Оглавление

Часть I

Релейный конструктор

Глава 1

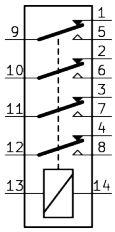
Введение

Глава 2

Релейная логика

2.1 Электромагнитное реле

В конструкторе используются электромагнитные реле с четырьмя переключающими контактами. В электрических схемах мы используем нестандартное, но наглядное обозначение таких реле:

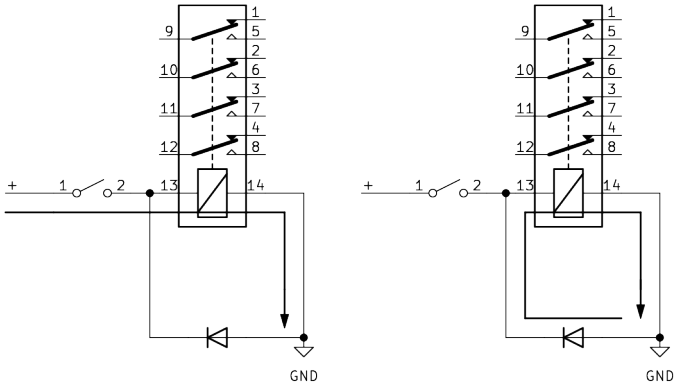


Контакты 13 и 14 подключены к катушке, управляющей состоянием контактов. Когда напряжение на катушку не подано (и ток через неё не течёт), контакты переключателей соединены так: $9-1$, $10-2$, $11-3$, $12-4$.

Если же реле включится, подвижная часть контактов сдвинется, разомкнув цепи, описанные выше, и замкнув следующие: $9-5$, $10-6$, $11-7$, $12-8$.

В каждое реле конструктора встроен светодиод, поэтому можно легко увидеть, когда оно включается. Это помогает отлаживать схемы или наблюдать хранящиеся в релейной памяти значения.

Во всех модулях параллельно каждой из катушек реле подключён диод в обратном направлении. Это сделано для предотвращения искрения контактов других реле при их размыкании. Искрение возникает из-за того, что несмотря на разрыв цепи из-за индуктивности катушки ток должен продолжать течь какое-то время. Но из-за разрыва это невозможно в обычном режиме, поэтому возникает электрическая дуга, постепенно разрушающая контакты.

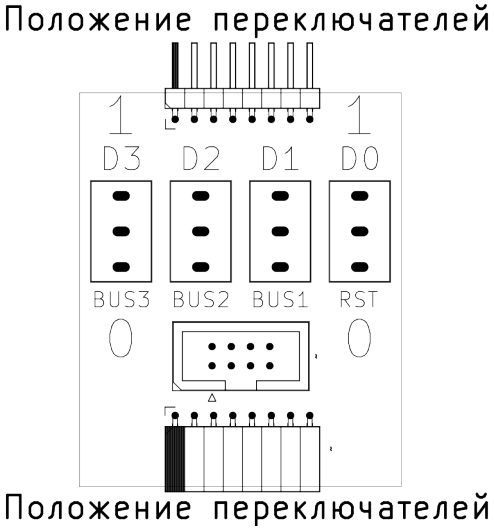


Чтобы контакты меньше портились и нужен диод. Оставшийся электрический импульс проходит уже через него, а не через разомкнутые контакты.

2.2 Логические уровни

При использовании электромагнитных реле логической единице соответствует наличие напряжения (цепь замкнута), а логическому нулю — отсутствие напряжения (цепь разомкнута).

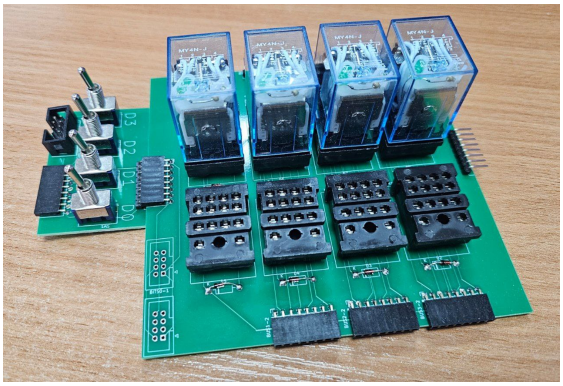
2.3 Тумблеры



Модуль с тумблерами используется для ручного включения и выключения реле. Присоединяя его к разным разъёмам, можно задавать четырёхбитное число, либо переключать до четырёх управляющих сигналов.

Проще всего проверить работу тумблеров, подключив их к управляющей шине регистрового модуля, в который вставлены только 4 реле.

2.3.1 Практикум



1. Переключать тумблеры. Убедиться, что положение одного тумблера меняет состояние одного реле.
2. Запомнить включенное и выключенное состояния тумблера, чтобы позднее не было проблем с управлением другими схемами.

2.3.2 Задачи

1. Придумать, как соединить тумблеры для выполнения логического ИЛИ.
2. Придумать, как соединить тумблеры для выполнения логического И.

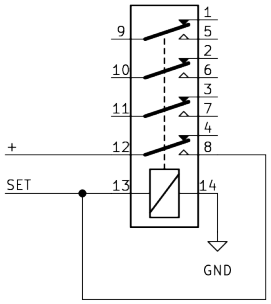
Глава 3

Хранение и передача данных

3.1 RS-триггер

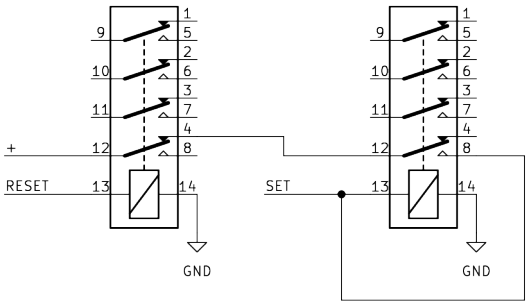
Триггер — это простейшее устройство для хранения одного бита информации. Если он включен, там хранится единица, а если выключен — хранится ноль.

Хранение одного бита можно сделать с помощью одного реле. Если оно включается, то сохранена единица. Чтобы это включённое состояние запоминалось, нужно организовать обратную связь через один из контактов. Как только реле включается с помощью входа SET, на его обмотку подаётся напряжение через его же замкнутый контакт. Поэтому даже если перестать подавать напряжение на вход SET, реле всё равно останется включённым:



Но если такое реле уже включилось, выключить его не получится. Единственный способ — это снять питание. Тогда контакты разомкнутся и при повторной подаче питания реле будет в выключенном состоянии.

Чтобы не отключать всю схему, нужно предусмотреть отдельное реле, позволяющее отключить питание только для этой схемы:



Теперь подав сигнал на вход $RESET$ можно сбросить триггер. И там опять окажется ноль.

