

Министерство образования Республики Беларусь

**Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»**

Кафедра «Автоматизированный электропривод»

**МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА
В АВТОМАТИЗИРОВАННОМ
ЭЛЕКТРОПРИВОДЕ**

ПРАКТИКУМ

**по одноименной дисциплине
для студентов специальности 1-53 01 05
«Автоматизированные электроприводы»
дневной формы обучения**

Гомель 2019

УДК 62-83-52:004.315(075.8)
ББК 31.291+32.965.7я73
М59

*Рекомендовано научно-методическим советом
факультета автоматизированных и информационных систем
ГГТУ им. П. О. Сухого
(протокол № 11 от 04.06.2018 г.)*

Составитель *В. А. Савельев*

Рецензент: доц. каф. «Теоретические основы электротехники» ГГТУ им. П. О. Сухого
канд. техн. наук, доц. *А. В. Козлов*

Микропроцессорные средства в автоматизированном электроприводе : практикум
М59 по одной дисциплине для студентов специальности 1-53 01 05 «Автоматизированные электроприводы» днев. формы обучения / сост. В. А. Савельев. – Гомель : ГГТУ им. П. О. Сухого, 2019. – 42 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: <https://elib.gstu.by>. – Загл. с титул. экрана.

Содержит примеры решения типовых задач программирования микроконтроллеров и ПЛК по дисциплине «Микропроцессорные средства в автоматизированном электроприводе», а также задания для самостоятельной работы.

Для студентов специальности 1-53 01 05 «Автоматизированные электроприводы» дневной формы обучения.

**УДК 62-83-52:004.315(075.8)
ББК 31.291+32.965.7я73**

© Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», 2019

Тема 1. Управление светодиодами

Постановка задачи

После замыкания контактов выключателя (рис.1.1) поочередно парами загораются 16 светодиодов. После того как все диоды загорятся, они должны одновременно погаснуть. При повторном замыкании контактов выключателя все операции повторяются. Если контакты выключателя остаются в разомкнутом состоянии, светодиоды не горят.

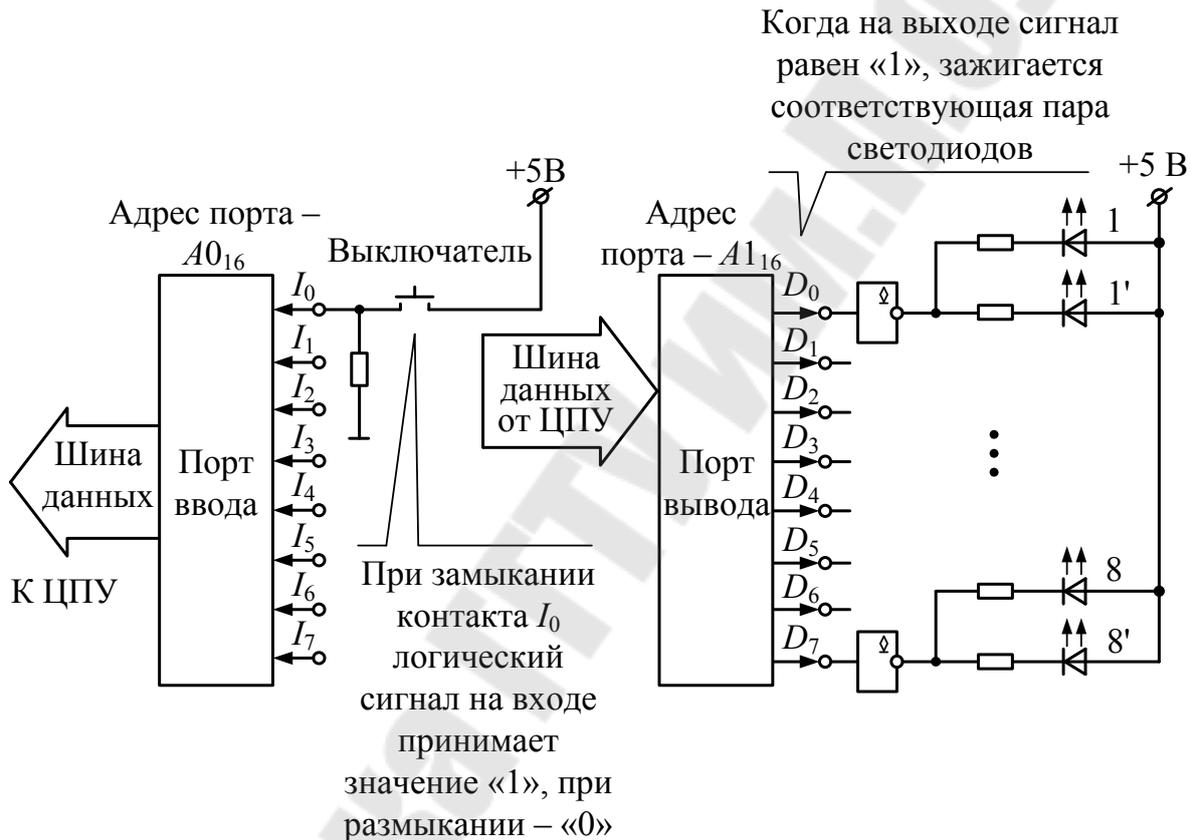


Рис. 1.1. Подключение выключателя и светодиодов к портам ввода-вывода

Решение задачи

В данном примере входными и выходными устройствами микроконтроллера, выполняющего функцию контроллера, являются выключатель и светодиоды.

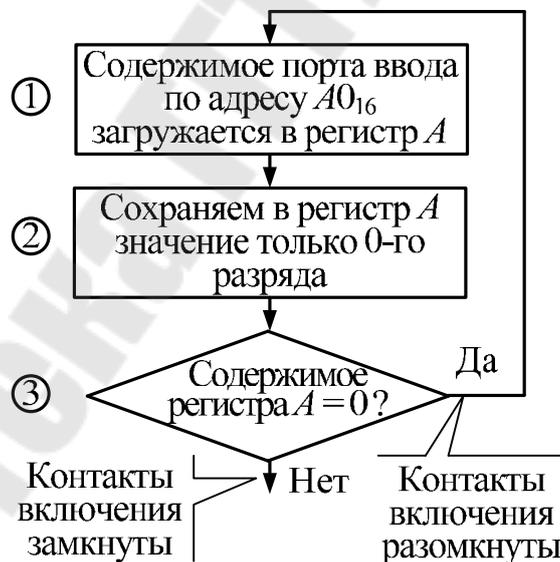
Как видно из рис.1.1 выключатель соединен с выводом I_0 (0-м разрядом) порта ввода. Светодиоды подключены парами к восьми выводам порта вывода.

Для того чтобы надежно зафиксировать низкий уровень логического сигнала на входе I_0 порта ввода, соединенного с выключателем, между соответствующим выводом микросхемы и общим выводом подключен резистор, обеспечивающий смещение потенциала на этом входе при разомкнутом выключателе до уровня логического «0». Соответственно, если выключатель перевести в замкнутое состояние, напряжение на входе I_0 поднимется до уровня логической «1».

К каждому из 8 выходов ($D_0 \dots D_7$) 8-разрядного порта вывода посредством логических элементов «НЕ» присоединена группа из двух светодиодов. При таком способе подключения пара светодиодов загорается, когда сигнал на соответствующем выводе (D_n) равен «1».

Программирование «по частям»

Учитывая конкретное содержание данной задачи, можно разбить ее на две самостоятельные процедуры — процедуру определения состояния выключателя и процедуру поэтапного включения светодиодов. В соответствии с этим целесообразно составить программу из двух частей, а затем объединить их в одно целое. На рис. 1.2 представлены схема алгоритма и соответствующая ей программа определения состояния выключателя.



Программа в мнемокоде:

```

LOOP1: IN, A0      ①
        ANI, 01    ②
        JZ, LOOP1  ③
  
```

Рис. 1.2. Программа определения состояния переключателя

Посредством команды ввода (*IN*, *A0*) содержимое из порта, имеющего адрес *A0*, загружается в регистр *A*. Затем выполняется логическая операция «И» (*AND*, *01*), в результате которой все разряды регистра *A*, кроме 0-го, обнуляются. Если после выполнения этой операции содержимое регистра *A* не равно 0, это означает, что контакты выключателя замкнуты. Если содержимое регистра *A* равно 0, контакты выключателя разомкнуты. При размыкании контактов содержимое регистра *A* становится равным 0. Если контакты разомкнуты, переходим к команде ввода и повторяем описанную операцию определения состояния контактов. Эта последовательность операций повторяется до тех пор, пока контакты переключателя не замкнутся.

На рис. 1.3 проиллюстрирован принцип организации последовательного включения светодиодов и приведена соответствующая программа.

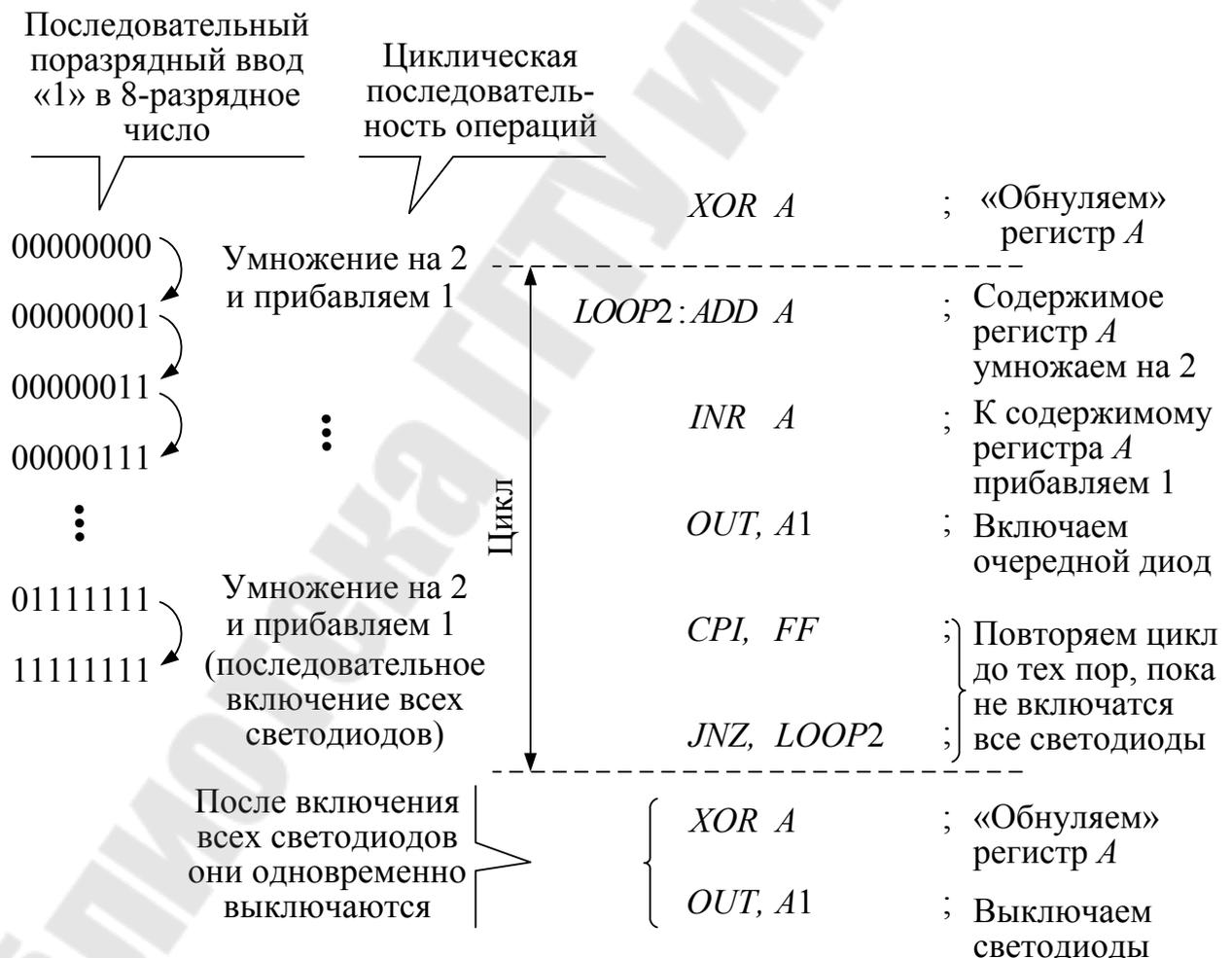


Рис. 1.3. Программа процедуры последовательного включения светодиодов

Задача сводится к последовательной замене нулей 8-разрядного числа 00000000_2 единицами начиная с 0-го разряда, и пересылки полученных чисел ($00000000_2 \dots 11111111_2$) в порт вывода.

Поэтапное заполнение разрядов исходного числа «единицами» осуществляется путем удвоения предыдущего результата и прибавления к нему 1. Программа дополнена операцией одновременного выключения светодиодов, после того как все они будут включены.

