототипом архитектур большинства последующих компьютеров. В своих работах фон Нейман указывал устройства, которые, как он считал, должны присутствовать в компьютерах. Одно из них было особенно примечательно – арифметико-логическое устройство (АЛУ). Фон Нейман считал, что арифметико-логическое устройство необходимо для компьютера, потому что оно гарантирует, что компьютер будет выполнять базовые математические операции, включая сложение, вычитание, умножение и деление.

С нынешним уровнем развития информационных технологий, обладая нужными знаниями, спроектировать устройство, изобретенное фон Нейманом полвека назад, не составит труда.

Цели и задачи

Целью данной курсовой работы является освоение методики проектирования и разработки операционных и управляющих устройств ЭВМ, а также подготовки технической документации на эти устройства.

Задания на курсовую работу включают в себя некоторый набор исходных данных и ограничений для проектирования арифметико-логического устройства. Согласно варианту, оно должно выполнять операции сложения, вычитания и конъюнкции. Кроме того устройство должно содержать два флага: Z — признак нулевого результата и OV — признак арифметического переполнения.

**Проектирование АЛУ**

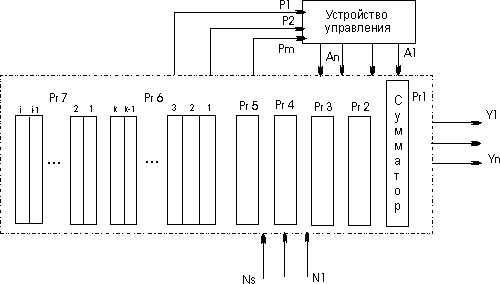
Прежде чем перейти к проектированию арифметико-логического устройства нужно для начала разобрать его структуру и функционал.

Общая структура АЛУ

АЛУ состоит из регистров, сумматора с соответствующими логическими схемами и элемента управления выполняемым процессом. Устройство работает в соответствии с сообщаемыми ему именами (кодами) операций, которые при пересылке данных нужно выполнить над переменными, помещаемыми в регистры.

Арифметико-логическое устройство функционально можно разделить на две части:

1. микропрограммное устройство (устройство управления), задающее последовательность микрокоманд (команд);
2. операционное устройство (АЛУ), в котором реализуется заданная последовательность микрокоманд (команд).

[](http://wiki.mvtom.ru/index.php/%D0%A4%D0%B0%D0%B9%D0%BB:Alu.gif)

[http://wiki.mvtom.ru/skins/common/images/magnify-clip.png](http://wiki.mvtom.ru/index.php/%D0%A4%D0%B0%D0%B9%D0%BB:Alu.gif)

Структурная схема арифметико-логического устройства

Структурная схема АЛУ и его связь с другими блоками машины показаны на рисунке. В состав АЛУ входят регистры Рг1 - Рг7, в которых обрабатывается информация, поступающая из оперативной или пассивной памяти N1, N2, ...NS; логические схемы, реализующие обработку слов по микрокомандам, поступающим из устройства управления.

Закон переработки информации задает микропрограмма, которая записывается в виде последовательности микрокоманд A1,A2, ..., Аn-1,An. При этом различают два вида микрокоманд: внешние, то есть такие микрокоманды, которые поступают в АЛУ от внешних источников и вызывают в нем те или иные преобразования информации (на рис. 1 микрокоманды A1,A2,..., Аn), и внутренние, которые генерируются в АЛУ и воздействуют на микропрограммное устройство, изменяя естественный порядок следования микрокоманд. Например, АЛУ может генерировать признаки в зависимости от результата вычислений: признак переполнения, признак отрицательного числа, признак равенства 0 всех разрядов числа др. На рис. 1 эти микрокоманды обозначены р1, p2,..., рm.

Результаты вычислений из АЛУ передаются по кодовым шинам записи у1, у2, ...,уs, в ОЗУ. Функции регистров, входящих в АЛУ:

* Рг1 - сумматор (или сумматоры) - основной регистр АЛУ, в котором образуется результат вычислений;
* Рг2, РгЗ - регистры слагаемых, сомножителей, делимого или делителя (в зависимости от выполняемой операции);
* Рг4 - адресный регистр (или адресные регистры), предназначен для запоминания (иногда и формирования) адреса операндов и результата;
* Ргб - k индексных регистров, содержимое которых используется для формирования адресов;
* Рг7 - i вспомогательных регистров, которые по желанию программиста могут быть аккумуляторами, индексными регистрами или использоваться для запоминания промежуточных результатов.

Часть операционных регистров является программно-доступной, то есть они могут быть адресованы в команде для выполнения операций с их содержимым. К ним относятся : сумматор, индексные регистры, некоторые вспомогательные регистры.

Остальные регистры программно-недоступные, так как они не могут быть адресованы в программе. Операционные устройства можно классифицировать по виду обрабатываемой информации, по способу обработки информации и логической структуре.

Требуемая функциональность

Проектируемое АЛУ будет несколько отличаться от полноценного арифметико-логического устройства своей функциональностью, потому что в изначальные задачи входит лишь три операции: сложение, вычитание и конъюнкция. Данные операции являются лишь частью всего функционала, который может выполнять полноценное АЛУ.

Арифметико-логическое устройство оперирует лишь нулями и единицами, поэтому вся информация входит и выходит в двоичной системе. Арифметические действия в двоичной системе выполняются так же, как и в десятичной. Но, если в десятичной системе счисления перенос и заём осуществляется по десять единиц, то в двоичной - по две единицы, так как в ней используются только две цифры (0 и 1).

При сложении в двоичной системе счисления двух единиц в данном разряде будет *0* и появится перенос единицы в старший разряд.