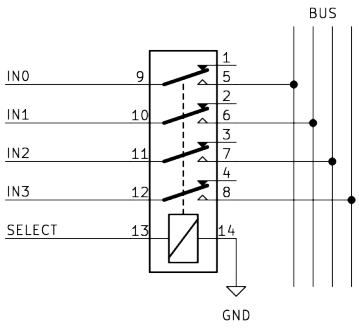
Триггер с двумя входами $RESET$ и SET называется RS-триггером. На практике удобно объединить несколько таких триггеров в многобитовый регистр и использовать для них единый сигнал сброса, сэкономив несколько реле.

3.2 Шина

Шина — это набор проводников, объединённых общим назначением. К этим проводникам подключаются несколько устройств для обмена сигналами.

Например, по шине данных процессор может читать или записывать данные из памяти или жёсткого диска. Шина адреса используется для выбора определённой ячейки в памяти. А по шине управления приходят сигналы, позволяющие отличать обращение к памяти от обращения к диску.

Так как к шине обычно подключается больше одного входа и больше одного выхода, необходимо изолировать лишние устройства, когда шина используется другими. В релейном компьютере мы можем использовать для подключения к четырёхбитной шине шинный формирователь из одного реле:



Когда такое реле включается, оно может соединить какое-то устройство (например, регистр) с шиной. Подключение будет двунаправленным, поэтому с помощью этой схемы можно как записывать данные в регистр, так и считывать оттуда.

Шинные формирователи подключаются к каждому устройству, поэтому для передачи данных и вычислений требуется формировать множество сигналов $SELECT$.

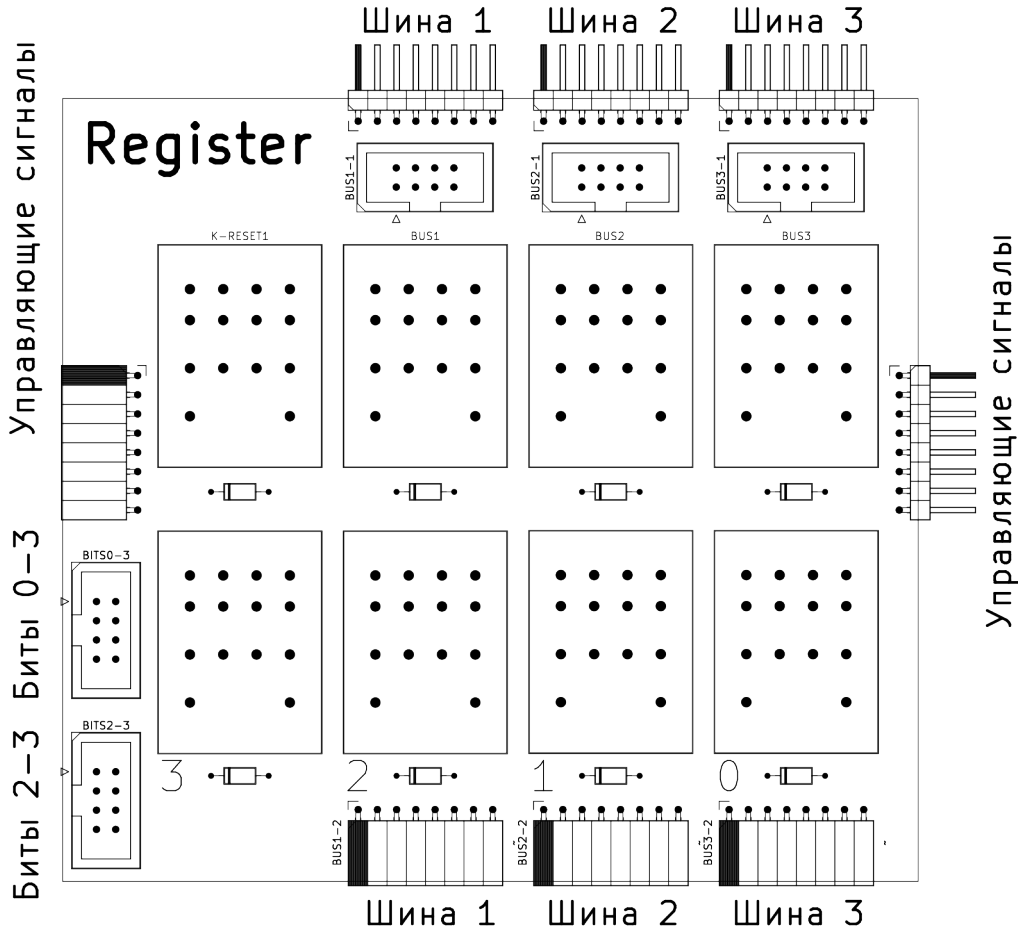
3.3 Шины в конструкторе

В конструкторе все компоненты соединяются с помощью восьмибитных шлейфов. В каждом из них есть питание и земля. Кроме того, большинство из таких подключений это шина в которой от 1 до 4 сигналов.

Например, 4 сигнала могут использоваться для управления модулем регистра. Или для передачи данных из одного регистра в другой. Каждым из таких сигналов можно управлять с помощью модуля переключателей, соединив его с соответствующей шиной.

3.4 Регистр

Регистры внутри процессоров (или периферийных устройств) используются для хранения данных и для проведения вычислений. Регистры могут иметь выделенное назначение (например, хранить режим работы устройства) или же используются почти для всего (адрес, операнд арифметических операций, значение для ввода-вывода).



Модуль четырёхбитного регистра состоит из четырёх реле-триггеров, одного реле для обнуления регистра и трёх реле для подключения к шинам данных.

Для хранения данных используются RS-триггеры, описанные выше, но сигнал для сброса у них общий. Таким образом, записывать данные можно только во все четыре бита сразу. Поэтому такой модуль и реализует функции цельного регистра, а не нескольких разрозненных RS-триггеров.

Также модуль регистра содержит три шинных формирователя. Их можно использовать для подключения триггеров к разным устройствам через три шины. Например, одна шина может быть предназначена для первого операнда при вычислениях, вторая для второго операнда, а третья для копирования данных между регистрами.

Если в модуль установить только реле шинных формирователей, получится модуль, который может подключать четырёхбитные сигналы (полученные из разъёма для прямого чтения значений триггеров) к одной из выбранных шин.