Когда мы говорим, что светодиоды зажигаются «последовательно», учитываем ли мы при этом, с каким интервалом они должны зажигаться. Если управлять светодиодами по программе, представленной на рис. 1.3, то мы не увидим последовательного процесса включения светодиодов. Они загораются слишком быстро для того, чтобы человек мог заметить последовательность включения.

Чтобы светодиоды включались с интервалом 0,1 или 1 с необходимо в интервалах между операциями включения «отвлечь» ЦПУ на выполнение других операций, чтобы «убить» время (подпрограмма задержки)

Эту проблему решает подпрограмма задержки, представленная на рис. 1.4.

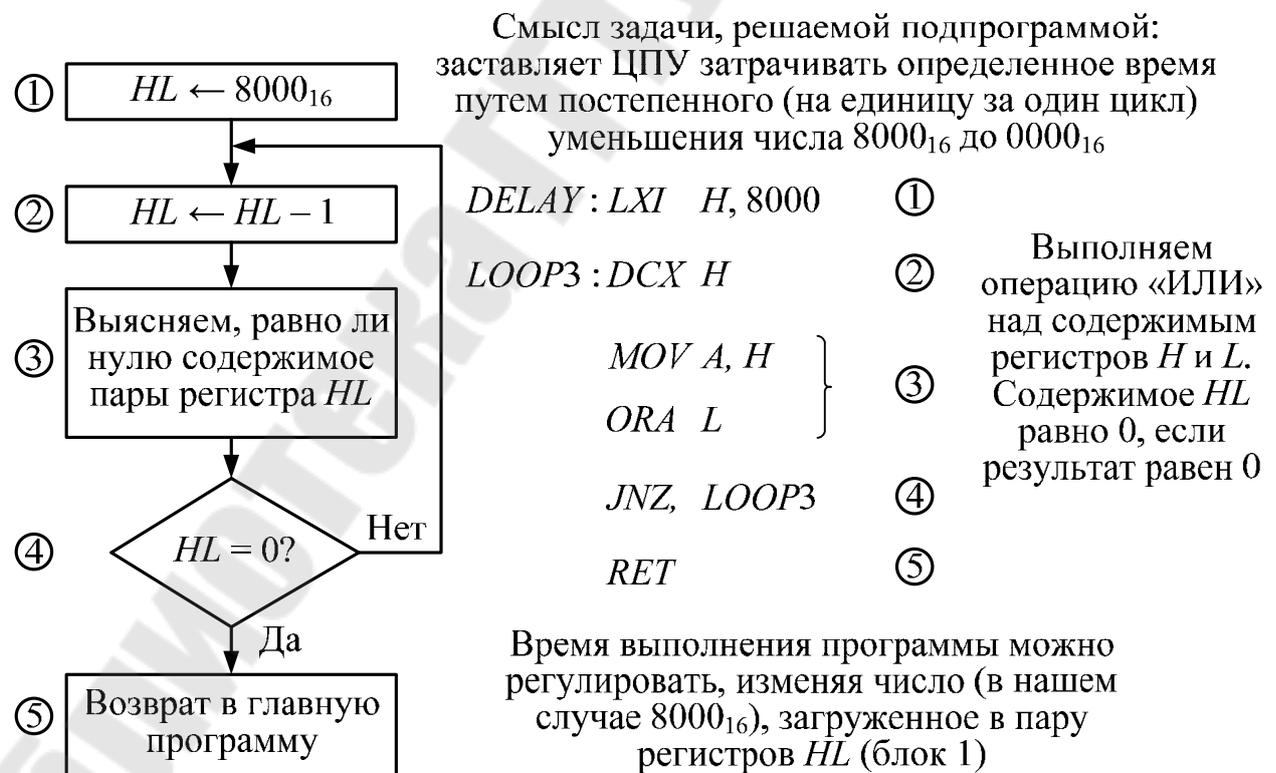


Рис. 1.4. Подпрограмма задержки

Поскольку генератор тактовой частоты работает непрерывно, ЦПУ не может остановиться. Следовательно, чтобы организовать задержку, ЦПУ нужно дать какую-нибудь задачу, на выполнение которой потребуется достаточно продолжительное время. В качестве такой задачи подпрограмма предлагает ЦПУ «сосчитать», например, до 32768_{10} . Вообще время, требуемое для подсчета, определяется тем, какое число загружено в двоянный регистр *HL* до начала подсчета.

Рассмотрим структуру подпрограммы задержки. Сначала подсчитаем время выполнения подпрограммы. Число тактовых импульсов, требуемых для выполнения одного цикла подпрограммы, равно сумме импульсов, необходимых для выполнения каждой из команд цикла (от метки *LOOP3* до команды условного перехода *JNZ*). Это число равно: $6 + 4 + 4 + 10 = 24$. Если в регистровую пару *HL* загружено число 8000_{16} , то указанный цикл будет повторяться 8000_{16} раз (в десятичной системе счисления — 32768 раз). Так как длительность периода тактовой частоты (2,5 МГц) равна 0,4 мкс, время выполнения подпрограммы примерно равно 0,3 с. Точный расчет производится следующим образом:

Время выполнения подпрограммы = число импульсов × число повторений × период = $24 \times 32768 \times 0,4 \times 10^{-6} \text{ с} = 0,3 \text{ с}$.

Если в двоянный регистр загружено число 5000_{16} , время выполнения подпрограммы равно 0,2 с.

Вызов подпрограммы задержки из главной программы производится с помощью оператора «*CALL DELAY*», где *DELAY* — это метка языка ассемблера, представляющая собой адрес первой команды этой подпрограммы, заданный в виде символической константы. При трансляции операторов языка ассемблера в коды машинных команд устанавливается числовое значение адреса, соответствующее данной символической константе.

Теперь объединим приведенные выше программы в единое целое. При этом следует обратить особое внимание на то, как изменяется содержимое регистров, используемых в нескольких программах. Например, во всех программах используется регистр *A*. Если при объединении программ содержимое регистра *A* в каждом конкретном случае будет отличаться от того, каким оно было при отдельном выполнении данной части программы, то программа будет работать неправильно. Поэтому, перед тем как содержимое регистра *A* будет изменено другой программой, необходимо «спрятать» текущее значение

в другом регистре. Объединенная программа, составленная с учетом этого обстоятельства, приведена на рис. 1.5.

3 процедуры – 1, 2 и 3 объединяем в одну программу

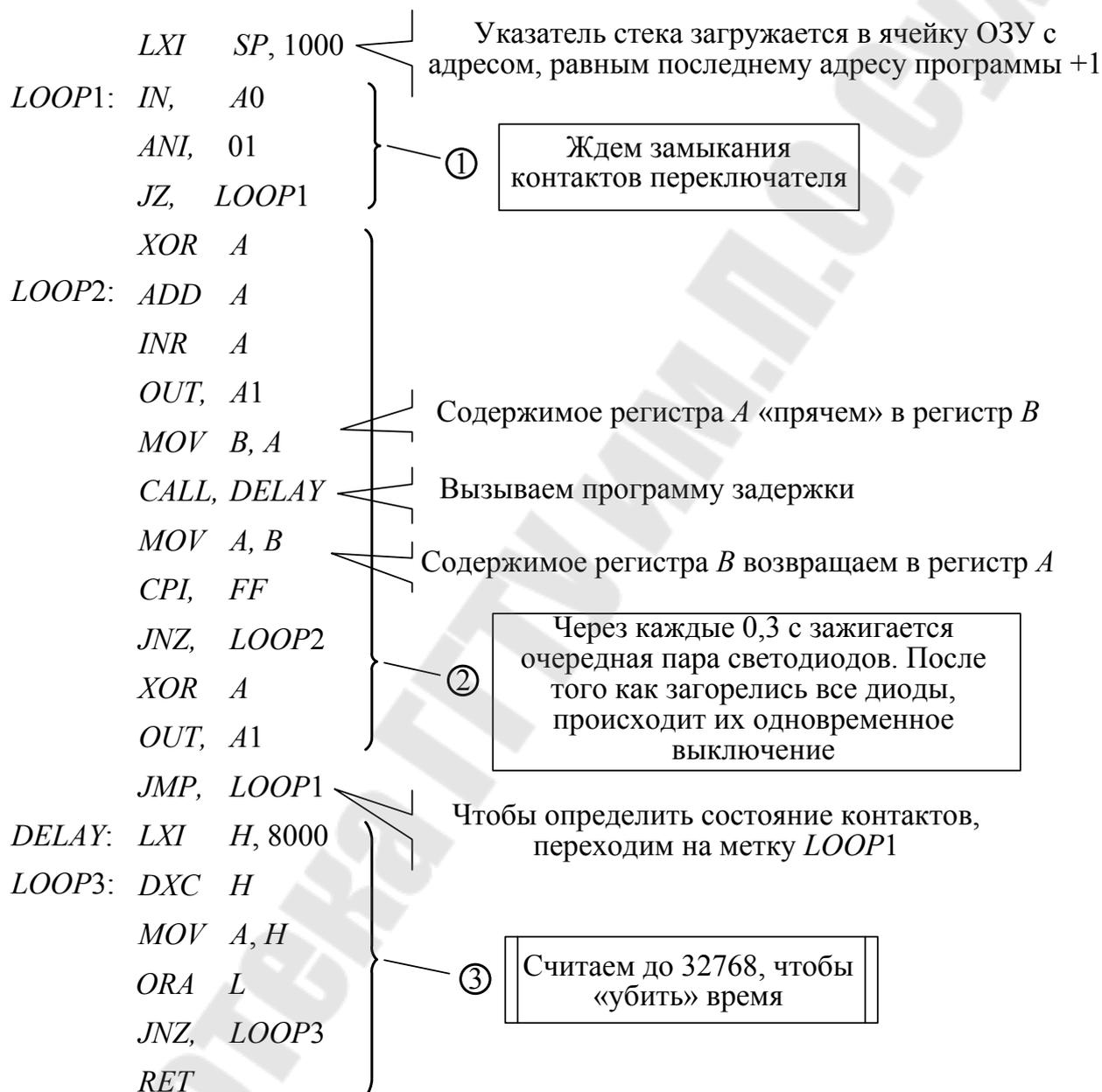


Рис. 1.5. Объединенная программа выполнения задачи № 1

В процедуру последовательного включения диодов введен оператор *CALL*, вызывающий подпрограмму задержки. При возвращении из подпрограммы задержки содержимое регистра *A* всегда равно 0. Поэтому на время выполнения подпрограммы содержимое регистра *A*

записывается в регистр B . Запись содержимого регистра A в регистр B перед командой вызова подпрограммы, а также считывание оттуда сохраненного значения этого числа обратно в регистр A после выполнения подпрограммы осуществляется посредством оператора загрузки MOV .

Первая команда программы « $LXI SP, 1000_{16}$ » устанавливает начальное значение указателя стека 1000_{16} , тем самым указывая область стека. Хотя это не имеет явного отношения к выполняемой задаче, указывать область стека в начале программы необходимо.

Задание для самостоятельной работы

После замыкания контактов выключателя (см. рис. 1.1) сначала загораются первые 8 светодиодов, а затем, с интервалом в 1 секунду – вторые 8 светодиодов. После того как все диоды загорятся, они должны несколько раз мигнуть с интервалом в 1 секунду и погаснуть. При повторном замыкании контактов выключателя все операции повторяются. Если контакты выключателя остаются в разомкнутом состоянии, светодиоды не горят.

Тема 2. Запоминание последовательности сигналов и воспроизвести ее с соблюдением интервалов между сигналами

Постановка задачи

Сигналы подаются с помощью 6 выключателей (рис.2.1), подключенных к порту ввода. Последовательность подачи сигналов с контактов выключателей из порта ввода заносится в память. Через порт вывода контакты шести реле в той же последовательности воспроизводят сигналы контактов выключателей. При воспроизведении сигналов осуществляется задержка (приблизительно 0,5 с) между командами замыкания реле. Начало и окончание записи сигналов задаются посредством отдельных выключателей.

Решение задачи

На схеме (см. рис. 2.1) показано 6 выключателей $PS_1 \dots PS_6$ подключенных соответственно ко входам $I_1 \dots I_6$ и задающих последовательность сигналов, которую предстоит запомнить микроконтроллеру. Момент начала записи последовательности сигналов определяется состоянием контактов выключателя SW_0 , соединенного с выводом I_0 ,

порта ввода. Окончание записи сигналов и начало их воспроизведения микроконтроллер будет определять по состоянию контактов выключателя SW_7 в цепи входа I_7 порта ввода. При замыкании контактов этого переключателя прекращается запись сигналов, а при размыкании начинается их воспроизведение.