При вычитании из нуля единицы производится заём единицы из старшего разряда, где есть *1*. Единица, занятая в этом разряде, даёт две единицы в разряде, где вычисляется действие, а также по единице, во всех промежуточных разрядах.

|  |  |
| --- | --- |
| **Сложение**  0 + 0 = 0  1 + 0 = 1  0 + 1 = 1  1 + 1 = 10 | **Вычитание**   0 – 0 = 0   1 – 0 = 1   1 – 1 = 0   10 – 1 = 1 |
|  |  |
|  |  |

В отличие от сложения и вычитания, конъюнкция является сугубо логической операцией. Конъюнкция (от лат. conjunctio союз, связь) — логическая операция, по своему применению максимально приближённая к союзу "и". Синонимы: логическое "И", логическое умножение, иногда просто "И".

В проектируемом арифметико-логическом устройстве конъюнкция реализована поразрядным сравниванием значений. Только когда в сравниваемых разрядах будут единицы, то и на выходе в соответствующем разряде тоже появится единица, во всех остальных случаях на выходе будет ноль.

*Таблицы истинности для конъюнкции:*

|  |  |  |  |
| --- | --- | --- | --- |
| **a** | **b** | **a∧b** |  |
| 0 | 0 | 0 |  |
| 0 | 1 | 0 |  |
| 1 | 0 | 0 |  |
| 1 | 1 | 1 |  |
|  |  |  |  |

Также в данном АЛУ реализованы два флага:

* Флаг арифметического переполнения (OV) срабатывает тогда, когда при операции сложения или вычитания в старших разрядах целой или дробной части возникают биты, выходящие за пределы битов, изначально выделенных под выход.
* Флаг нулевого результата (Z) анализирует выходные данные. Флаг срабатывает, когда все выходящие биты, отвечающие за число, будут нулями.

Входные и выходные данные

Арифметико-логическое устройство, как говорилось выше, поддерживает лишь двоичную систему, т.е. оперирует лишь нулями и единицами, поэтому входными данными служит определенное количество нулей и единиц, в зависимости от требуемых чисел, и нужной операции. Каждый ноль или единица на входе является входящим битом.

Реализуются входные и выходные данные с помощью таблиц истинности, в которых идет перебор всех возможных входных значений (нулей и единиц), в зависимости от которых и будет выбрана нужная комбинация выходных битов.

В двоичной системе биты считаются в несколько ином порядке, но для простоты объяснения далее биты будут представлены в обычной нумерации (слева направо).

Входные данные подаются в количестве десяти битов:

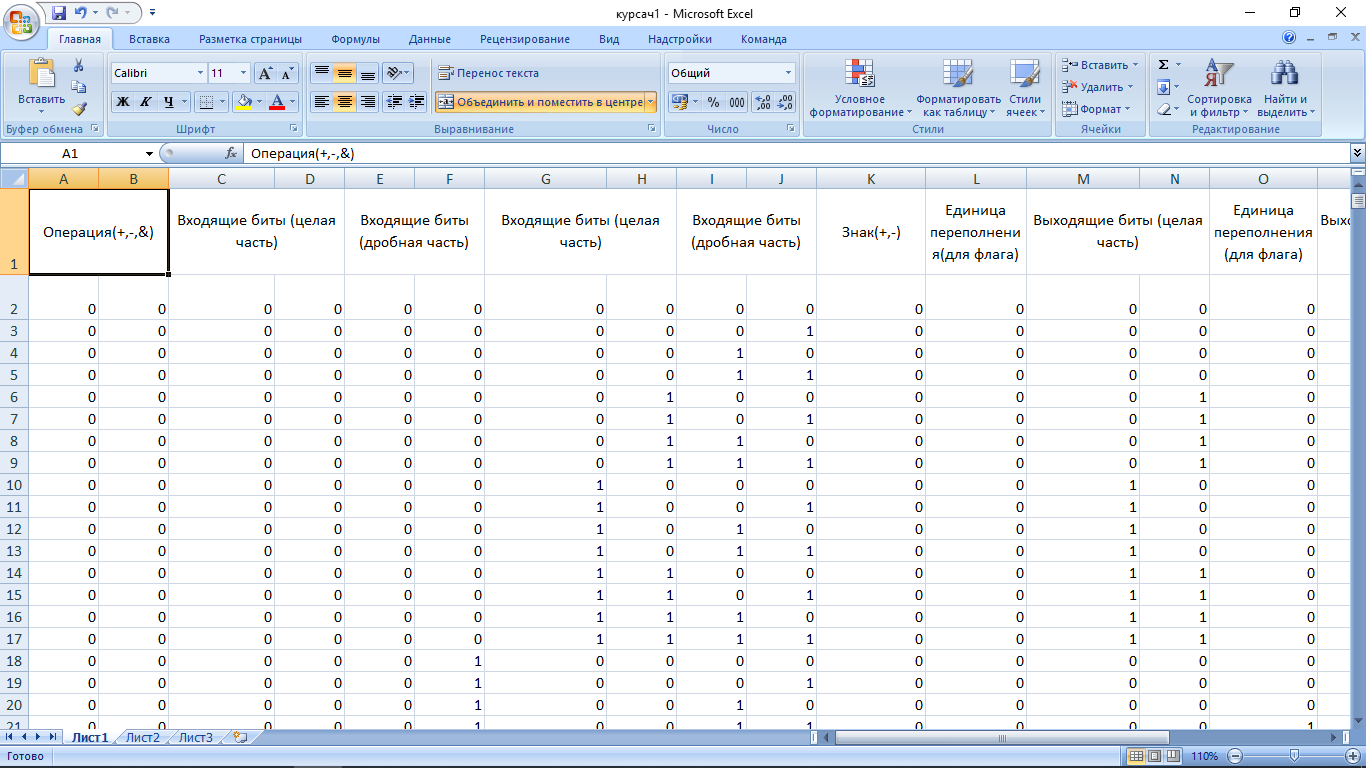
* Первые два бита отвечают за нужную пользователю операцию. Т.к. операции три, то и бит, отвечающий за операцию, не мог быть один, т.к. может содержать лишь два значения. Два бита могут содержать четыре значения, но т.к. требуется лишь три, то наибольшее значение не принимается в расчет(11). Остальные же отвечают за отведенные им операции: 00 – сложение, 01 – вычитание, 10 – конъюнкция.
* Последующие восемь битов представляют собой два числа, между которыми и будет произведена операция. Каждое число представляет собой четыре бита, два из которых отвечают за целую часть числа, а остальные два – за дробную. Соответственно максимальное числа, которые могут получиться на входе – это 3.3, которые в двоичном виде имеют вид 11.11(но в устройстве целая часть от дробной не отделяется точкой).