Модуль регистра имеет следующие разъёмы:

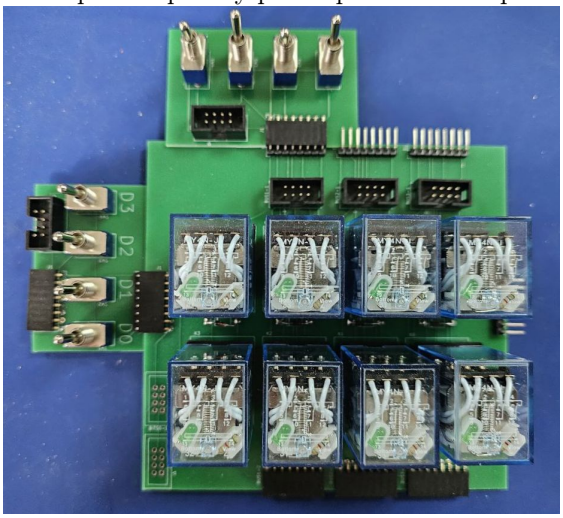
- Слева и справа: управляющие сигналы сброса и выборки. Можно подключить тумблеры для ручного включения сигналов. Также можно соединить несколько модулей регистра, чтобы управлять одним набором сигналов сразу для 8, 12 ... бит.
- Сверху и снизу: три шины данных. Реле регистра могут подключаться к шинам для записи или чтения данных.
- Дополнительные разъёмы с битами $0-3$ и $2-3$ для чтения или записи значения без подключения к шине.

Чтобы прочитать значение регистра нужно активировать сигнал выборки его на нужную шину. А уже через шину сигналы поступят на вход какой-то другой схемы.

Для записи значения в регистр нужно сначала обнулить его триггеры, иначе запись не произойдёт в тех битах, где уже хранятся единицы. После этого регистр подключается к нужной шине, и триггеры защёлкивают нужное значение.

3.4.1 Практикум

Протестировать работу регистра можно собрав следующую схему:



- Тумблеры слева управляют работой регистра. Бит 0 — обнуление, бит 1 — выборка на шину 1.

- Тумблеры сверху нужны для ввода значения регистра. Когда он подключается к шине 1, значения, набранное на тумблерах, записывается в регистр.

Регистр без шины

1. Подключить тумблеры проводом к битам $0 - 3$ вместо шины.
2. Набирать значение, убедиться, что биты переключаются в 1, но не возвращаются в 0.
3. Обнулить тумблеры с данными.
4. Включить и выключить сигнал сброса. Убедиться, что значения всех битов теперь 0.

Регистр с шиной

1. Отключить все управляющие сигналы.
2. Набрать значение на тумблерах для данных. Убедиться, что это не влияет на регистр.
3. Включить и выключить сигнал выборки на шину 1. Убедиться, что данные записались в регистр.
4. Включить и выключить сигнал сброса. Убедиться, что значения всех битов теперь 0.

3.4.2 Задачи

1. Придумайте, как скопировать данные из одного регистра в другой, не запоминая их в голове.

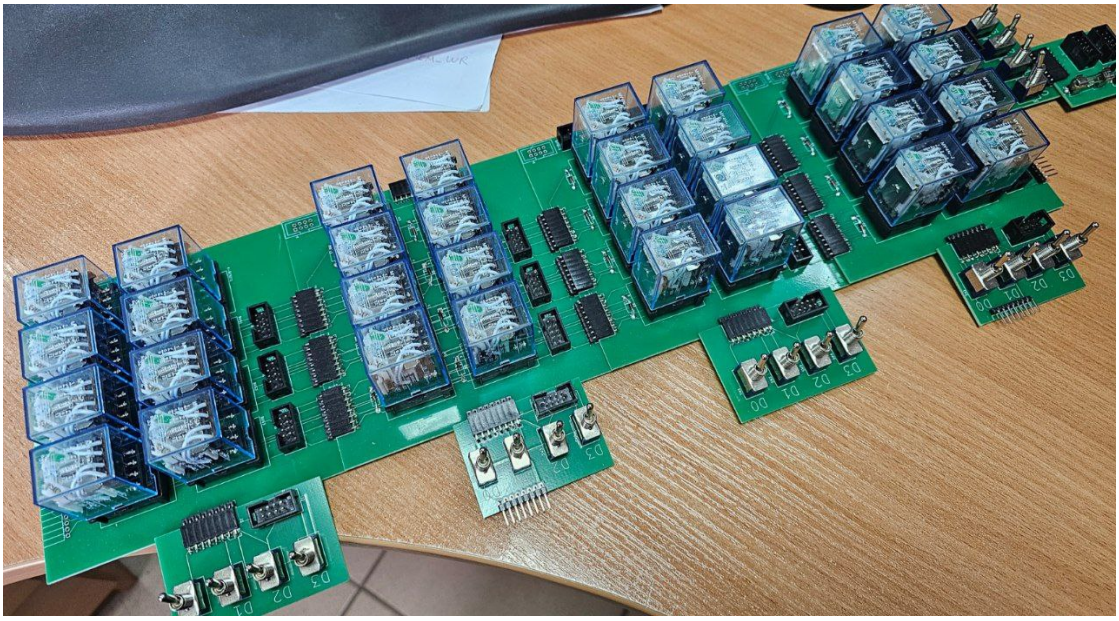
3.5 Шина и регистровый файл

Несколько регистров можно соединить в регистровый файл. Регистровый файл — это набор регистров. Он может быть однородным (все регистры эквивалентны) или нет (разные регистры имеют разное назначение).

У каждого из регистров есть сигналы выборки на одну из трёх шин. Если два регистра подключены к одной шине одновременно, то значения одного будут копироваться в другой. Если точнее, включённые биты включают аналогичные в другом регистре, то есть копирование возможно в обе стороны одновременно.

Нулевые биты при этом копироваться не могут. Для записи нулей регистр необходимо сбросить.

3.5.1 Практикум



Запись в регистры:

1. Отключить все управляющие сигналы.
2. Набрать значение на тумблерах, подключённых к шине данных 1.
3. Подключить с помощью тумблера регистр к шине 1. Убедиться, что в него записалось набранное значение.
4. Отключить регистр от шины.
5. Подключить другой регистр к шине 1. Убедиться, что в него записалось набранное значение.

Копирование значения:

1. Отключить все управляющие сигналы.
2. Подключить регистр с ненулевыми битами к шине 2.
3. Подключить пустой регистр к шине 2. убедиться, что он получил такое же значение, что и в первом регистре.
4. Отключить все управляющие сигналы.
5. Аналогично проверить шину 3.

Глава 4

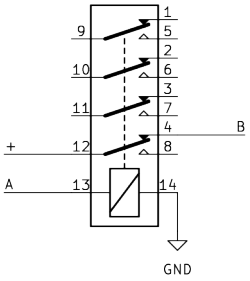
Вычисления

4.1 Логические операции

4.2 Отрицание

Одна из самых простых операций, которая используется в современных компьютерах — это операция отрицания. Она преобразует единицу в ноль, а ноль в единицу.