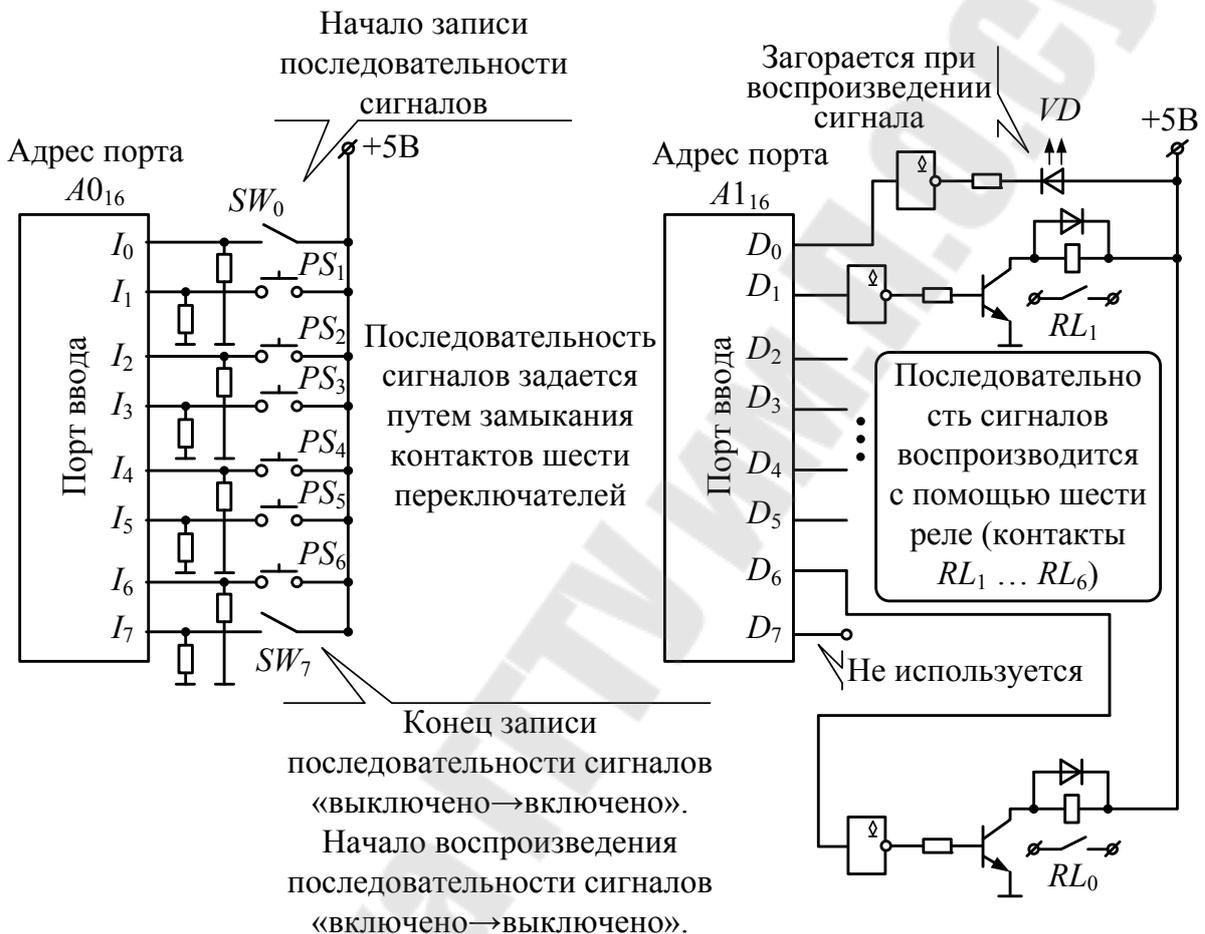


Рис. 2.1. Схема подключения внешних устройств к портам ввода-вывода

Воспроизведение записанной последовательности сигналов будет осуществляться путем замыкания контактов шести реле, подключенных к выходам $D_1 \dots D_6$ порта вывода. Кроме того, к выходу D_0 порта вывода подключен светодиод VD , который должен гореть во время воспроизведения записанной микроконтроллером последовательности сигналов.

Всю процедуру можно разбить на две самостоятельные задачи: «запомнить последовательность подачи сигналов» и «воспроизвести записанную последовательность». Как и в предыдущем случае, начи-

нать надо с составления программ для каждой из двух частей, чтобы потом объединить их в единую программу.

Процедура записи в память последовательности подачи сигналов

Как всегда при решении подобных задач, построим схему алгоритма процедуры «запомнить последовательность подачи сигналов» (рис. 2.2).

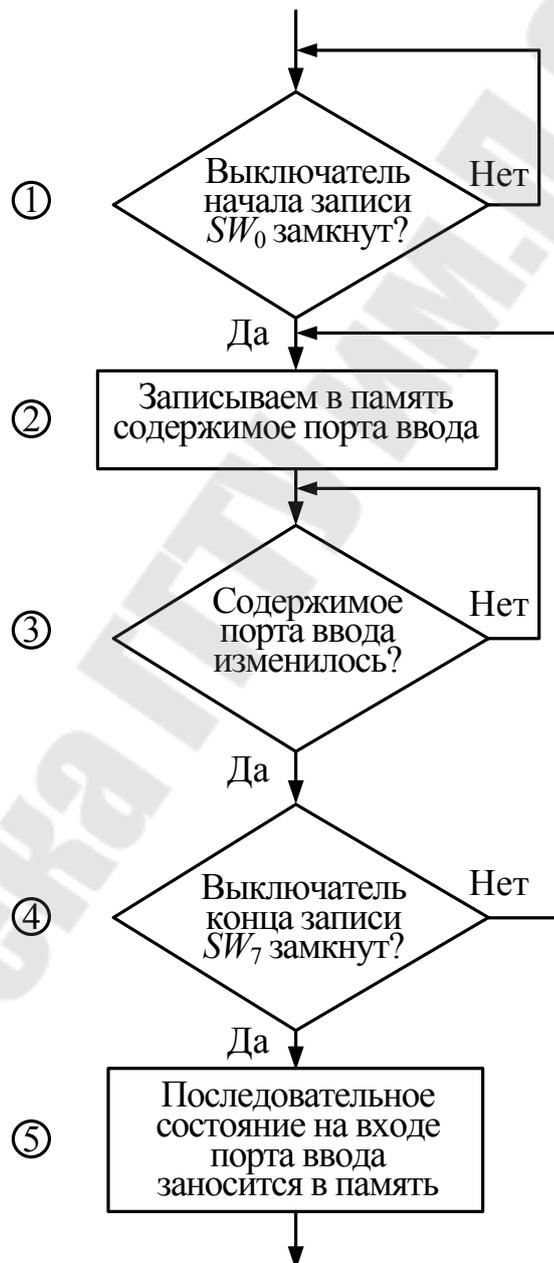


Рис. 2.2. Алгоритм процедуры «запомнить последовательность подачи сигналов»

Первый блок изображенной на рис. 2.2 схемы алгоритма точно такой же, как и в предыдущем примере. Это проверка состояния контактов выключателя SW_0 . Если контакты замкнуты, переходим ко второму блоку — загрузке в память содержимого порта ввода. Загрузка осуществляется следующим образом. Сначала с помощью команды ввода содержимое порта ввода считывается в регистр A , а затем посредством команды загрузки содержимое регистра A записывается по указанному адресу. На рис. 2.3 показано, как определяется адрес ячейки памяти для записи очередного числа из регистра A .

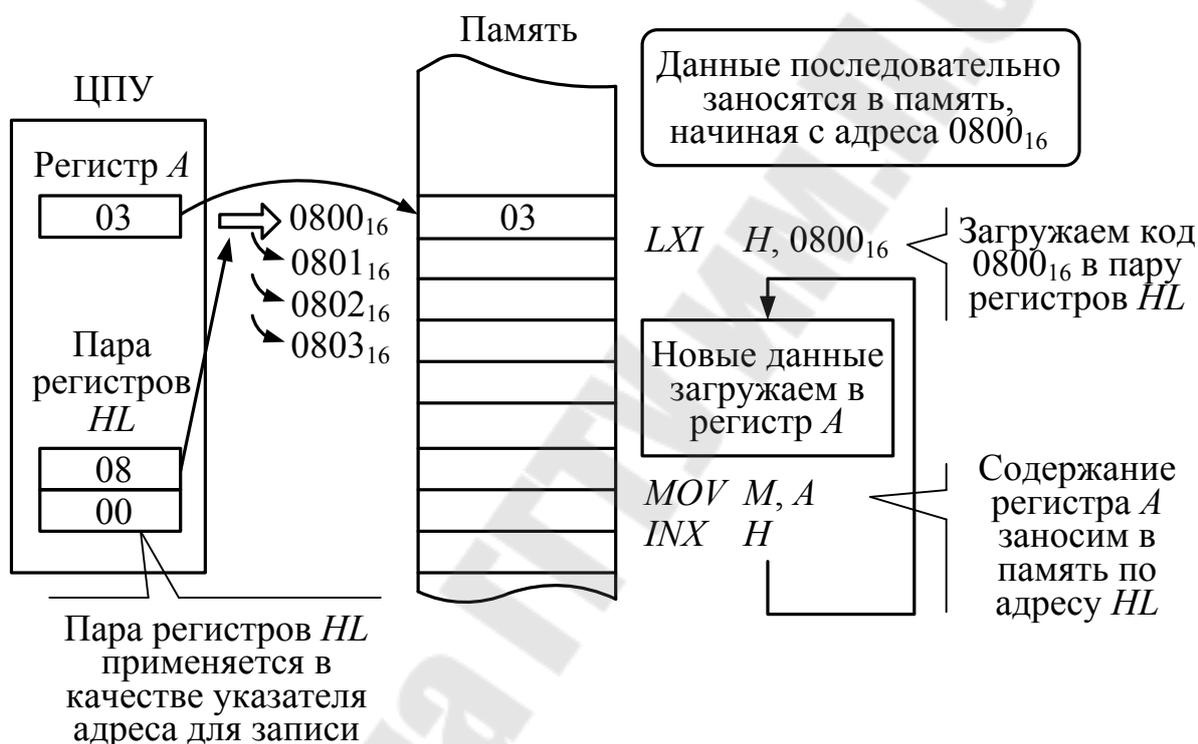


Рис. 2.3. Порядок загрузки данных в память

Для того чтобы организовать последовательную запись данных, поступающих из порта ввода, в ОЗУ ЦПУ необходимо занести адреса ячеек памяти, в которые будет производиться запись. В качестве указателя адресов, по которым будет производиться последовательная загрузка данных, используем сдвоенный регистр HL .

Указатель адреса работает следующим образом. Вначале в сдвоенный регистр HL загружается число 0800₁₆. Это число и будет первым адресом области памяти, в которую загружаются данные. Затем после каждой очередной загрузки данных по адресу, указанному в паре регистров HL , содержимое этой пары регистров увеличивается на 1

(команда «*INX H*»). Таким образом, содержимое указателя (пары регистров *HL*) будет всегда соответствовать адресу ячейки памяти, отведенной для загрузки очередной «порции» данных.

Блок 3 схемы алгоритма программы (см. рис. 2.2) проверяет, изменилось ли состояние контактов хотя бы одного из шести выключателей $PS_1 \dots PS_6$. Для этого данные предыдущей операции считывания из порта ввода, предварительно записанные, например, в регистр *B*, сравниваются с текущим состоянием на его входах (оператор условного перехода).

В блоке 4 проверяется состояние выключателя SW_7 (сигнализатора окончания записи сигналов). Если этот выключатель находится в положении «выключено», то повторяются операции, представленные на схеме блоками 2 и 3. Блок 5 схемы обозначает запись в память результатов последнего считывания из порта ввода, когда контакты выключателя SW_7 оказываются в замкнутом состоянии. Данные, считанные в этом цикле, используются в процедуре воспроизведения как информация о последнем сигнале.

Если описанный выше алгоритм записать в мнемокодах, то получим программу, представленную на рис. 2.4.

Процедура воспроизведения последовательности сигналов

Воспроизведение должно начаться после окончания процедуры записи команд по сигналу, поступившему с выключателя SW_7 , в момент размыкания его контактов. Данные, представляющие собой последовательность записанных команд, загружены в память начиная с адреса 0800_{16} . При воспроизведении они побайтно считываются и поочередно транслируются в порт вывода с определенным интервалом.

Схема алгоритма этой процедуры и соответствующая ей программа на ассемблере приведены на рис. 2.5. Блок 1 периодически проверяет состояние контактов выключателя SW_7 в ожидании момента, когда они перейдут из положения «замкнуто» в положение «разомкнуто».

Блок 2: в пару регистров *HL* заносим первый адрес памяти (0800_{16}), начиная с которого были загружены данные в процессе записи последовательности сигналов из порта ввода.

Блок 3: используя пару регистров *HL* в качестве указателя, поочередно считываем данные из памяти и пересылаем их в порт вывода.