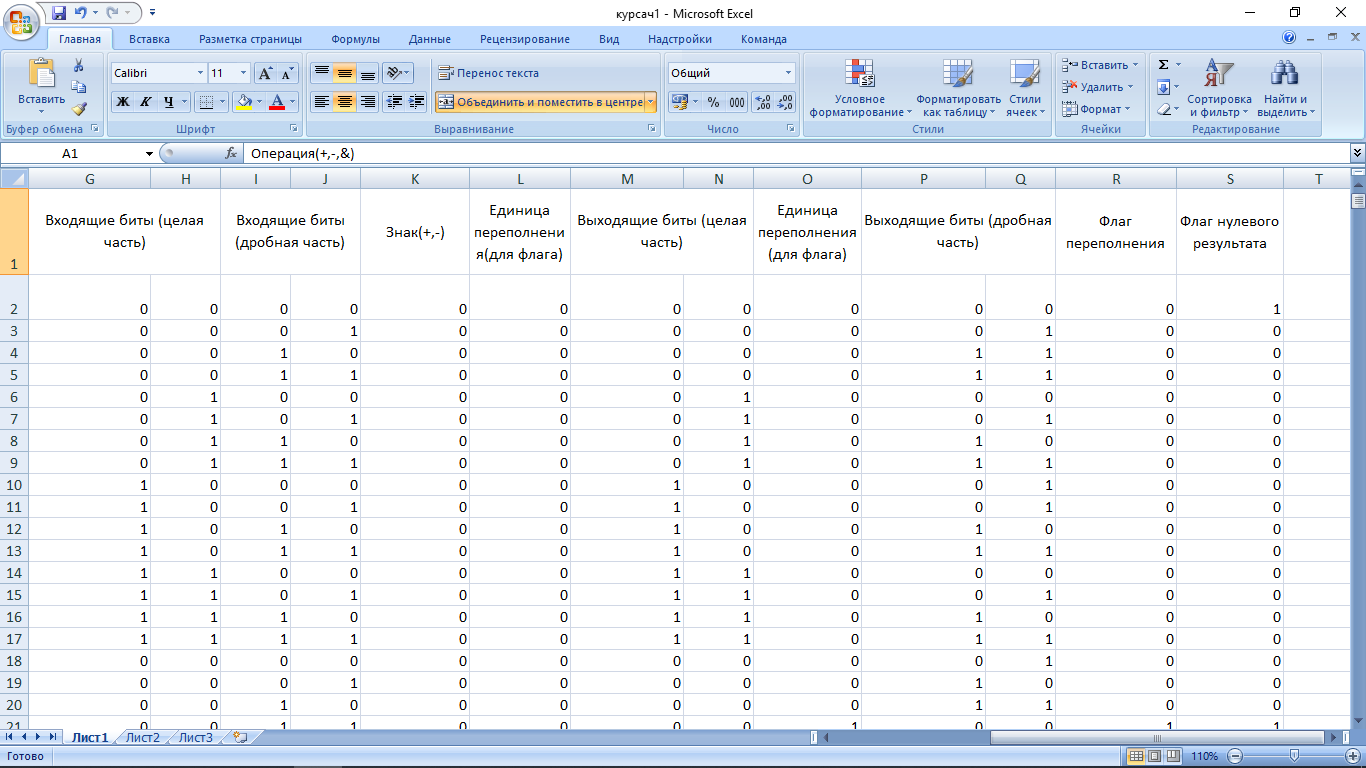
Далее идут выходные данные в количестве 7 битов:

* Первый бит на выходе отводится под знак: плюс или минус (0 и 1 соответственно).
* Затем четыре бита отводятся под число, которое будет служить ответом на операцию над входными числами. Максимальное значение, как и на входе, - это 3.3. В случае, когда в результате операции в целой или дробной частях образуется число больше 3, например 11.11+11.11 = 110.110, либо 01.00-00.10=00.1000, то и в целой и в дробной частях на выход идут два самых младших разряда (т.е. результатом двух вычислений, приведенных выше, будет 10.10 и 00.00 соответственно), а также формируется флаг переполнения.
* Далее идут два бита, отвечающие за флаги. Первый – за флаг арифметического переполнения, второй – за флаг нулевого результата.