Такую логику можно реализовать с помощью нормально замкнутого контакта реле:



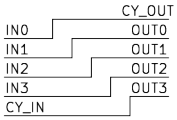
Когда реле выключено, ток может течь через контакт, поэтому на выходе B получается единица. Если же на входе A появляется единица, реле переключается и ток через нормально замкнутый контакт не течёт. На выходе B теперь ноль.

4.3 Сдвиг

Операция сдвига ещё более простая по сравнению с отрицанием. Суть её в том, что биты в числе перемещаются по какому-либо принципу (влево, вправо, по кругу).

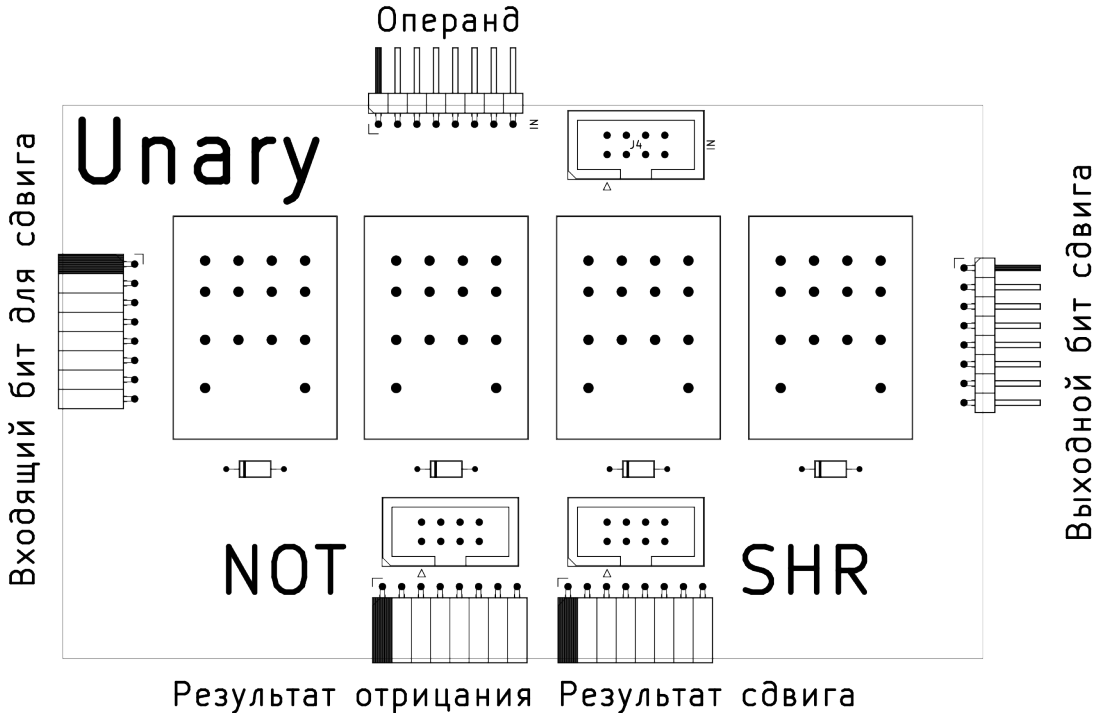
В конструкторе реализован сдвиг вправо. Нулевой бит вытесняется из числа (и может использоваться отдельно). Первый бит становится нулевым, второй первым и так далее. Такой сдвиг эквивалентен делению на два.

Для реализации сдвига вправо можно даже не использовать реле:



В конструкторе есть только сдвиг вправо. Ещё в вычислениях часто используется сдвиг влево, но его легко реализовать с помощью операции сложения числа с самим собой.

4.4 Модуль унарных логических операций

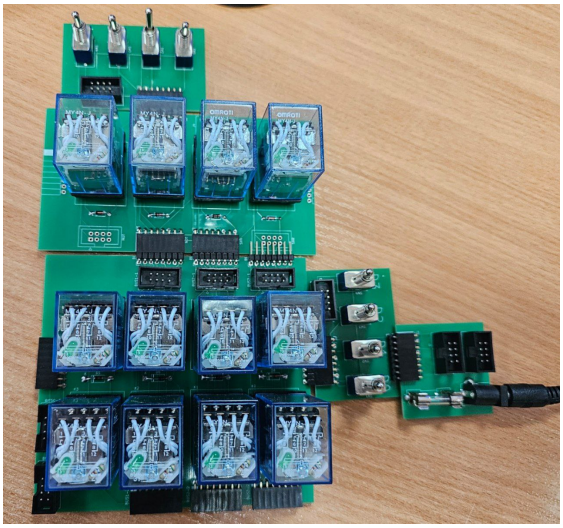


Модуль для унарных операций выполняет действия над 4-битным числом: сдвиг вправо и инверсия битов.

Может каскадироваться для сдвига 8-битных чисел.

4.4.1 Практикум

На вход модуля унарных операций подключается модуль с тумблерами. Выходы подключаются к шинам регистра.



1. Отключить все управляющие сигналы.

2. Набрать на тумблерах со входными данными значение 1100.

3. Подключить выходной регистр к шине 1. Убедиться, что в него записалось значение 0011 (инверсия).

4. Отключить регистр от шины, сбросить его значение.

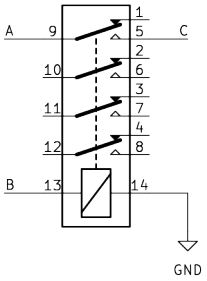
5. Подключить выходной регистр к шине 2. Убедиться, что в него записалось значение 0110 (сдвиг вправо).

4.4.2 Задачи

1. Собрать устройство, позволяющее сдвигать содержимое регистра и записывать результат обратно. Занести в регистр значение 1000 и сдвигать его до тех пор, пока не получится 0001. Вторую часть можно выполнять на скорость.

4.5 Логическое И

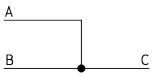
Реле представляет собой управляемый выключатель или переключатель. Если на один контакт A выключателя подать сигнал, то на другом C он появится только если на обмотке реле B есть напряжение (логическая единица).



Получается, что на выходе выключателя сигнал будет только тогда, когда реле включено и на входе выключателя тоже не ноль. Это означает, что такая схема реализует операцию логическое «И»: $C = A \wedge B$.