Главная программа

Подпрограмма

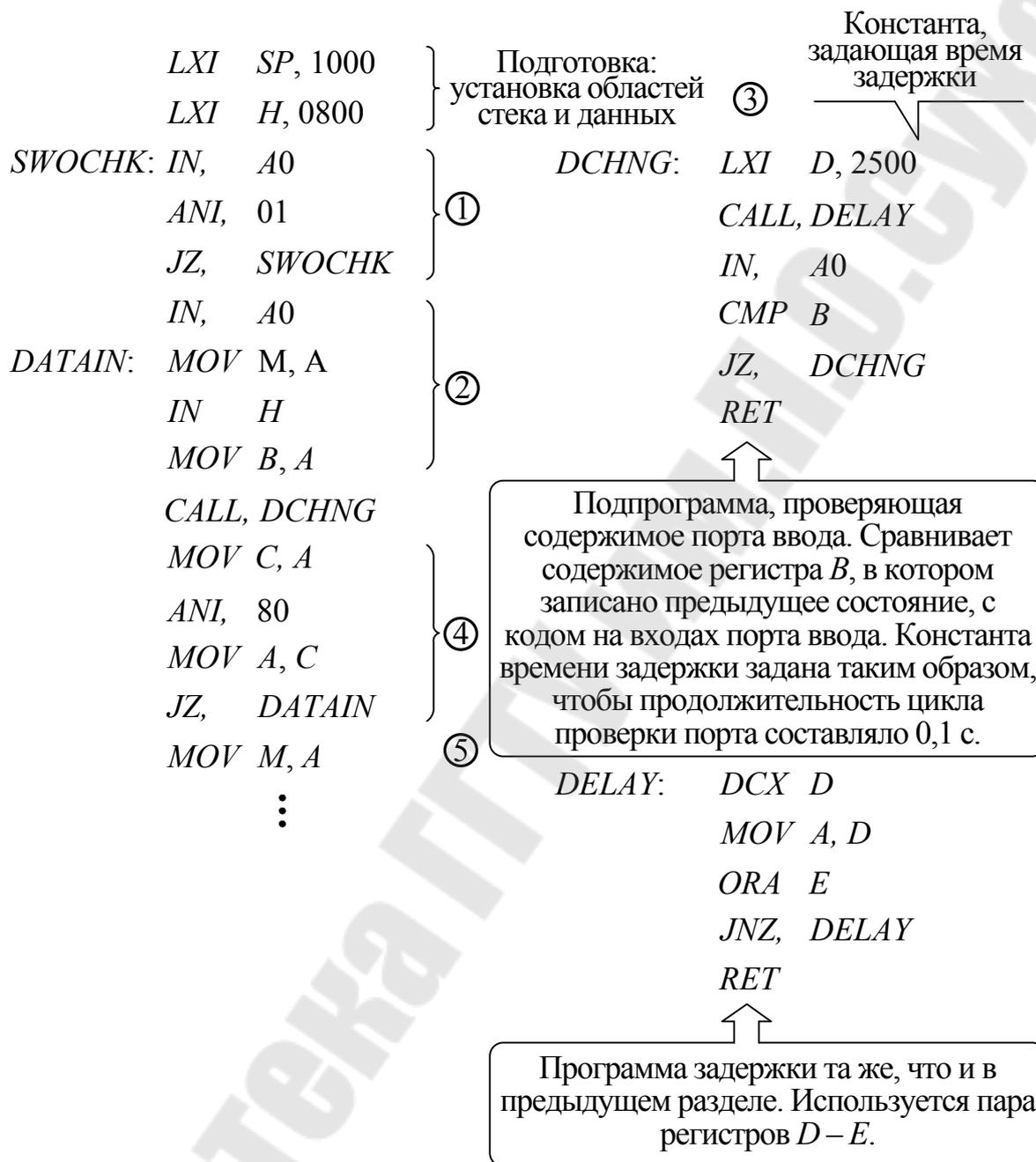


Рис. 2.4. Программа процедуры «запомнить последовательность подачи сигналов»

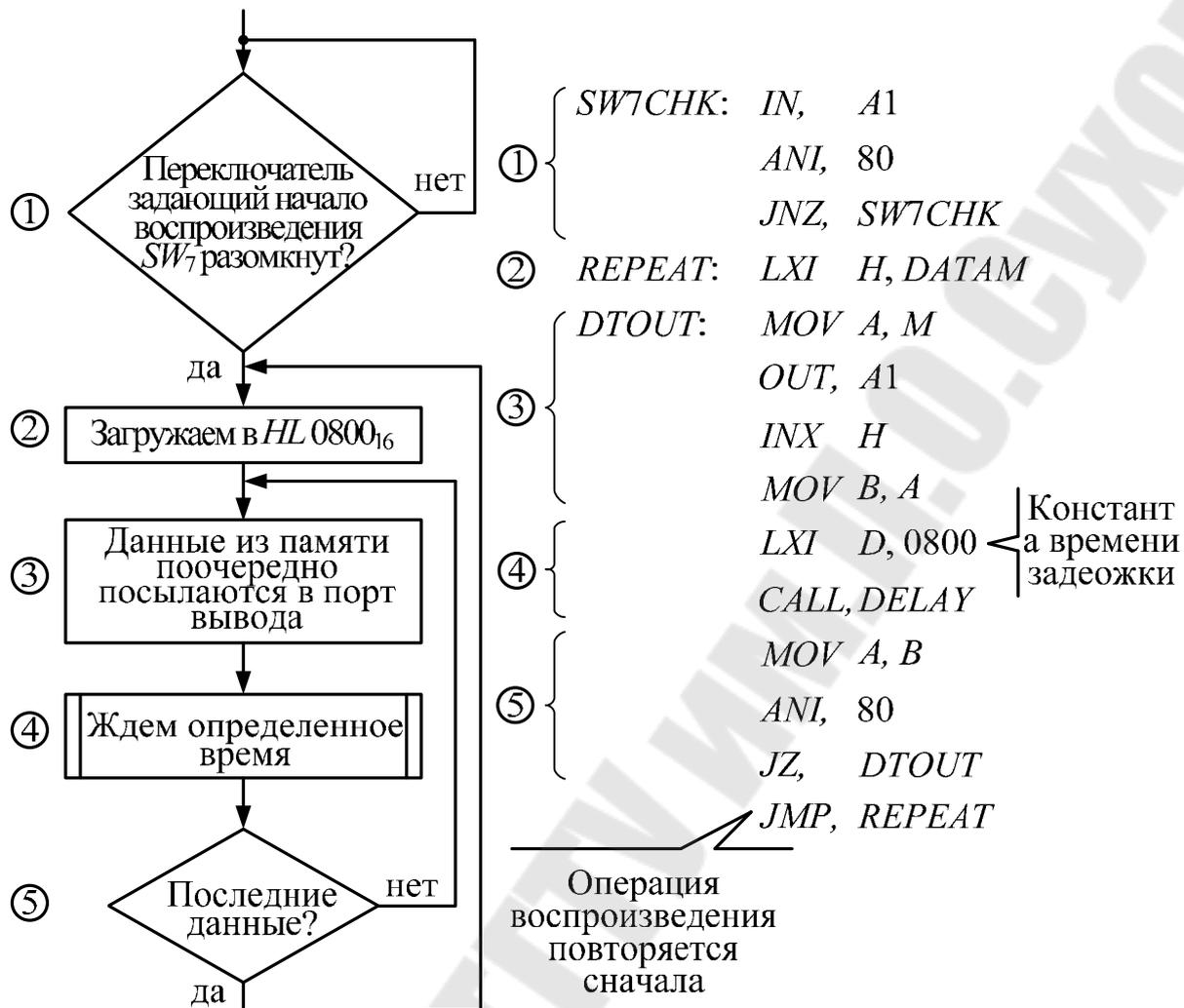


Рис. 2.5. Алгоритм и программа процедуры воспроизведения

Блок 4 осуществляет задержку на некоторое установленное время между циклами пересылки данных в порт вывода. Здесь используется та же подпрограмма (метка *DELAY*), что и в программе «запомнить последовательность подачи команд». Можно регулировать длительность интервала, изменяя константу времени задержки.

Функция блока 5: определять момент окончания процедуры воспроизведения. Если 7-й (старший) разряд числа, считанного из памяти, равен «1», операция воспроизведения считается законченной. Последняя команда «*JMP REPEAT*» обеспечивает повторение с самого начала процедуры воспроизведения.

Подобная программа управления устройством, воспроизводящим порядок изменения входных сигналов, применяется, например, в

контроллере печатного устройства, в частности, печатной машинки. Порядок нажатия клавиш печатной машинки загружается в микроконтроллер при вводе в него текста с клавиатуры, а затем может быть воспроизведен (причем с гораздо более высокой скоростью) в процессе вывода текста на печать.

Задание для самостоятельной работы

1. Сигналы подаются с помощью 6 выключателей, подключенных к порту ввода. Последовательность подачи сигналов с контактов выключателей из порта ввода заносится в память. Через порт вывода контакты шести реле в обратной последовательности воспроизводят сигналы контактов выключателей. При воспроизведении сигналов осуществляется задержка (приблизительно 1 с) между командами замыкания реле. Начало и окончание записи сигналов задаются посредством отдельных выключателей.

2. Последовательность подачи сигналов, задаваемая с помощью шести выключателей, из порта ввода заносится в память с учетом интервала времени между ними. Посредством контактов шести реле, подключенных к порту вывода, эта последовательность воспроизводится в обратном порядке с удвоением интервалов между сигналами. Схема подключения выключателей к порту ввода, а также светодиодов и реле к порту вывода такая же, как и в предыдущей задаче (см. рис. 2.1)

Тема 3. Программы с использованием таблицы данных

Постановка задачи

Задана таблица соответствия между 4-разрядными входными данными и временем свечения светодиода. При замыкании переключателя (рис.3.1) считываются входные данные и зажигается светодиод. Время в секундах, в течение которого он светится, равно произведению соответствующего значения из таблицы на 0,1.

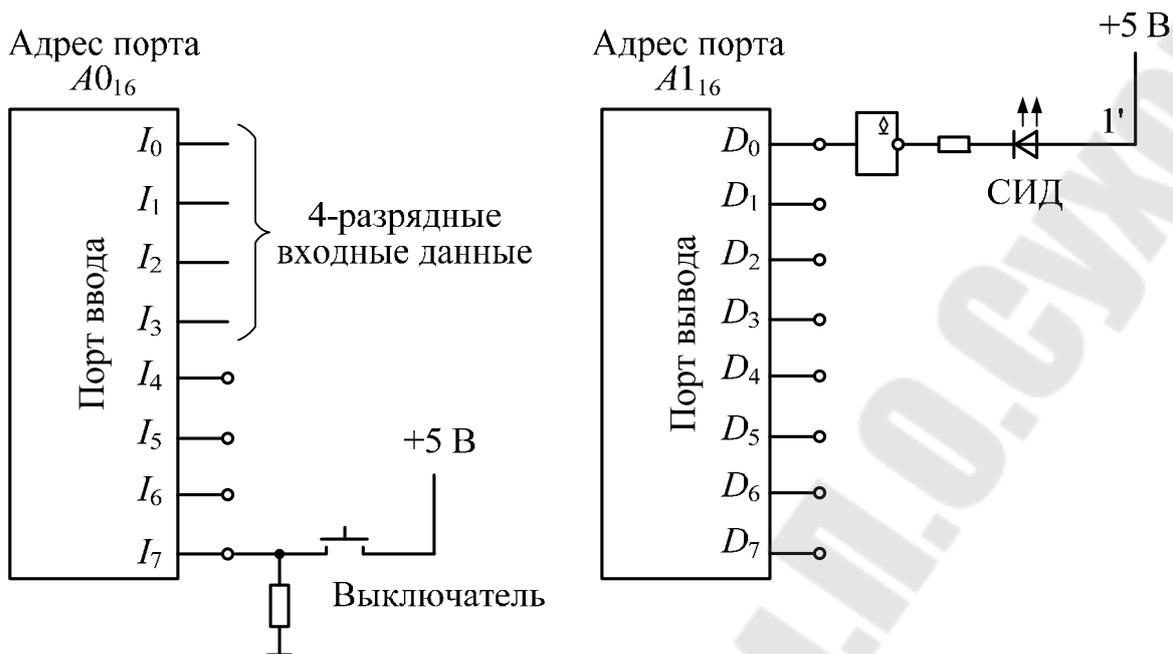


Рис. 3.1. Подача и считывание данных из портов ввода-вывода

Решение задачи

Чтобы понять смысл задачи, сначала разберемся в работе схемы, представленной на рис. 3.1. На схеме показаны порты ввода-вывода микропроцессора. К 7-му разряду порта ввода (адрес $A0_{16}$) присоединен кнопочный выключатель, с помощью которого подается сигнал начала процедуры управления. На входы порта ввода, соответствующие разрядам с 0-го по 3-й, подаются входные данные (4-разрядные двоичные коды). К выходу 0-го разряда порта вывода (адрес $A1_{16}$) через инвертор подключен светодиод, который загорается, когда сигнал на выходе этого разряда равен «1».