Вид входных и выходных данных

Ниже приведен фрагмент таблиц истинности, которые для удобства составлялись в программе Microsoft Office Excel. Как и в описании выше, данная таблица заполнена таким образом, что первые два бита отводятся под операцию, последующие восемь – под два дробных числа. Потом идут выходные биты, первый из которых отводится под знак. Затем 6 битов, отвечающих за выходное число. Но самый старший разряд, как в целой, так и дробной частях не учитывается, и по данным разрядам лишь формируется флаг арифметического переполнения (т.е. в результате имеем не шесть, а четыре бита на выходе). Далее два бита отвечающие за флаги, первый – уже упомянутый, флаг арифметического переполнения, второй – флаг нулевого результата.





Согласно основополагающей модели проектирования последовательностных цифровых устройств, следует составить систему уравнений совершенной дизъюнктивной нормальной формы. Имея таблицы истинности, это не составляет труда.

Алгоритм получения СДНФ по таблице истинности:

1. Отметить те строчки таблицы истинности, в последнем столбце которых стоят 1:

|  |  |  |
| --- | --- | --- |
| *X* | *Y* | *F(X,Y)* |
| 0 | *0* | *0* |
| *0* | *1* | *1\** |
| *1* | *0* | *1\** |
| *1* | *1* | *0* |

1. Выписать для каждой отмеченной строки конъюнкциювсех переменных следующим образом: если значение некоторой переменной в данной строке равно *1*, то в конъюнкцию включать саму эту переменную, если равно *0*, то ее отрицание:

– для 2-й строки;

– для 3-й строки.

1. Все полученные конъюнкции связать в дизъюнкцию: .

Ниже приведены примеры уравнений для выходных битов.

Выходной бит, отвечающий за знак.

Операция1 ~Бит2 ~Бит1 ~ДробБит2 ~ДробБит1 ДробБит11 + Операция1 ~Бит2 ~Бит1 ~ДробБит2 ДробБит22 + Операция1 ~Бит2 ~Бит1 ~ДробБит1 ДробБит22 ДробБит11 + Операция1 ~Бит2 ~Бит1 Бит11 + Операция1 ~Бит2 ~ДробБит2 ~ДробБит1 Бит11 ДробБит11 + Операция1 ~Бит2 ~ДробБит2 Бит11 ДробБит22 + Операция1 ~Бит2 ~ДробБит1 Бит11 ДробБит22 ДробБит11 + Операция1 ~Бит2 Бит22 + Операция1 ~Бит1 ~ДробБит2 ~ДробБит1 Бит22 ДробБит11 + Операция1 ~Бит1 ~ДробБит2 Бит22 ДробБит22 + Операция1 ~Бит1 ~ДробБит1 Бит22 ДробБит22 ДробБит11 + Операция1 ~Бит1 Бит22 Бит11 + Операция1 ~ДробБит2 ~ДробБит1 Бит22 Бит11 ДробБит11 + Операция1 ~ДробБит2 Бит22 Бит11 ДробБит22 + Операция1 ~ДробБит1 Бит22 Бит11 ДробБит22 ДробБит11

Выходной бит, отвечающий за флаг арифметического переполнения.

~Операция ~Операция1 ДробБит1 ДробБит22 ДробБит11 + ~Операция ~Операция1 ДробБит2 ДробБит22 + ~Операция ~Операция1 ДробБит2 ДробБит1 ДробБит11 + ~Операция ~Операция1 Бит1 Бит22 Бит11 + ~Операция ~Операция1 Бит2 Бит22 + ~Операция ~Операция1 Бит2 Бит1 Бит11 + ~Операция Операция1 ~Бит2 ~Бит1 ДробБит1 Бит11 ~ДробБит22 ~ДробБит11 + ~Операция Операция1 ~Бит2 ДробБит1 Бит22 ~ДробБит22 ~ДробБит11 + ~Операция Операция1 ~Бит1 ДробБит1 Бит22 Бит11 ~ДробБит22 ~ДробБит11 + ~Операция Операция1 ~Бит2 ~Бит1 ДробБит2 Бит11 ~ДробБит22 + ~Операция Операция1 ~Бит2 ДробБит2 Бит22 ~ДробБит22 + ~Операция Операция1 ~Бит1 ДробБит2 Бит22 Бит11 ~ДробБит22 + ~Операция Операция1 ~Бит2 ~Бит1 ДробБит2 ДробБит1 Бит11 ~ДробБит11 + ~Операция Операция1 ~Бит2 ДробБит2 ДробБит1 Бит22 ~ДробБит11 + ~Операция Операция1 ~Бит1 ДробБит2 ДробБит1 Бит22 Бит11 ~ДробБит11 + ~Операция Операция1 Бит1 ~ДробБит2 ~ДробБит1 ~Бит22 ~Бит11 ДробБит11 + ~Операция Операция1 Бит1 ~ДробБит2 ~Бит22 ~Бит11 ДробБит22 + ~Операция Операция1 Бит1 ~ДробБит1 ~Бит22 ~Бит11 ДробБит22 ДробБит11 + ~Операция Операция1 Бит2 ~ДробБит2 ~ДробБит1 ~Бит22 ДробБит11 + ~Операция Операция1 Бит2 ~ДробБит2 ~Бит22 ДробБит22 + ~Операция Операция1 Бит2 ~ДробБит1 ~Бит22 ДробБит22 ДробБит11 + ~Операция Операция1 Бит2 Бит1 ~ДробБит2 ~ДробБит1 ~Бит11 ДробБит11 + ~Операция Операция1 Бит2 Бит1 ~ДробБит2 ~Бит11 ДробБит22 + ~Операция Операция1 Бит2 Бит1 ~ДробБит1 ~Бит11 ДробБит22 ДробБит11