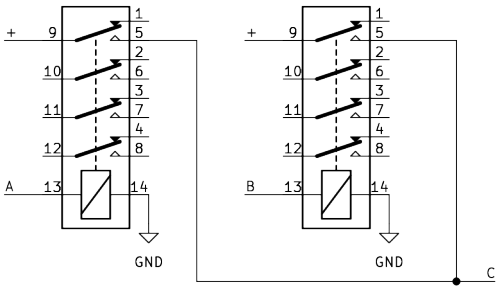
4.6 Логическое ИЛИ

Схему для выполнения логического «ИЛИ» можно получить, если соединить два выхода (две цепи). Тогда напряжение на объединённом выходе появится в любом из вариантов, если на первом выходе единица или на втором: $C = A \vee B$.



Но у этого способа есть один недостаток. Если на входе A окажется единица, то так как все входы и выходы соединены в одну цепь, поданное через A напряжение будет влиять и на вход B. И когда B используется ещё где-то, всё заработает неправильно, какое-то реле включится. Такой вот паразитный сигнал — A влияет на выходы, зависящие от B.

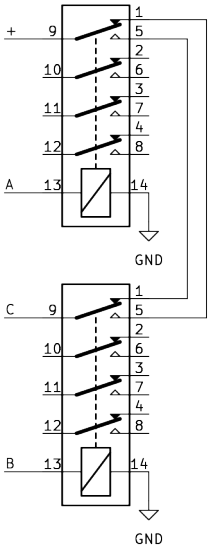
Поэтому иногда стоит использовать схему с реле для логического «ИЛИ»:



4.7 Исключающее ИЛИ

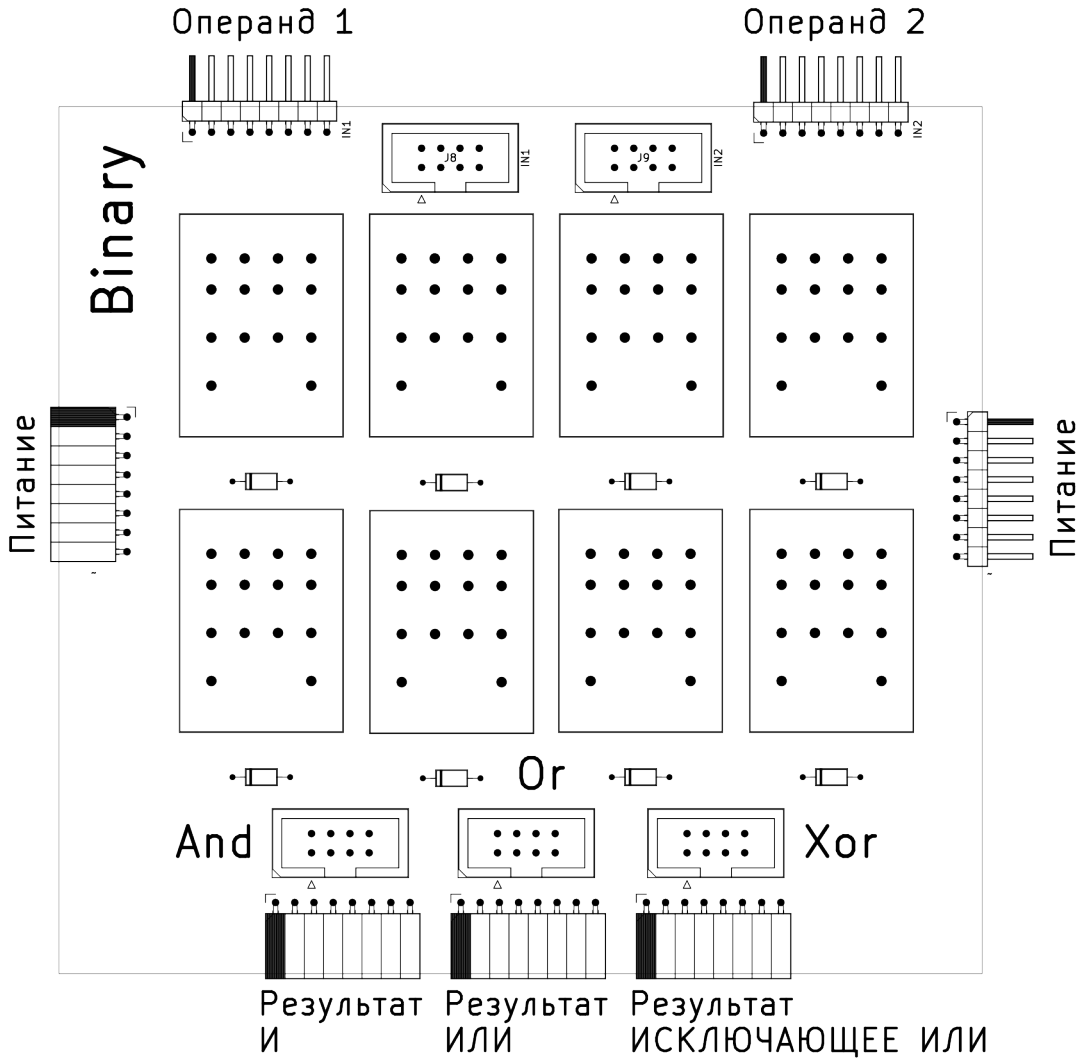
Схема для вычисления «исключающего или» несколько сложнее. Результат операции должен равняться единице только тогда, когда операнды не равны. То есть если на одном из входов уже было напряжение, а потом оно появляется и на другом входе, выход из состояния единицы должен переключиться в ноль.

Такую логику проще всего реализовать с помощью двух переключающих контактов:



Благодаря перекрёстному соединению переключателей проводниками, сигнал на выходе C появляется только тогда, когда переключатели на реле находятся в противоположных положениях.

4.8 Модуль бинарных логических операций



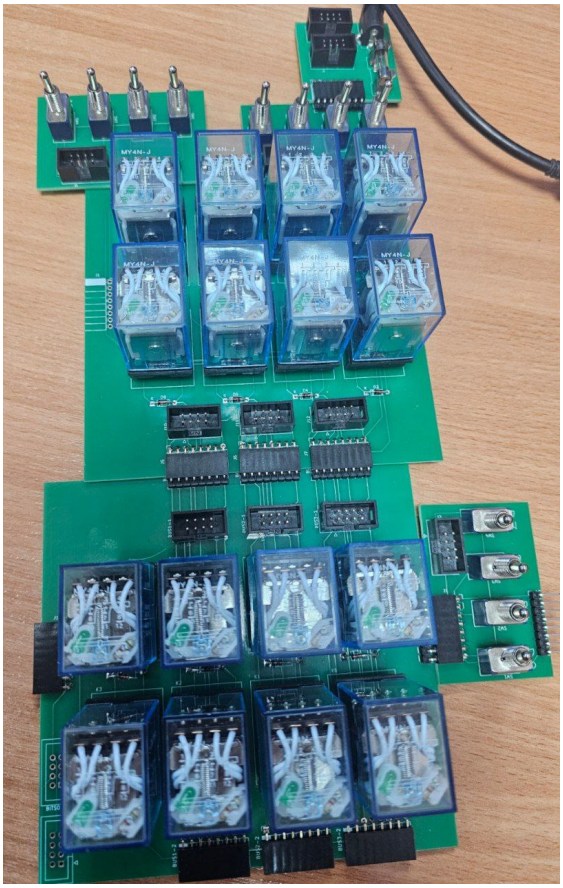
Модуль логических операций поддерживает вычисление AND, OR и XOR. У модуля есть два четырёхбитных входа и три выхода. Каждый выход отвечает за одну операцию.