Теперь, когда мы ознакомились со схемой подключения цепей внешних устройств к портам ввода-вывода, можно сформулировать задачу для микроконтроллера, которая состоит в том, чтобы устанавливать продолжительность свечения светодиода в зависимости от сигнала на входах $I_0 \dots I_3$. Эта зависимость задана с помощью таблицы (табл. 3.1), в первой колонке которой даны числовые значения сигнала на входе порта ввода, а во второй — соответствующие им временные множители, определяющие продолжительность работы светодиода.

Таблица данных

4-разрядный входной сигнал (двоичное число)	Множитель (шестнадцатеричное число)
0 (0000)	100 (64)
1 (0001)	95 (5F)
2 (0010)	90 (5A)
3 (0011)	75 (4B)
4 (0100)	65 (41)
5 (0101)	60 (3C)
6 (0110)	55 (32)
7 (0111)	40 (28)
8 (1000)	35 (23)
9 (1001)	30 (1E)
10 (1010)	25(19)
11 (1011)	20(14)
12 (1100)	18(12)
13 (1101)	16(10)
14 (1110)	14 (0E)
15 (1111)	12 (0C)

Время свечения светодиода (в секундах) должно быть равно значению множителя, приведенного во второй колонке таблицы, умноженного на 0,1. Загрузку табличных значений временных множителей в память микроконтроллера будем производить, начиная с некоторого адреса в той же последовательности, в которой они приведены в табл. 3.1 (100, 95 ... 14, 12).

Таким образом, значение временного множителя, соответствующее некоторому двоичному числу на входе порта ввода, будет размещаться по адресу, равному сумме этого числа и начального адреса таблицы. В программе нужно указать, начиная с какого адреса загружается таблица.

Схема алгоритма управления светодиодом, состоящая из 9 блоков, изображена на рис. 3.2.

Блок 1: проверка состояния контактов кнопочного выключателя, замкнутое состояние которых должно стать сигналом к началу процедуры.

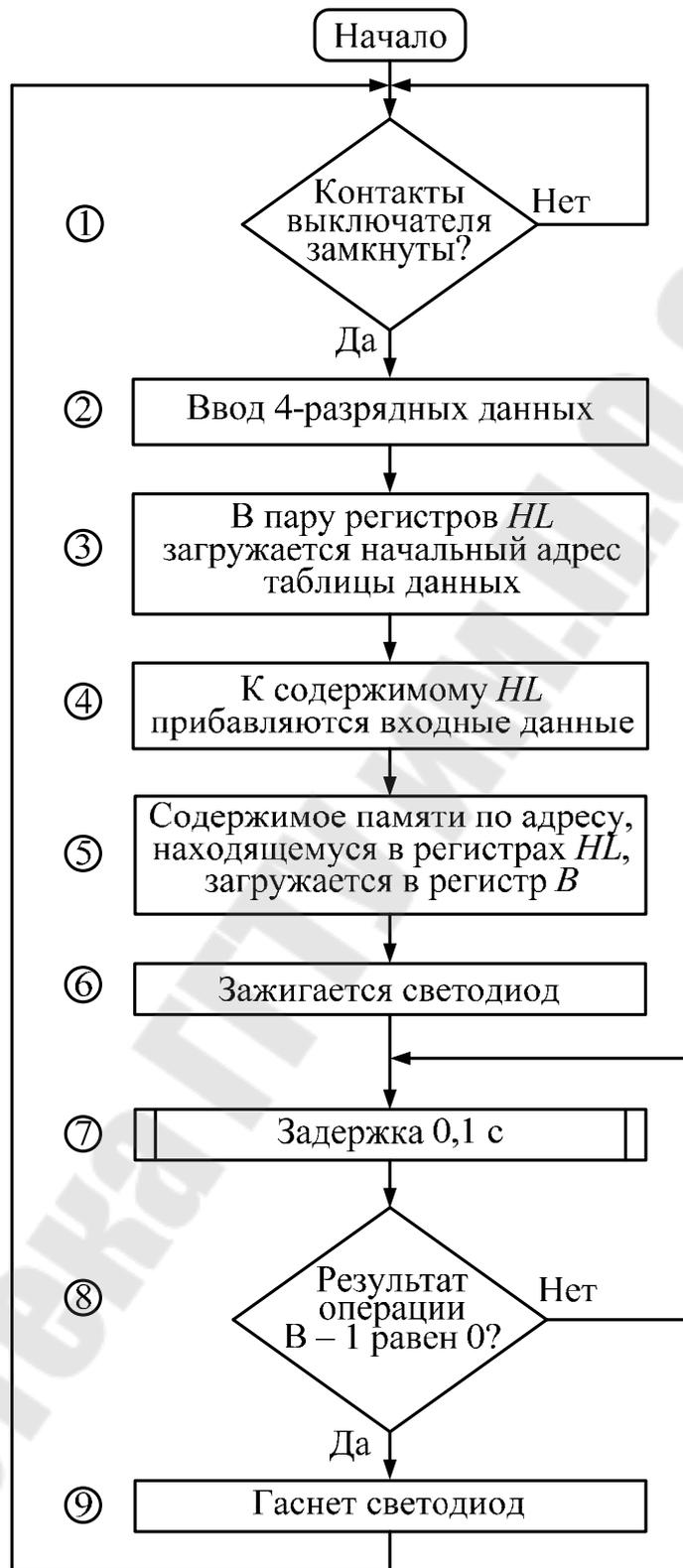


Рис. 3.2. Схема алгоритма

Блок 2: из порта ввода считываются входные данные.

Блок 3: в пару регистров *HL*, используемых в качестве указателя, заносится начальный адрес таблицы данных.

Блок 4: к текущему значению адреса прибавляется содержимое порта ввода.

Блок 5: содержимое ячейки памяти (временной множитель) по адресу, полученному в результате операции сложения и записанному в парный регистр *HL*, загружается в регистр *B*.

Блок 6: подается команда на включение светодиода.

Блок 7: отработка цикла задержки 0,1 с.

Блок 8: из содержимого регистра *B* вычитается 1. Если в результате операции вычитания содержимое регистра *B* не стало равно 0, повторяется цикл задержки с последующим вычитанием 1 из содержимого регистра *B*. Поскольку в регистр *B* был загружен временной множитель (соответствующий данной комбинации входных сигналов порта ввода), количество циклов задержки будет равно значению временного множителя.

Блок 9: отработка команды выключения светодиода и возврат в состояние ожидания замыкания контактов кнопочного выключателя.

Текст программы на языке ассемблера, соответствующей рассмотренному алгоритму, представлен на рис. 3.3. Фигурными скобками с цифрами, как и ранее, обозначены соответствующие блоки алгоритма.

В этой программе много функциональных блоков, идентичных тем, которые мы рассматривали в предыдущих задачах. Будет нелишним сделать несколько пояснений. Команда «*ANI, 0F₁₆*» в блоке 2 нужна для того, чтобы, сохранив значения четырех младших разрядов введенных данных, обнулить все остальные разряды.

Символическая константа *ТАВТОР* в блоке 3 определяет начальный адрес, по которому загружены числовые значения из второй колонки таблицы данных. В блоке 4 к содержимому пары регистров *HL*, представляющему собой 16-разрядное число, прибавляется число, находящееся в регистре *A*. Сначала суммируется содержимое регистров *A* и *L*, и если был перенос разряда, то к содержимому регистра *H* прибавляется 1. Так выполняется сложение 16-разрядного и 8-разрядного чисел. Для того чтобы реализовать задержку, соответствующую временному множителю, загруженному в регистр *B* (блок 5), в блоке 8 применяется команда «*DJNZ*». По команде «*DJNZ*» из содержимого регистра *B* вычитается 1 и, если результат отличен от 0, производится процедура задержки «*DELAY*». На ее выполнение про-

цессору требуется время 0,1 с. В блоке 9 по команде «XOR A», производится логическая операция «Исключающее ИЛИ». При этом в качестве обоих операндов выступает одно и то же число, содержащееся в регистре A, в результате чего регистр A обнуляется.



Рис. 3.3. Текст программы

Последние в тексте программы команды, начиная с команды «TABTOP: 64H, ...», означают последовательную загрузку 1-байтовых чисел, равных временным множителям таблицы данных, в память ЦПУ, начиная с адреса, указанного меткой TABTOP.

Рассмотренный принцип построения программы часто используется при вводе данных с разнообразных датчиков, например, когда нужно изменить число оборотов двигателя. Этот прием можно применять и тогда, когда по введенному числу мы хотим выполнить обработку данных разными программами. В этом случае мы задаем в виде таблицы соотношение между вводимым числом и начальным адресом программы, соответствующей этому числу.

Задание для самостоятельной работы

Задана таблица соответствия между 4-разрядными входными данными и временем свечения светодиода (см. табл. 3.1). При замыкании переключателя считываются входные данные, и зажигается светодиод. Время в секундах, в течение которого он светится, равно произведению соответствующего значения из таблицы на 0,5.

Тема 4. Управление шаговым двигателем

Постановка задачи

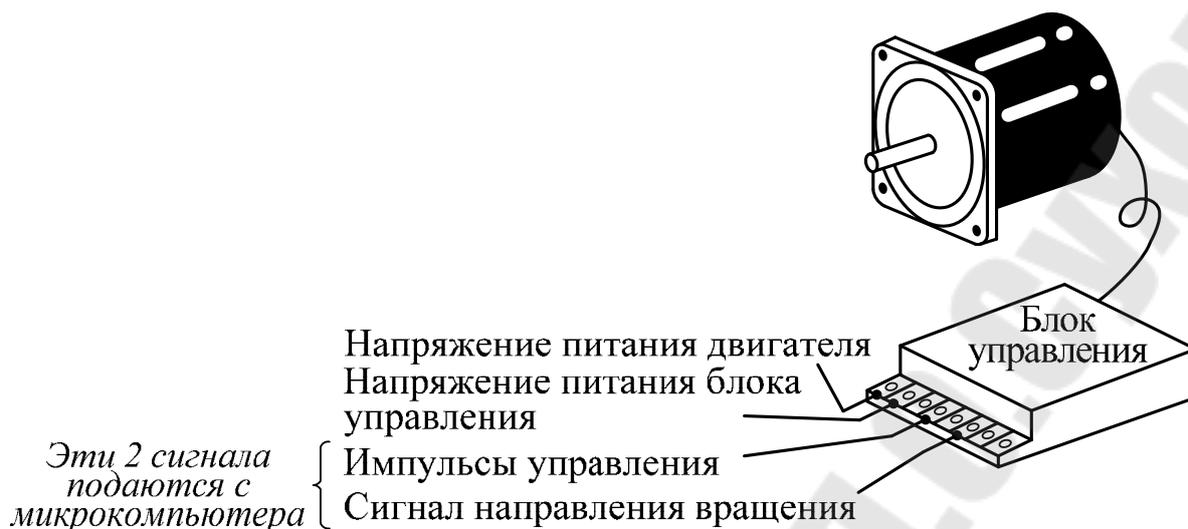
При замкнутых контактах выключателя вал шагового двигателя равномерно поворачивается на 180° по часовой стрелке, а затем таким же образом на 180° против часовой стрелки. Эта процедура повторяется многократно. Поворот в одну сторону производится за 2 с.

Решение задачи

Внешний вид шагового двигателя с блоком управления показан на рис. 4.1.

Шаговый двигатель работает только в паре с блоком управления, на который необходимо подать напряжение питания двигателя (например, 15 В), питание схемы управления (5 В), управляющие импульсы и сигнал, задающий направление вращения. Амплитуда управляющих импульсов и сигнала, задающего направление вращения, хорошо согласуется с уровнями сигналов *TTL*. Поэтому их можно сформировать на выходах порта вывода микроконтроллера и использовать для передачи непосредственно в блок управления шаговым двигателем.

Согласно техническим данным двигателя, представленного на рис. 4.1, при подаче одного импульса управления его вал поворачивается на угол $1,8^\circ$. Следовательно, полный оборот вала (360°) осуществляется в результате воздействия 200 таких импульсов. Частота импульсов управления для двигателей данного типа не должна превышать 300 Гц. Предположим, требуется обеспечить поворот вала двигателя на 180° примерно за 2 с. Так как для поворота вала на 180° требуется подать 100 импульсов управления, их частота должна быть приблизительно равна 50 Гц.



Технические данные

Напряжение питания	60 В (<i>max</i>)
Потребляемый ток в импульсе	4,5 А (<i>max</i>)
Напряжение питания схемы управления	5 В
Угол поворота вала на 1 импульс	1,8°
Частота импульсов управления	0 ... 300 Гц
Длительность импульса управления	3 мкс (<i>min</i>)
Длительность фронта импульса управления	2 мкс (<i>max</i>)
Уровни сигнала управления	ТТЛ

Рис. 4.1. Шаговый двигатель и блок управления

Завод-изготовитель гарантирует стабильность работы двигателя, если длительность импульса управления составляет не менее 3 мкс. Шаговый двигатель данного типа совершает один шаг поворота вала по отрицательному (спадающему) фронту импульсов управления.