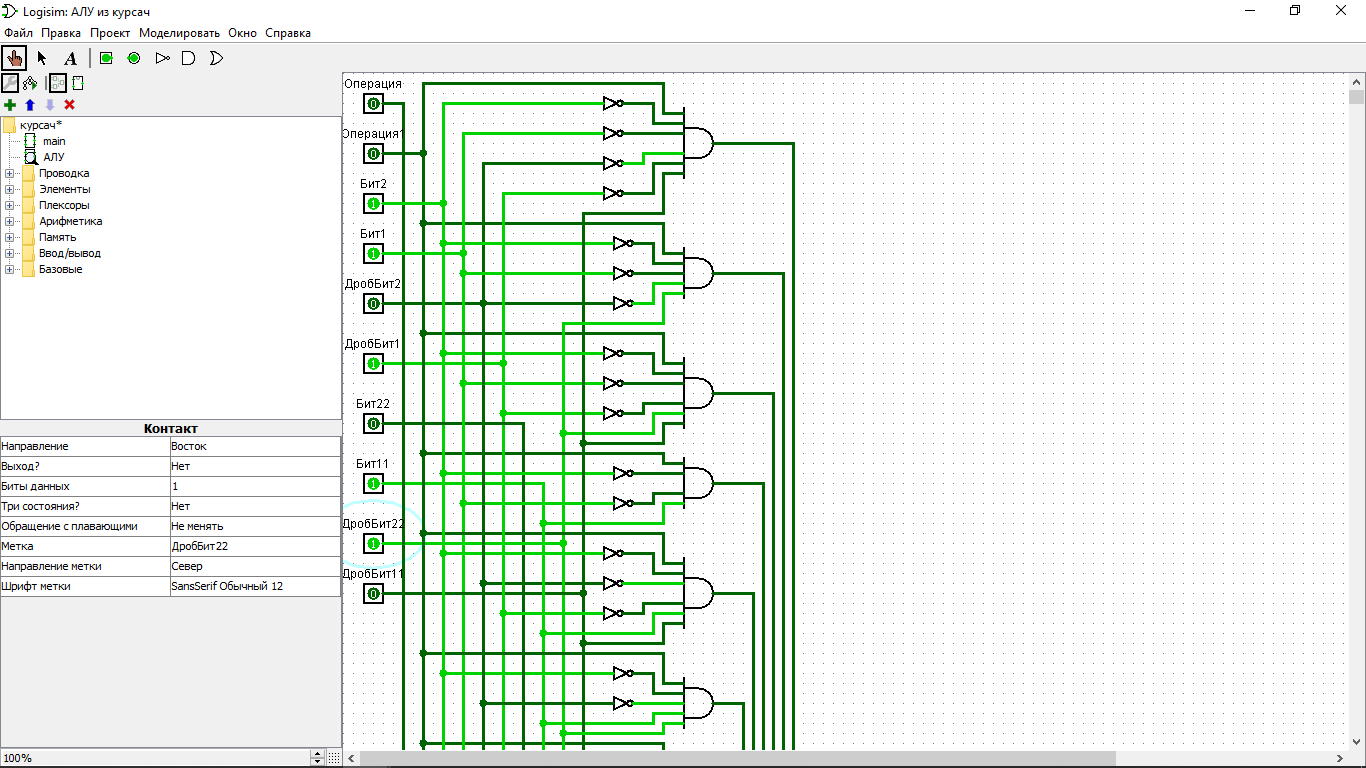
Система обозначений взята из программы logisim:

|  |  |  |
| --- | --- | --- |
| **высший приоритет** | ~ | **НЕ** |
|  | (отсутствие символа) | **И** |
|  | ^ | **Исключающее ИЛИ** |
| **низший приоритет** | + | **ИЛИ** |