4.8.1 Подготовка

1. Как могла бы выглядеть схема для выполнения операцию XOR?

4.8.2 Практикум

Ко входам модуля подключаются два модуля с тумблерами, а к выходам — регистр, куда будет записываться результат.



1. Отключить все управляющие сигналы.

2. Набрать на тумблерах первого операнда значение 1100.

3. Набрать на тумблерах второго операнда значение 1010.

4. Подключить выходной регистр к шине 1. Убедиться, что в него записалось значение 1000 (AND).

5. Отключить регистр от шины, сбросить его значение.

6. Подключить выходной регистр к шине 2. Убедиться, что в него записалось значение 1110 (OR).

7. Отключить регистр от шины, сбросить его значение.

8. Подключить выходной регистр к шине 3. Убедиться, что в него записалось значение 0110 (XOR).

4.9 Арифметические операции

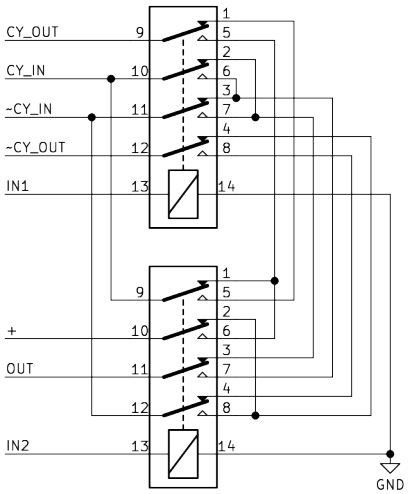
4.10 Сложение

Чтобы складывать числа, сначала нужно научиться складывать отдельные биты. Сумма двух битов может дать результат 0_2, 1_2 или 10_2. То есть при сложении получается уже двухбитовое число.

Схему сложения удобнее всего строить из однотипных компонентов, складывающих фиксированное количество битов. На выходе такого компонента будет результат сложения, а также бит переполнения (переноса). Для каскадирования таких модулей нужно также иметь возможность подавать на вход перенос от сумматоров младших битов.

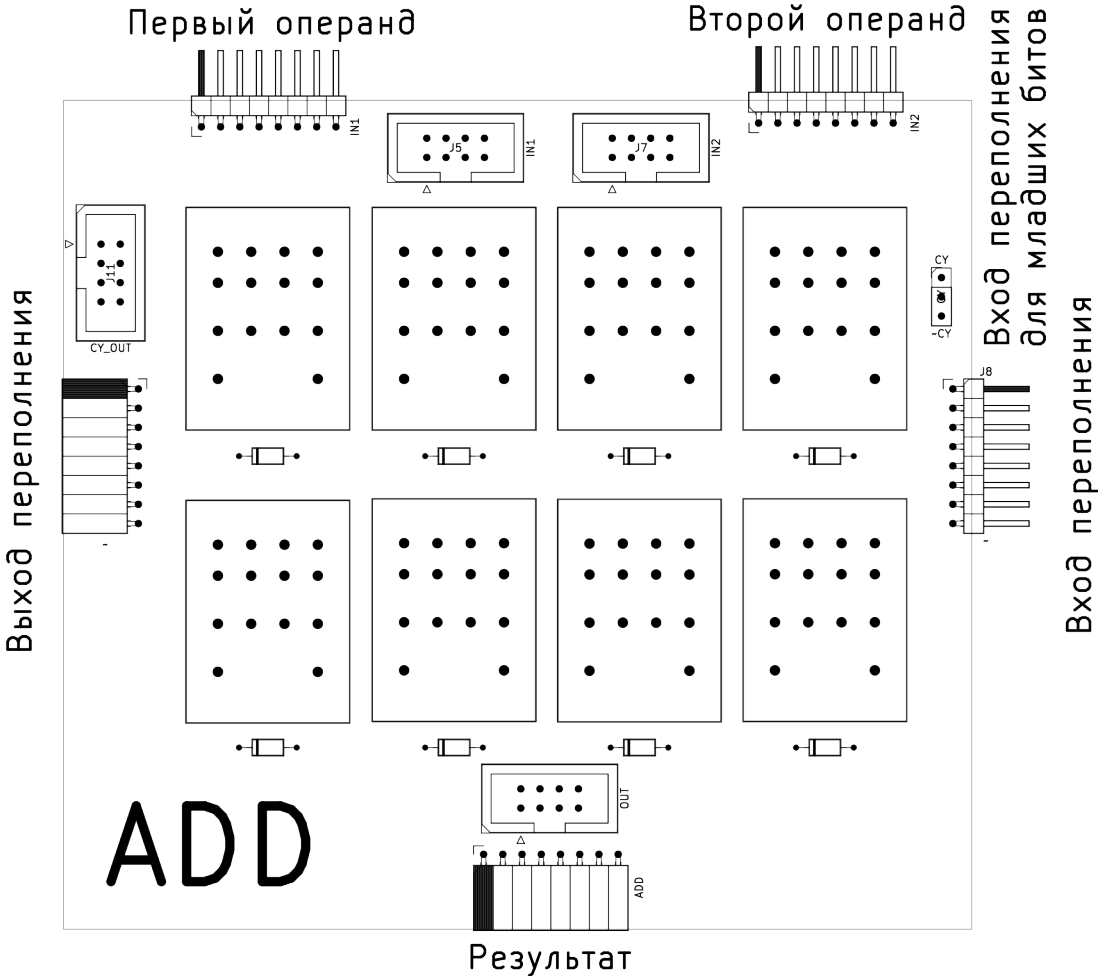
Таким образом, сумматор получает на вход бит переноса и два бита-слагаемых, а на выходе у него тоже бит переноса и однобитная сумма.

Особенность реализации сумматора с помощью реле в том, что на входе требуется не только перенос, но и инвертированный перенос. Такой же инвертированный перенос можно генерировать и на выходе.