Схема подключения шагового двигателя и кнопки включения к портам ввода-вывода показана на рис. 4.2. Составим программу таким образом, чтобы сигнал направления вращения подавался с нулевого разряда порта вывода, а импульсы управления — с первого разряда этого порта.

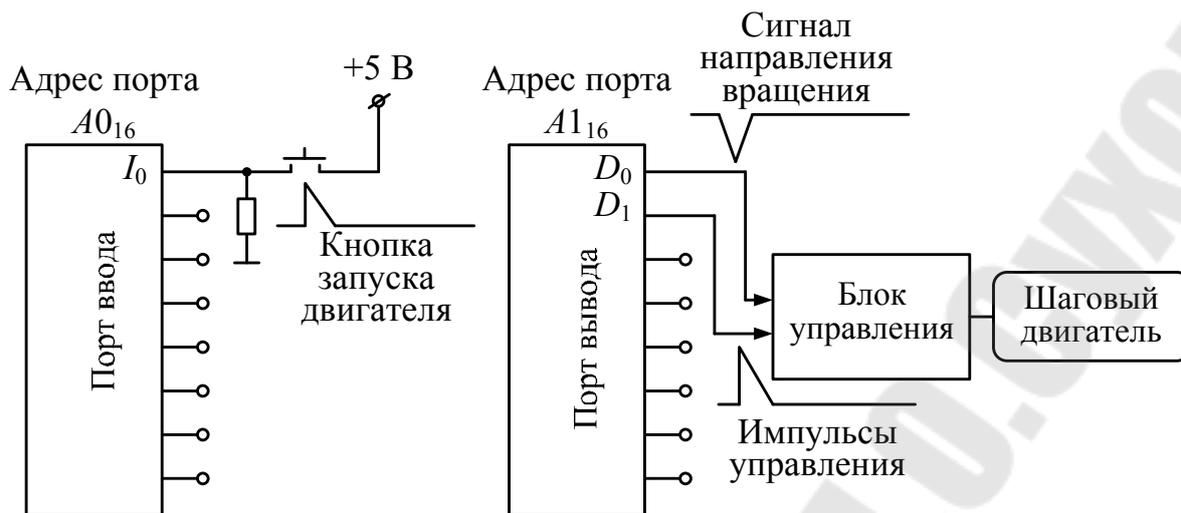


Рис. 4.2. Конфигурация портов ввода-вывода

Импульсы управления

Для того чтобы за 2 с вал двигателя повернулся на 180°, нужно, чтобы за это время было подано 100 управляющих импульсов. То есть период следования импульсов должен быть равен 20 мс. На рис. 4.3 показана временная диаграмма сигнала, который нужно подать на блок управления шаговым двигателем.

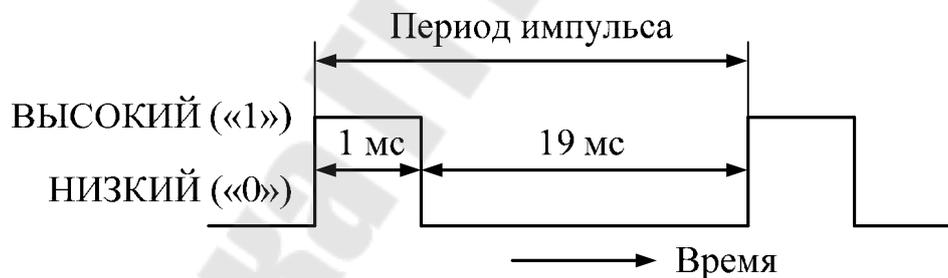


Рис. 4.3. Временная диаграмма сигнала управления шаговым двигателем

Длительность импульса управления выбрана равной 1 мс, что значительно превышает минимально допустимое значение этого параметра (3 мкс). Таким образом, чтобы программно реализовать эту последовательность импульсов, нужно в первом разряде порта вывода в течение 1 мс сохранять состояние логической «1», а затем в течение 19 мс — состояние «0». Эту последовательность требуется повторить 100 раз.

Описание программы

Будем считать прямым такое направление вращения двигателя, когда на блок управления с выхода D_0 порта вывода подается сигнал «0». Соответственно при подаче сигнала «1» будет происходить обратное вращение. Внимательно изучив схему алгоритма решения данной задачи, представленную на рис. 4.4, несложно понять назначение ее блоков, которые обозначают соответствующую последовательность операций, необходимую для выполнения задачи. Теперь, чтобы понять, как работает программа, сравним текст программы со схемой алгоритма.

Команды программы, соответствующие блокам 1 и 2 схемы алгоритма, мы уже разбирали ранее, поэтому не будем останавливаться на них. Блок 3: число 100 (64_{16}), равное количеству импульсов, которые нужно подать на блок управления, чтобы повернуть вал двигателя на 180° , загружается в регистр B . Блок 4: команда «*ORI, 02*» поразрядно выполняет логическую операцию «ИЛИ» с содержимым регистра A и двоичным числом 00000010 (02_{16}), после чего второй разряд числа, представляющего результат логического сложения (он находится в регистре A), устанавливается в «1». Блок 5: выполняется цикл процедуры задержки (1 мс). Так как время задержки сравнительно невелико, эта процедура организована с помощью операторов *DCR* и *JNZ*. Для отсчета времени используется регистр B , поэтому, чтобы сохранить содержимое этого регистра (заданное число импульсов управления) перед входом в блок 5 и после выхода из него, выполняются соответственно команды «*PUSH B*» и «*POP B*» (содержимое пары регистров BC сначала загружается в стек, а после выполнения цикла задержки снова выгружается в регистр B).

Блок 6: команда «*ANI FD*» выполняет операцию, обратную команде «*ORI, 02*», а именно, сбрасывает в «0» значение второго разряда регистра A . Блок 7: вызов подпрограммы задержки (метка *DELAY*), аналогичной той, которую мы уже неоднократно применяли. В данном случае время задержки составляет 19 мс. Блок 8: команд *DCR* в *JNZ*, которые еще раз применяются к содержимому регистра B , проверяют, было ли выдано 100 импульсов управления.

Блок 9: формируется команда изменения направления вращения вала. Для этого посредством команды «*XOR, 01*» инвертируется первый разряд регистра A , то есть происходит замена «0» на «1» или наоборот.

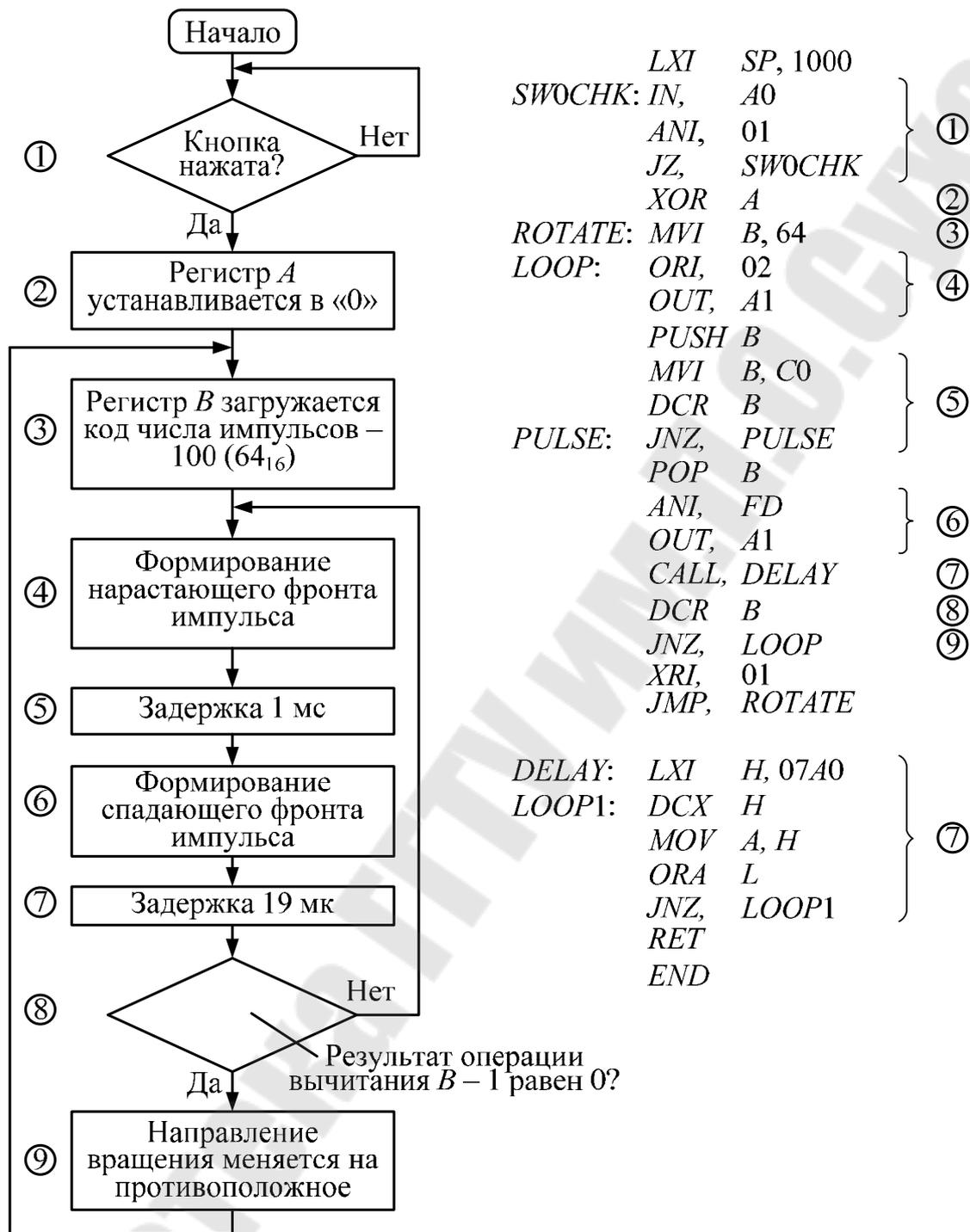


Рис. 4.4. Программа и схема алгоритма

В рассмотренном выше примере осуществляется управление вращением вала шагового двигателя с постоянной скоростью на 180° . Можно произвольно задавать угол поворота вала двигателя, для чего в тексте программы, приведенной на рис. 4.4, изменить число (64_{16}) подаваемых импульсов (блок 3). Чтобы изменить скорость вращения,

нужно задать другое значение константы ($07A0_{16}$), определяющей время задержки (блок 7). При этом единственным ограничением является заявленная заводом-изготовителем предельная частота импульсов управления для данного типа шагового двигателя.

Задание для самостоятельной работы

К входу I_0 порта ввода, имеющего адрес 30_{16} , присоединен выключатель. Зная, что состоянию «включено» соответствует логический сигнал «1», а состоянию «выключено» — сигнал «0», составьте программу на ассемблере для загрузки в память ЦПУ значений интервалов времени между моментами смены состояний переключателя («выключено» — «включено» — «выключено»). Вычисления производить с точностью 1 мс. Тактовая частота ЦПУ — 2,5 МГц, время нахождения переключателя в состоянии «включено» не превышает 60 с.

Тема 5. Управление охранной сигнализацией

Постановка задачи

Требуется создать установку охранной сигнализации, имеющую несколько контуров сигнализации и функции включения/выключения с задержкой времени:

- с помощью выключателя с ключом установка переводится в охранное состояние по истечении времени задержки длительностью 20 секунд. Благодаря этой задержке остается время на то, чтобы покинуть дом. В это время установка показывает, замкнуты ли сигнальные контуры;

- при обрыве сигнального контура должна срабатывать сигнализация (принцип нормально замкнутой цепи, т. е. сигнализация срабатывает и в случае саботажа). Дополнительно установка должна показывать, какой сигнальный контур вызвал сигнализацию;

- по истечении времени ожидания длительностью 10 секунд должен включаться звуковой сигнал и сигнальный фонарь. (Задержка срабатывания сигнализации нужна для того, чтобы пользователь имел возможность отключить сигнализацию, вернувшись домой. По этой причине имеется также отдельная лампа, которая показывает, находится ли установка в охранном режиме);

- звуковой сигнал должен звучать 30 секунд. Однако световая сигнализация должна оставаться включенной до дезактивации установки охранной сигнализации;

- должна иметься возможность сброса сигнализации с помощью выключателя с ключом.

Определение входных и выходных сигналов

В качестве следующего шага необходимо установить, какие входные и выходные сигналы должны обрабатываться. Из описания принципа работы установки видно, что для управления установкой нужен выключатель с ключом и 4 сигнальные лампы. Кроме того, используются еще как минимум три входа для сигнальных контуров и два выхода для звукового сигнала и проблескового фонаря. В общей сложности используются 4 входа и 6 выходов. После этого сигналы сопоставляются входам и выходам контроллера (табл. 5.1).