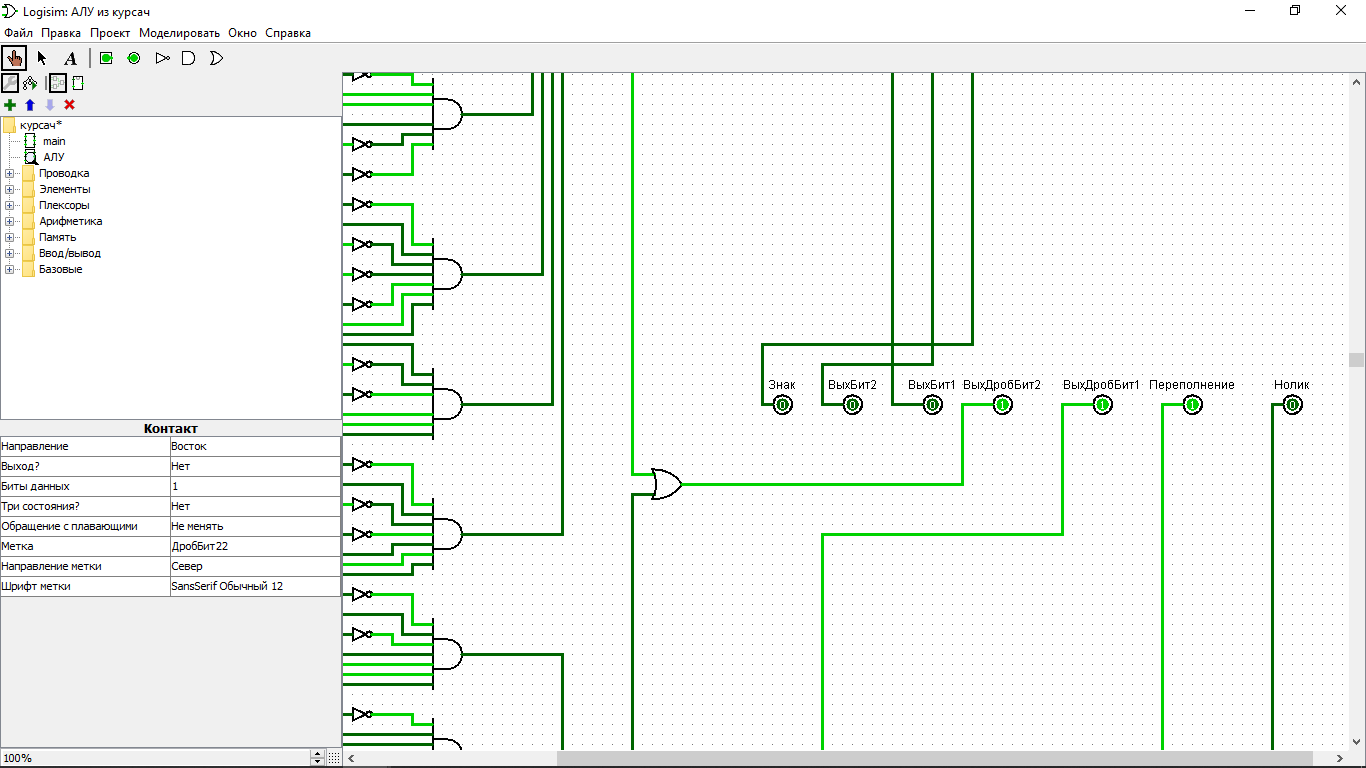
Примеры работы

1)Операция сложения(00): для примера в качестве входных чисел подадим 11.01 и 01.10.В общем виде входные биты должны подаваться в таком виде – 0011010110. В результате должно получиться 100.011, но так как берем только по два последних разряда, то итоговый результат сложения должен получиться 00.11 и активируется флаг арифметического переполнения. Результат работы программы должен выглядеть таким образом – 0001110.

Входные данные для операции сложения.

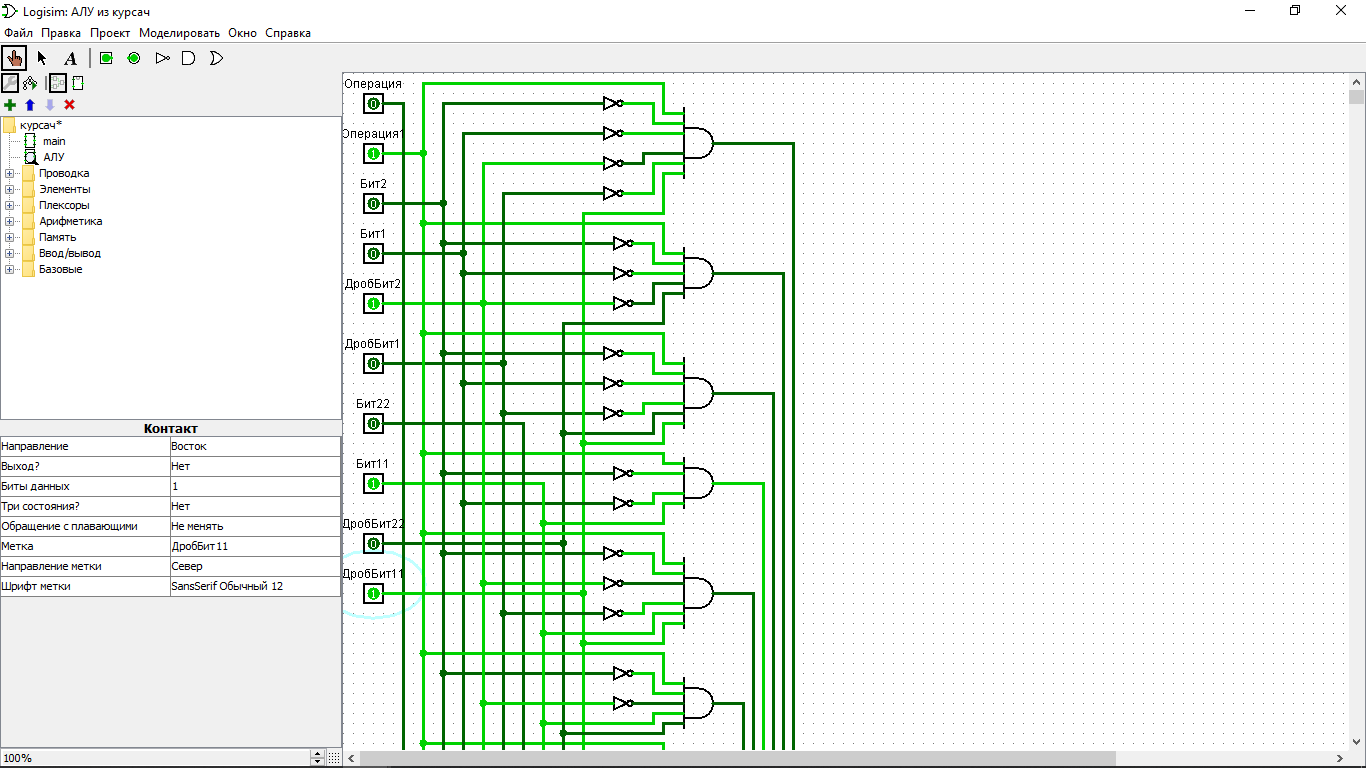


Выходные данные для операции сложения.