Эту схему сумматора изобрёл Конрад Цузе для своих компьютеров в 1940х годах. Несколько таких сумматоров можно соединить последовательно, чтобы складывать многобитные числа. Для этого соединяются соответствующие входы и выходы для переноса и инвертированного переноса, чтобы переполнение переходило из бита 0 в бит 1, из бита 1 в бит 2 и так далее.

4.11 Модуль сумматора



Сумматор складывает два четырёхбитных числа, один бит переноса и выдаёт четырёхбитное число и бит переноса. У модуля есть два четырёхбитный вход и один выход. Каждый выход отвечает за одну операцию.

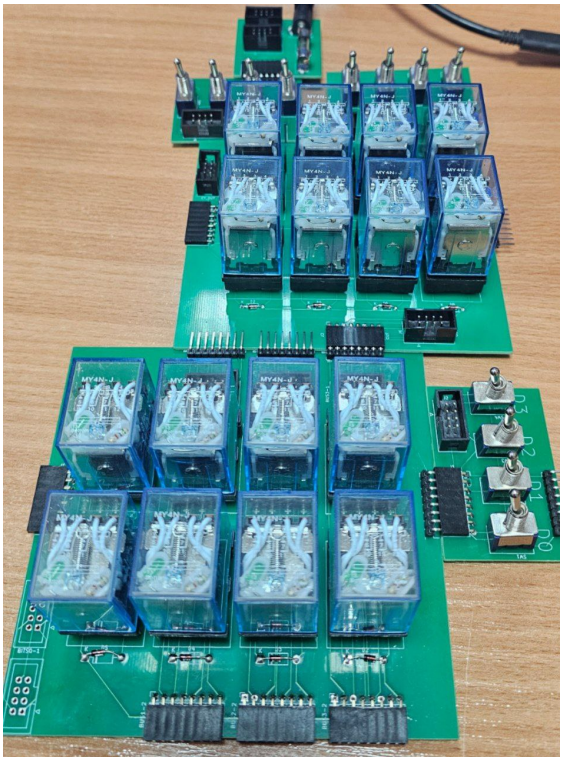
4.11.1 Подготовка

1. Придумайте, как можно было бы составить схему из реле, чтобы вычислять сумму однобитовых чисел.

2. Как можно использовать предыдущую схему для вычисления суммы двухбитовых чисел?

4.11.2 Практикум

Ко входам модуля подключаются два модуля с тумблерами, а к выходам — регистр, куда будет записываться результат.



1. Отключить все управляющие сигналы.
2. Установить перемычку для подачи сигнала на CY.
3. Набрать на тумблерах первого операнда значение 0011 (число 3).
4. Набрать на тумблерах второго операнда значение 1010 (число 10).
5. Подключить выходной регистр к шине 3. Убедиться, что в него записалось значение 1101 (число 13).
6. Отключить регистр от шины, сбросить его значение.
7. Установить перемычку для подачи сигнала на CY.
8. Подключить выходной регистр к шине 3. Убедиться, что в него записалось значение 1110 (число 14).

4.11.3 Задачи

1. Собрать устройство для сложения значения из регистра с константой, набранной на тумблерах. Должна быть предусмотрена запись результата обратно в регистр. Дальше управлять сигналами таким образом, чтобы к регистру последовательно прибавлялась единица, пока он не достигнет значения 1111.

4.12 Вычитание

Вычитание в компьютерах делается с помощью трюков с переполнением. Что такое $10-7=3$ в нашем четырёхбитном вычислителе? Это $1010-0111=0011$. Так как разрядов всего 4, тот же результат можно получить с помощью сложения. Сначала к 10 прибавим 5, получив 15 ($1010+0101=1111$). Потом прибавим ещё единицу. Должно было бы получиться 16 (10000), но так как значащих битов всего 4, пятый бит результата пропадёт и останется значение 0 (0000).

Остаётся увеличить это число на единицу ещё три раза, чтобы получился правильный ответ. Итого, мы прибавили к 10 число 9 вместо вычитания 5 и получили нужный результат.

Следовательно, вместо вычитания 7 (0111) можно прибавлять 9 (1001), когда речь идёт о четырёхбитных вычислениях.

Такой же фокус можно проделать с любым числом. Быстрый алгоритм работает так: инвертируем все биты числа, а затем прибавляем единицу. Это называется дополнительным обратным кодом.

Например, для 0111 сначала получится 1000, а потом 1001, что и требовалось. Работает и в обратную сторону — из 1001 инверсией получаем 0110, а после добавления единицы будет 0111.

Таким образом, для вычитания не нужно строить сложную схему, а достаточно использовать дополнительный обратный код для второго операнда при сложении.

4.13 Вычитание через сложение с дополнительным обратным кодом

Чтобы вычесть из одного числа другое, можно вычитаемое представить в дополнительном обратном коде, а затем сложить это число с уменьшаемым.

Число в дополнительном обратном коде получается, если сначала инвертировать биты исходного числа, а затем прибавить к нему единицу.

Например, для числа $3=0011$ инверсия будет выглядеть, как 1100, а дополнительный обратный код, как 1101.

Практикум

Собрать схему для сложения.

1. Отключить все управляющие сигналы.
2. Установить перемычку для подачи сигнала на CY.
3. Набрать на тумблерах первого операнда значение 1010 (число 10).
4. Набрать на тумблерах второго операнда значение 1101 (число -3).
5. Подключить выходной регистр к шине 3. Убедиться, что в него записалось значение 0111 (число 7).

Задачи

1. Выполните деление в столбик числа 14 на 3 с помощью последовательности вычитаний. Записывайте промежуточные результаты на бумаге.

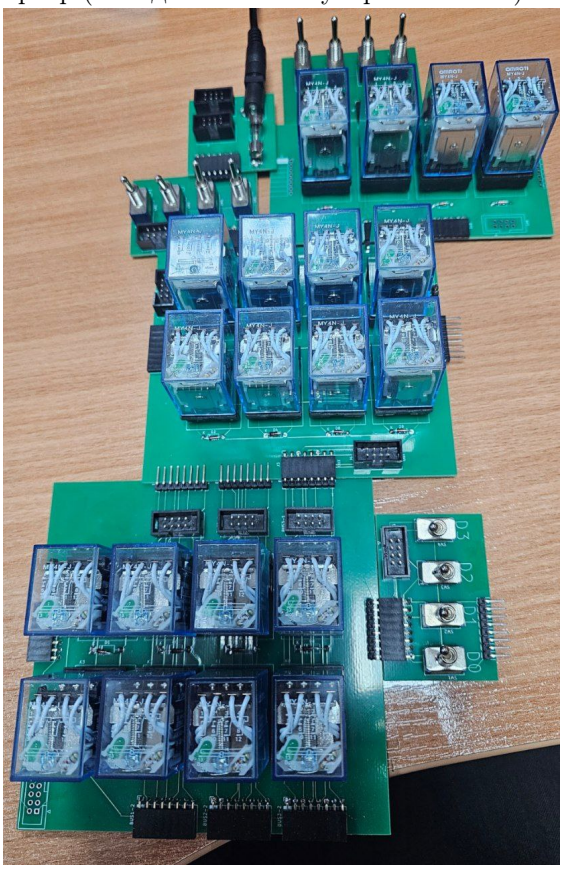
2. Сделайте то же самое, только сохраняйте все промежуточные результаты в разных регистрах регистрового файла.