Таблица 5.1

Сигналы входа и выхода контроллера

Функция		Обозн	Адрес	Примечание
В х о д ы	Установка в охранном режиме	S1	X1	Замыкающий контакт (выключатель с ключом)
	Сигнальный контур 1	S11, S12	X2	Размыкающие контакты (сигнализация активируется, если вход имеет состояние «0».)
	Сигнальный контур 2	S21, S22	X3	
	Сигнальный контур 3	S31, S32	X4	
В ы х о д ы	Индикация «Охранный режим»	H0	Y0	Функция выходов выполняется, если соответствующий выход включается. Например, если включается Y1, раздается звуковой сигнал.
	Звуковой сигнал (сирена)	E1	Y1	
	Световая сигнализация (круговой фонарь)	H1	Y2	
	Индикация сигнального контура 1	H2	Y3	
	Индикация сигнального контура 2	H3	Y4	
	Индикация сигнального контура 3	H4	Y5	

Программирование

Теперь можно программировать. Нужны ли маркеры и сколько, обычно становится ясным лишь во время программирования. Заранее известно, что в этой установке сигнализации важную роль играют три устройства выдержки времени. В системе управления, основанной на физическом монтаже, применяются реле времени. В программируемом контроллере выдержка времени реализуется электронным способом. Эти «таймеры» можно установить уже до программирования (табл. 5.2).

Таблица 5.2

Выдержка времени

Функция		Адрес	Примечание
Датчик времени	Задержка постановки на охранный режим	T0	время: 20 секунд
	Задержка срабатывания сигнализации	T1	время: 10 секунд
	Время работы звукового сигнала	T2	время: 30 секунд

После этого решаются отдельные частичные задачи управления. Перевод установки в охранный режим с задержкой (рис. 5.1)

Релейная диаграмма



Список инструкций

```

0  LD   X001
1  OUT  T0    K200
4  LD   T0
5  OUT  Y000
    
```

Рис. 5.1. Пример перевода установки в охранный режим с задержкой

После включения выключателя с ключом датчик времени T0 отсчитывает реализованную задержку включения. По истечении 20 секунд ($K200 = 200 \times 0,1 \text{ с} = 20 \text{ с}$) подключенная к выходу Y000 контрольная лампа показывает, что установка включена.

Контроль сигнальных контуров и распознавание сигнализации (рис. 5.2).

В программе опрашивается также выход Y000 - чтобы определить, включена ли установка охранной сигнализации. Для этого можно было бы также использовать маркер, включающийся и выключающийся параллельно выходу Y000. Только в том случае, если установка находится в охранном режиме, при обрыве сигнального контура устанавливается маркер M1, который показывает, что сработала сигнализация. Дополнительно выходы CY003 по Y005 показывают, какой сигнальный контур был прерван. Маркер M1 и соответствующий выход остаются включенными и после того, как сигнальный контур был снова замкнут.

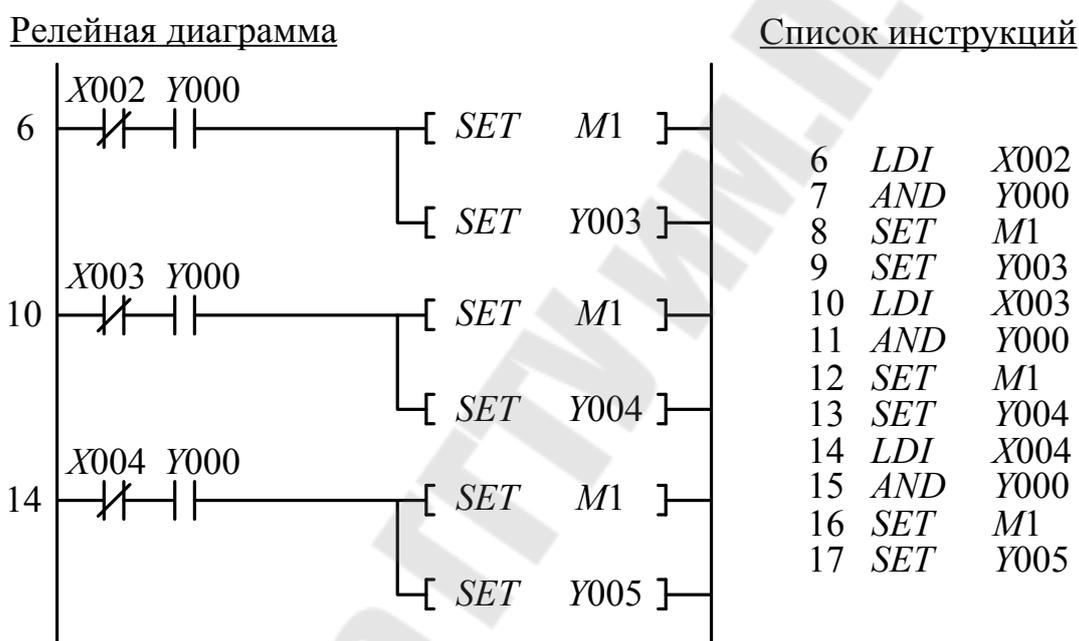


Рис. 5.2. Пример контроля сигнальных контуров и распознавание сигнализации

Задержка срабатывания сигнализации (рис. 5.3).

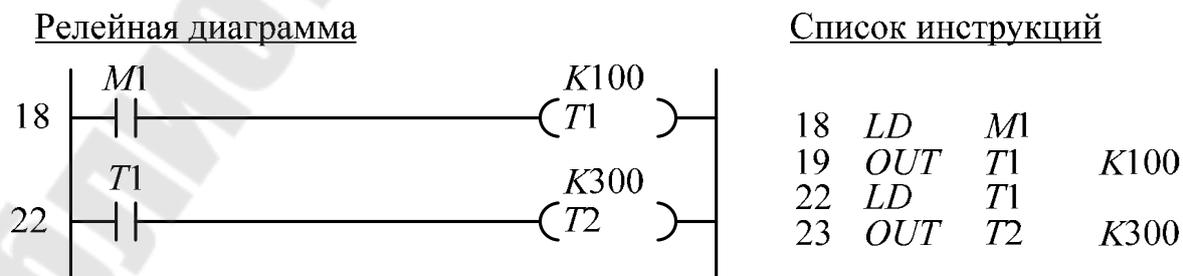


Рис. 5.3. Пример задержки срабатывания сигнализации

Если сигнализация сработала (M1 в этом случае имеет состояние «1»), начинает отсчитываться время задержки 10 секунд. По истечении этого времени T1 запускается датчик времени T2, который настроен на 30 секунд (время включенного состояния сирены).

Проявление сигнализации (включение сирены и кругового фонаря) (рис. 5.4).

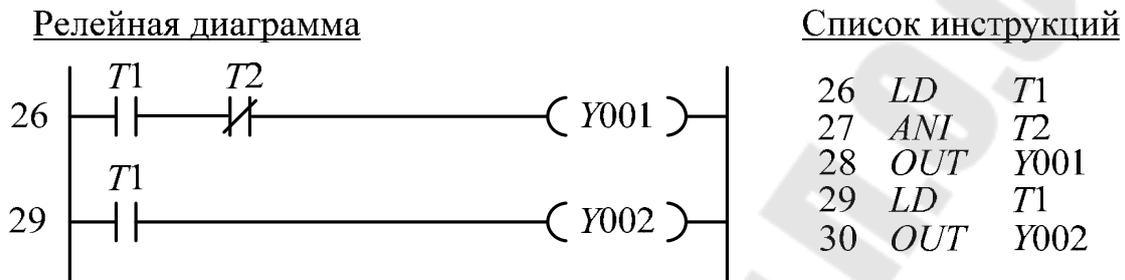


Рис. 5.4. Пример включения сирены и кругового фонаря

После истечения задержки включения длительностью 10 секунд (T1) и до тех пор, пока таймер T2 еще работает, включается сирена. По истечении 30 секунд (T2) сирена умолкает. Круговой фонарь также включается через 10 секунд. На рис. 5.5 показана диаграмма изменения сигналов для этой части программы:

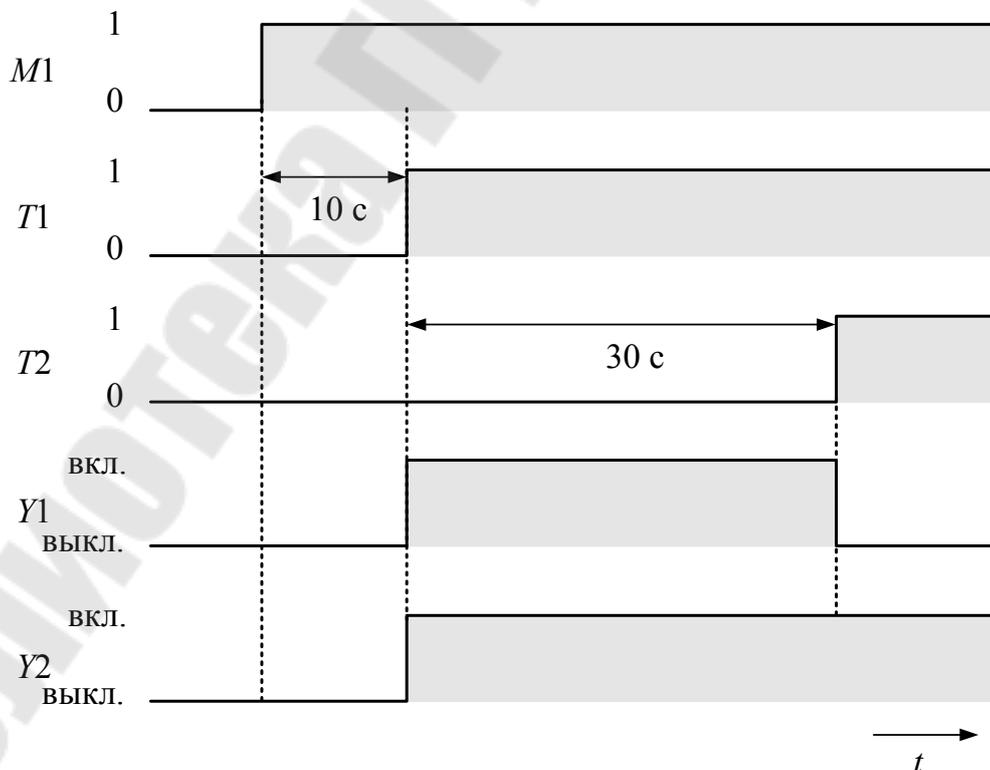
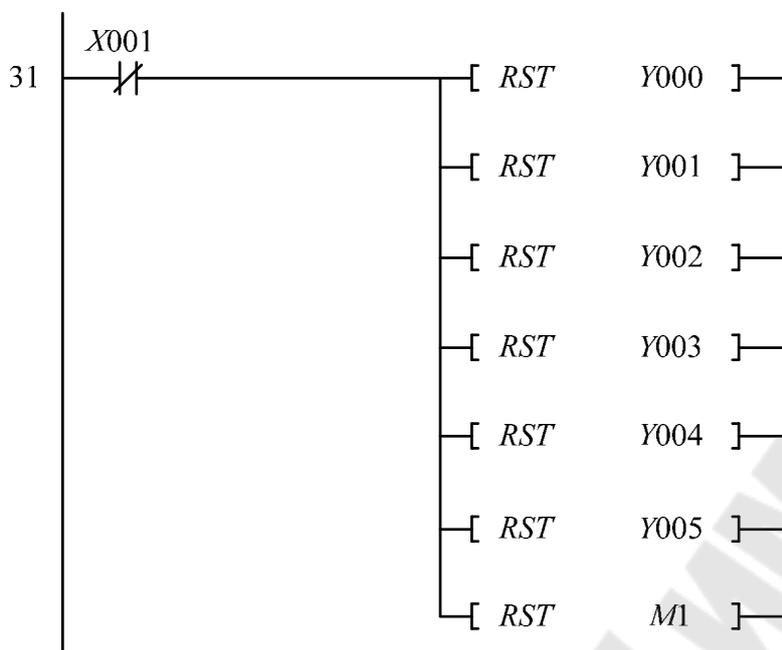


Рис. 5.5. Диаграмма изменения сигналов

Сброс всех выходов и маркеров (рис. 5.6).

Релейная диаграмма



Список инструкций

```
31 LDI X001
32 RST Y000
33 RST Y001
34 RST Y002
35 RST Y003
36 RST Y004
37 RST Y005
38 RST M1
```

Рис. 5.6. Сброс всех выходов и маркеров

Если пользователь выключил установку охранной сигнализации с помощью выключателя с ключом, все используемые выходы и маркер M1 сбрасываются. Если ранее сработала сигнализация, то до этого момента показывается, какой сигнальный контур был прерван.

Подключение контроллера

На рис. 5.7 показано, насколько малыми затратами можно реализовать установку охранной сигнализации, например, на основе контроллера FX1N-14MR.