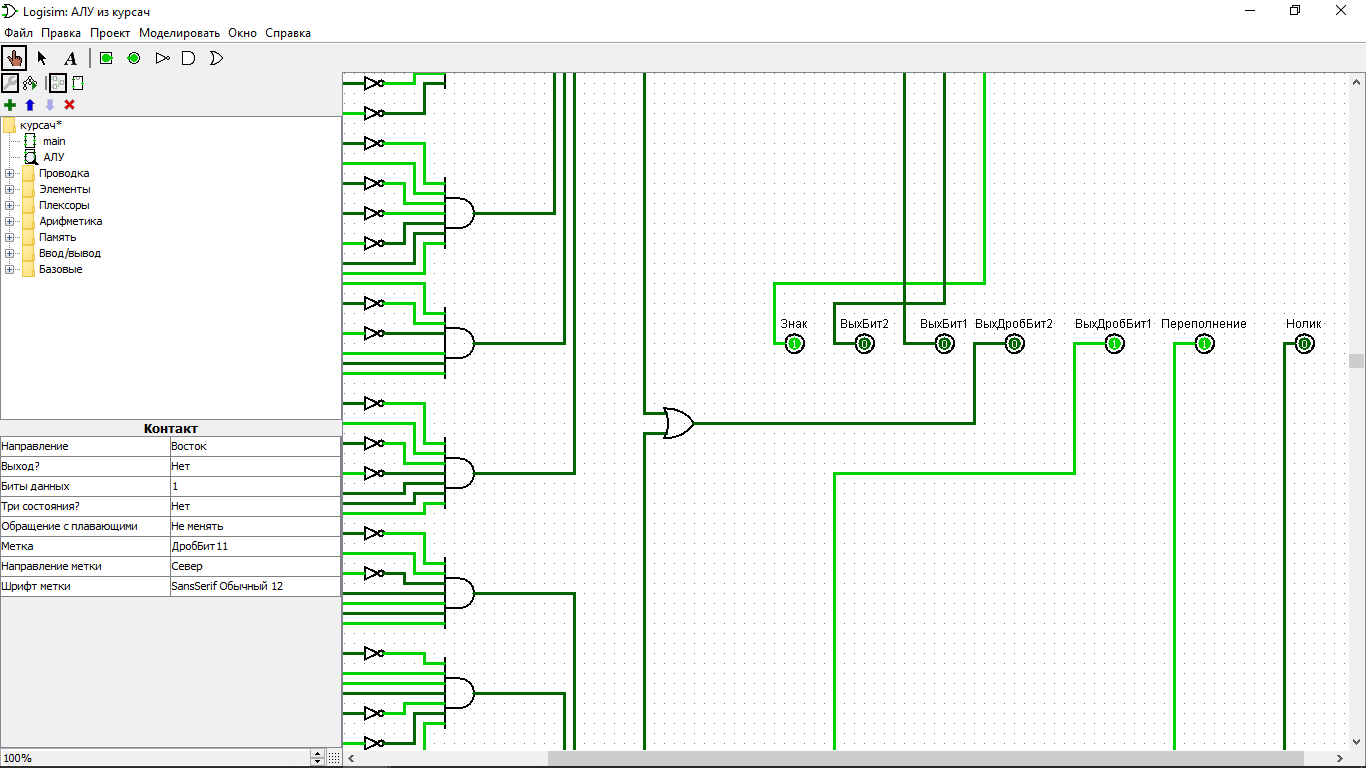


2) Операция вычитания(01): Для примера на вход подадим такие числа: 00.10 и 01.01. В общем виде на входе будет – 0100100101. Результатом данного вычисления должно получиться число -00.1001, но так как в ответ заносятся только два самых младших разряда, а минус представляется в качестве бита, то на выходе будет 1 00.01, а также активированный флаг арифметического переполнения. В общем виде на выходе будет 1000110.

Входные данные для операции вычитания.