4.14 Вычитание с помощью сумматора и инвертора

Вычитание можно делать с помощью сумматора. Чтобы на входе из операнда получался дополнительный обратный код, сначала нужно преобразовать его с помощью инвертора, а затем прибавить 1, включив вход переноса в сумматоре.

Практикум

Собрать схему для сложения, а затем добавить между тумблерами со вторым операндом инвертор (выход NOT блока унарной логики):



1. Отключить все управляющие сигналы.
2. Установить перемычку для подачи сигнала на CY.
3. Набрать на тумблерах первого операнда значение 1010 (число 10).
4. Набрать на тумблерах второго операнда значение 0011 (число 3).
5. Подключить выходной регистр к шине 3. Убедиться, что в него записалось значение 0111 (число 7).

Глава 5

Калькулятор

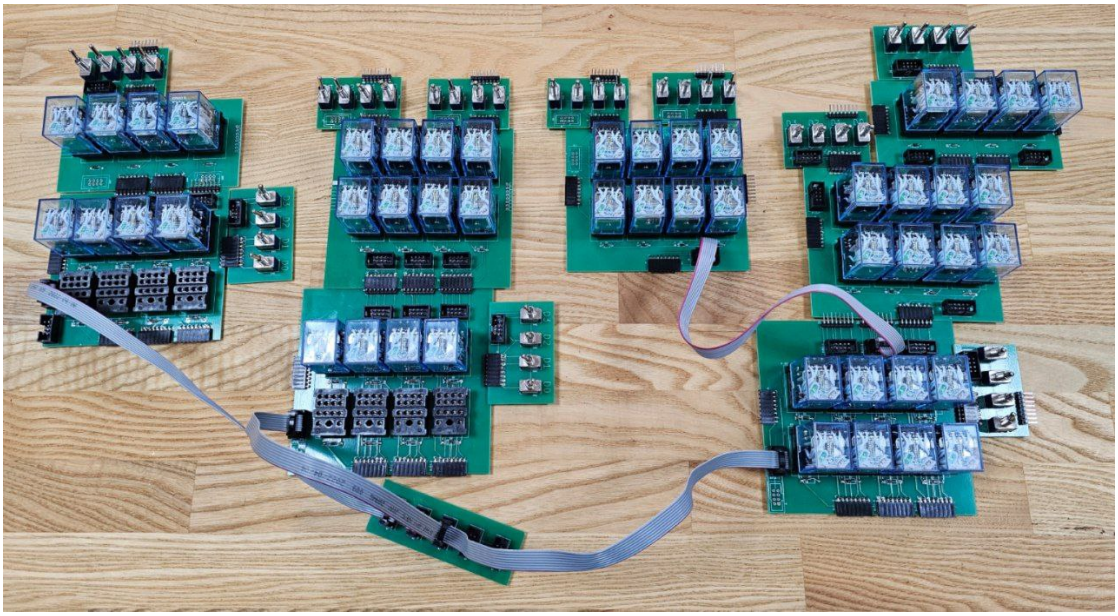
5.1 Калькулятор для 7 операций

Все вычислители можно соединить с одним и тем же регистром, как хранилищем результата, чтобы получить простейший калькулятор. Доступны следующие операции:

1. Инверсия
2. Сдвиг вправо
3. Побитовое И
4. Побитовое ИЛИ
5. Исключающее ИЛИ
6. Сложение
7. Вычитание

Практикум

Уже проверенные модули для разных операций соединяются вместе. Для этого используются несколько мультиплексоров (шинных формирователей) на платах регистров. На этих платах не установлены реле для хранения битов. Вместо этого шины подключаются к одному и тому же регистру. Так в него можно записывать любой из 7 результатов вычислений.



Любую из операций можно выполнить так:

1. Отключить все управляющие сигналы.
2. Набрать входные данные для нужной операции.
3. Подключить выход нужного модуля к регистру тумблером.
4. Наблюдать результаты вычислений.
5. Сбросить значение регистра.

5.2 Расширение вычислений до восьми бит

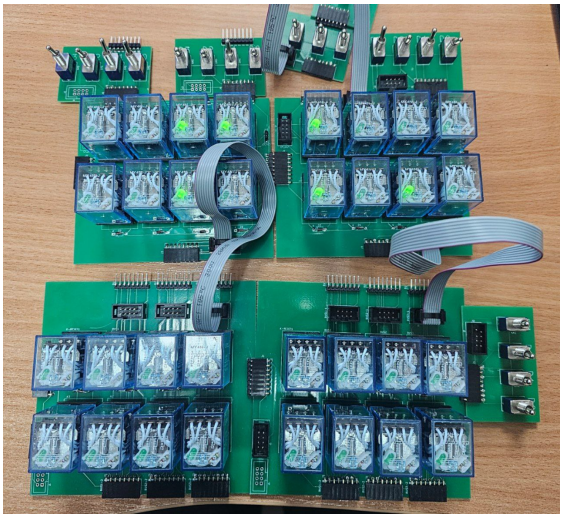
У регистров и вычислительных модулей справа и слева есть разъёмы для расширения разрядности.

Для регистров через эти разъёмы передаются сигналы сброса и подключения к шинам. Поэтому один набор тумблеров может использоваться для двух четырёхбитных плат-регистров, если они соединены в один восьмибитный регистр.

Для вычислительных модулей через боковые разъёмы передаются сигналы переноса (в случае сдвига и сложения).

Практикум

Соединить два регистра и два сумматора. Шина 3 регистров подключается к сумматору. У правого (младшего) сумматора входящий перенос перемычкой устанавливается в 0. У левого (старшего) сумматора перемычка для переноса убирается, потому что сигнал переноса приходит от младших битов (из правого сумматора).



1. Отключить все управляющие сигналы.
2. Набрать входные данные 00111001 и 00101001.
3. Подключить выход сумматора к регистру тумблером $D3$.
4. Наблюдать результат вычислений 01100010.

Глава 6

Элементы компьютера

6.1 Дешифратор

6.2 ПЗУ

6.3 Декодер инструкций

6.4 Тактовый генератор

Глава 7

Компьютер