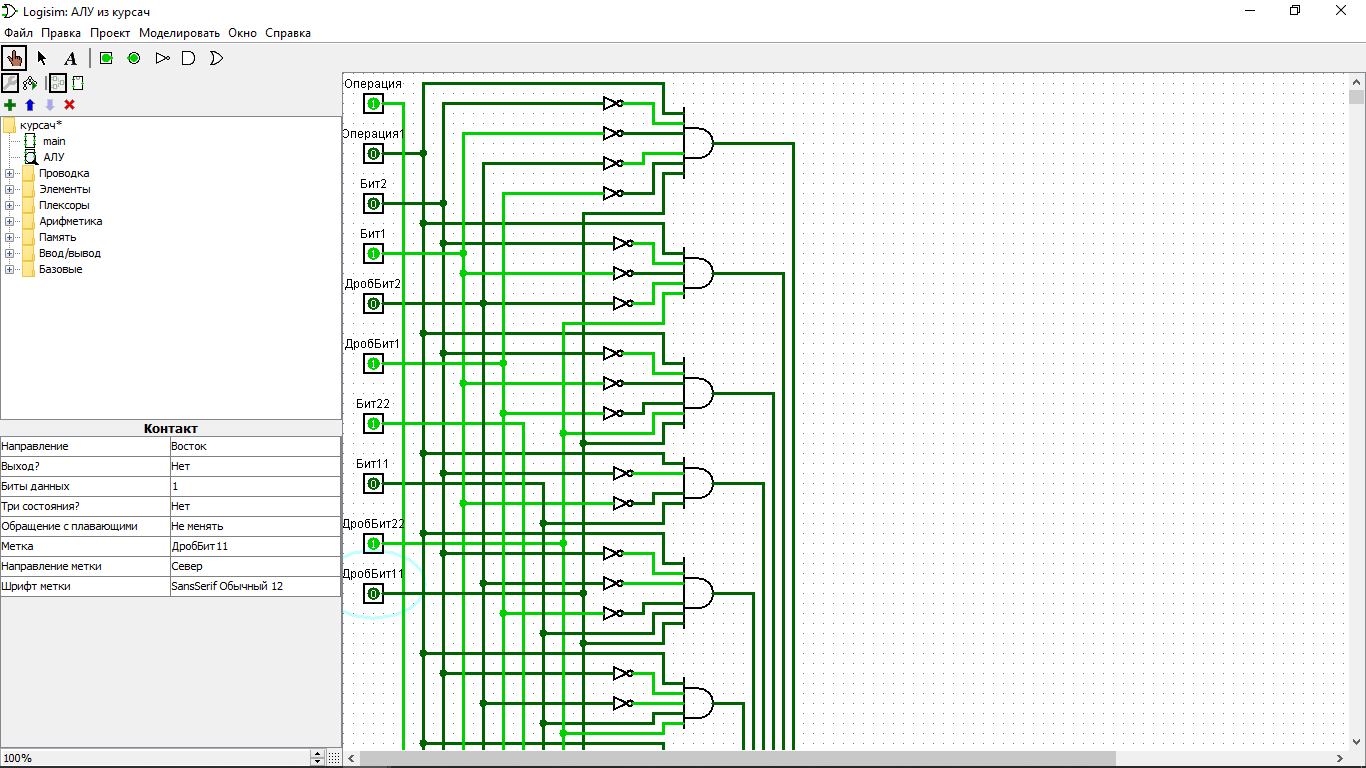


Выходные данные для операции вычитания.

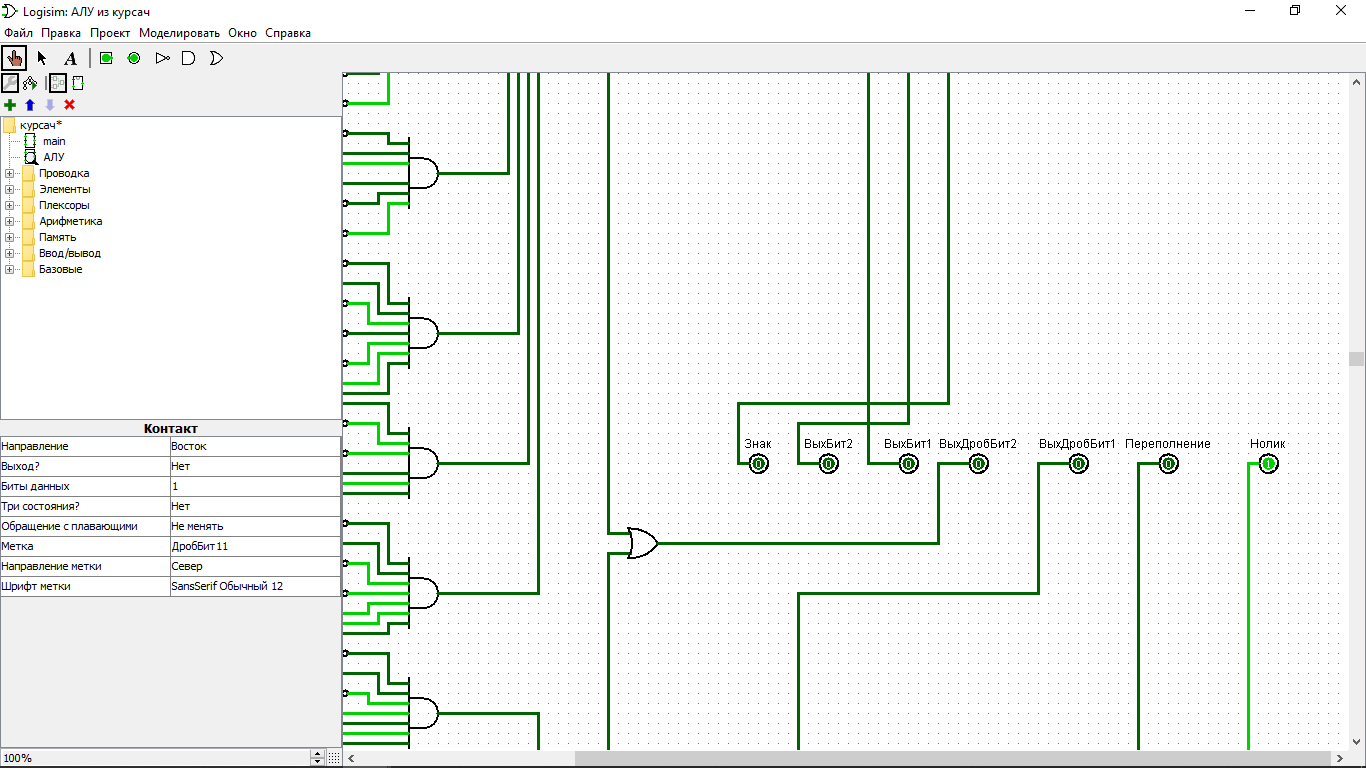


3) Конъюнкция (10): для примера подадим на вход числа 01.01 и 10.10. В общем виде – 1001011010. Так как конъюнкция подает на выход 1, только когда оба бита в одинакового разряда равны единице, а во входных данных этого не произойдет нигде, то на выходе будет 00.00, а также активированный флаг нулевого результата. В общем виде – 0000001.

Входные данные для конъюнкции.



Выходные данные для конъюнкции.



**Заключение**

В результате проделанной работы была получена схема простейшего арифметико-логического устройства с заданным функционалом. Были изучены системы структуры и функционирования арифметико-логических устройств. Разобраны и составлены алгоритмы составления уравнений совершенной дизъюнктивной формы. Обновлены и улучшены знания о работе в программе Microsoft Office Excel. Усвоены навыки по проектированию устройств в программе logisim, а также разобрана на практике совместимость данной программы с Microsoft Excel.