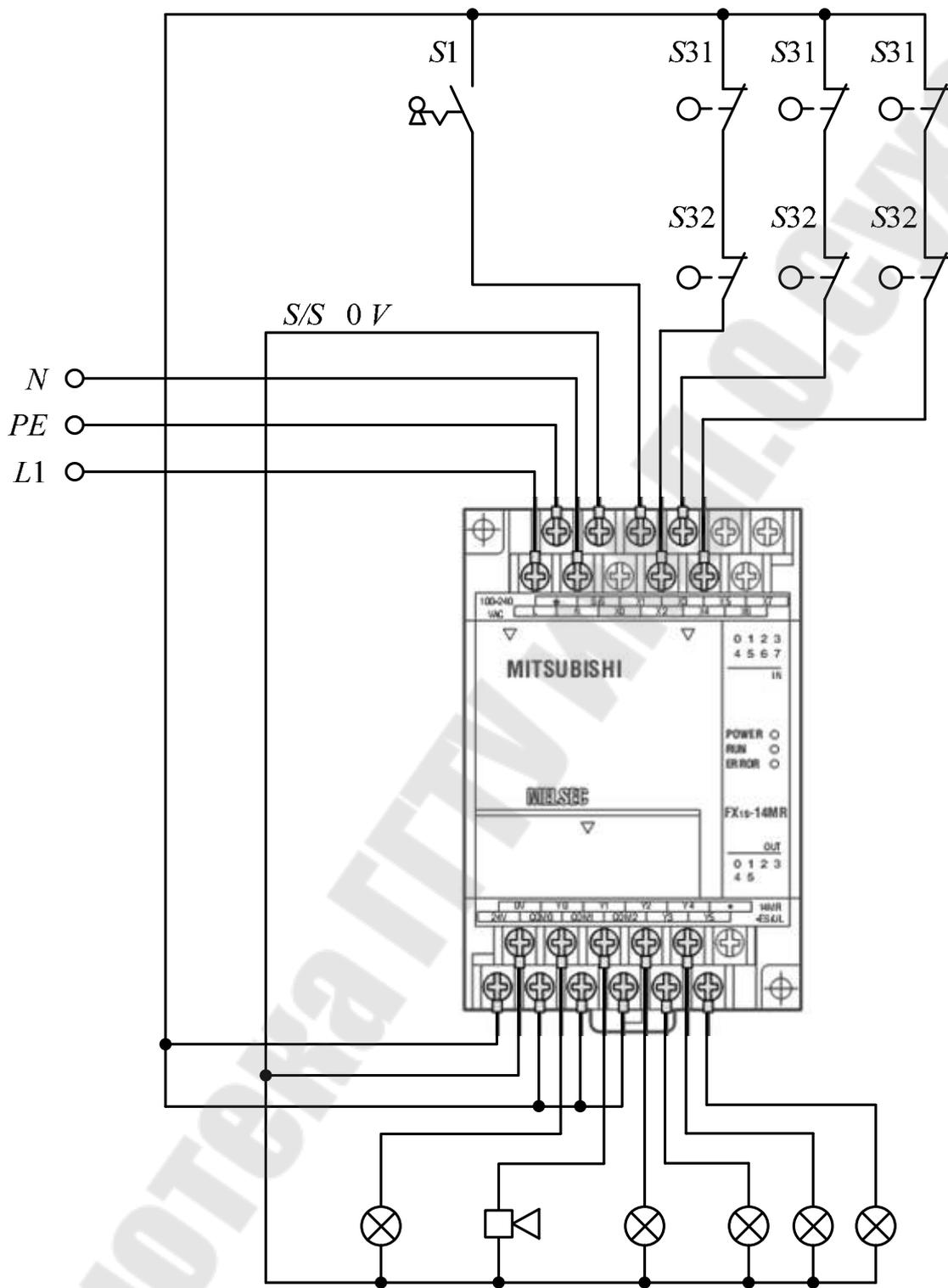


Рис. 5.7. Схема охранной сигнализации на основе контроллера FX1N-14MR

Задание для самостоятельной работы

Требуется создать установку охранной сигнализации, имеющую несколько контуров сигнализации и функции включения/выключения с задержкой времени:

- с помощью выключателя с ключом установка переводится в охранное состояние по истечении времени задержки длительностью 40 с. Благодаря этой задержке остается время на то, чтобы покинуть дом. В это время установка показывает, замкнуты ли сигнальные контуры;

- при обрыве сигнального контура должна срабатывать сигнализация (принцип нормально замкнутой цепи, т. е. сигнализация срабатывает и в случае саботажа). Дополнительно установка должна показывать, какой сигнальный контур вызвал сигнализацию;

- по истечении времени ожидания длительностью 20 с должен включаться звуковой сигнал и сигнальный фонарь. (Задержка срабатывания сигнализации нужна для того, чтобы пользователь имел возможность отключить сигнализацию, вернувшись домой. По этой причине имеется также отдельная лампа, которая показывает, находится ли установка в охранном режиме);

- звуковой сигнал должен звучать 10 секунд. Однако световая сигнализация должна оставаться включенной до дезактивации установки охранной сигнализации;

- должна иметься возможность сброса сигнализации с помощью выключателя с ключом.

Тема 6. Управление электроприводом шторных ворот

Постановка задачи

Шторные ворота склада (рис. 6.1) должны удобно управляться снаружи или изнутри. Но при этом должны учитываться и аспекты безопасности.

Управление:

- снаружи ворота должны открываться выключателем с ключом S1 и закрываться с помощью кнопки S5. Из помещения ворота должны открываться после нажатия кнопки S2 и закрываться после нажатия кнопки S4;

- кроме того, дополнительная функция управления по времени должна автоматически закрывать ворота, если они открыты дольше 20 секунд;

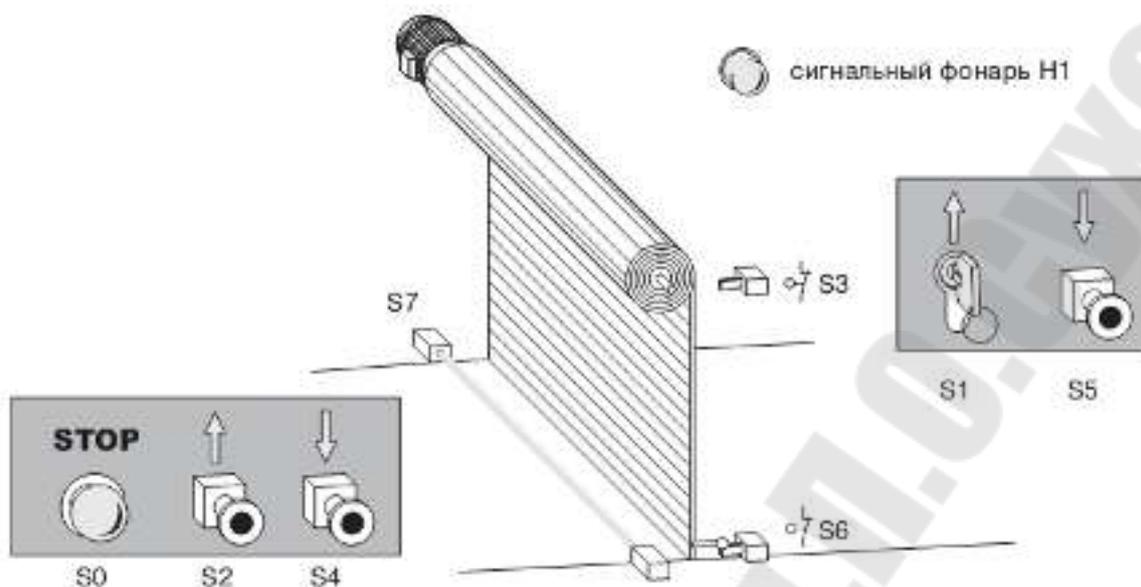


Рис. 6.1. Шторные ворота

- состояния «Ворота движутся» и «Ворота находятся в неопределенном положении» должны сигнализироваться мигающим сигнальным фонарем.

Защитные устройства:

- должна иметься возможность в любой момент остановить движение ворот кнопкой «Стоп» (S0). При этом ворота должны оставаться в занимаемом ими в данный момент положении. Эта кнопка «Стоп» не имеет функции аварийного выключения! По этой причине она обрабатывается только в контроллере и не коммутирует внешние напряжения;

- если при закрывании ворот фоторелейный барьер (S7) распознает препятствие, ворота должны автоматически открываться;

- для останова электродвигателя в обоих конечных положениях ворот предусмотрены два концевых выключателя S3 («Ворота открыты») и S6 («Ворота закрыты»).

Сопоставление входных и выходных сигналов

Для управления приводным электродвигателем нужны два выхода. Сигналы сопоставляются входам и выходам контроллера (табл. 6.1).

Таблица 6.1

Входные и выходные сигналы контроллера

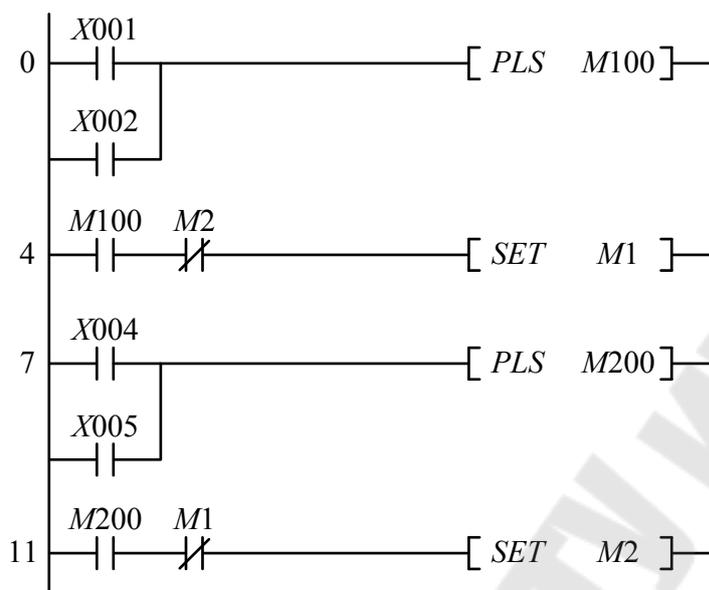
Функция		Обозн	Адрес	Примечание
В х о д ы	Кнопочный выключатель «СТОП»	S0	X0	Размыкающий контакт (при нажатии кнопки X0 = «0» и ворота останавливаются)
	Выключатель с ключом «Открыть ворота» (снаружи)	S1	X1	Замыкающие контакты
	Кнопка «Открыть ворота» (внутри)	S2	X2	
	Верхний концевой выключатель («Ворота открыты»)	S3	X3	Размыкающий контакт (X3 = «0», если ворота вверху и S3 активирован)
	Кнопка «Закрыть ворота» (внутри)	S4	X4	Замыкающие контакты
	Кнопка «Закрыть ворота» (снаружи)	S5	X5	
	Нижний концевой выключатель («Ворота закрыты»)	S6	X6	Размыкающий контакт (X6 = «0», если ворота внизу и S6 активирован)
	Фоторелейный барьер	S7	X7	Если распознается препятствие, X7 переходит в состояние «1»
В ы х о д ы	Сигнальный фонарь	H1	Y0	-
	Контактор для управления электродвигателем (левое вращение электродвигателя)	K1	Y1	Левое вращение = открыть ворота
	Контактор для управления электродвигателем (правое вращение электродвигателя)	K2	Y2	Правое вращение = закрыть ворота
Датчик времени	Задержка автоматического закрывания	-	T0	Время: 20 секунд

Программа

Управление шторными воротам с помощью кнопок.

Входные сигналы для управления воротами должны преобразовываться программой в две команды для приводного электродвигателя: «Открыть ворота» и «Заккрыть ворота» (рис. 6.2).

Релейная диаграмма



Список инструкций

```
0 LD X001
1 OR X002
2 PLS M100
4 LD M100
5 ANI M2
6 SET M1
7 LD X004
8 OR X005
9 PLS M200
11 LD M200
12 ANI M1
13 SET M2
```

Рис. 6.2. Пример управления шторными воротам с помощью кнопок

Так как речь идет о сигналах кнопок, т. е. сигналах, приложенных ко входам только короткое время, эти сигналы требуется сохранять в памяти. Для этого устанавливаются и сбрасываются два маркера, которые в программе поначалу заменяют собой выходы:

- M1: Открыть ворота;
- M2: Заккрыть ворота.

Сначала обрабатываются сигналы для открывания ворот: если включить выключатель с ключом S1 или нажать кнопку S2, вырабатывается импульс и маркеру M100 только на один программный цикл присваивается состояние «1». Таким образом, ворота невозможно заблокировать удержанием кнопки или ее заеданием.

Привод разрешается включать только в том случае, если он не вращается в противоположном направлении. По этой причине M1 устанавливается только в том случае, если M2 не установлен.

Блокировки направлений вращения должны быть дополнены блокировкой контактов самого контактора вне контроллера (см. электросхему).

Оценка сигналов кнопок S4 и S5 для закрывания ворот реализована аналогичным образом. Здесь опрос M1 на состояние «0» исключает возможность одновременной установки M1 и M2.

Автоматическое закрывание ворот через 20 секунд (рис. 6.3).

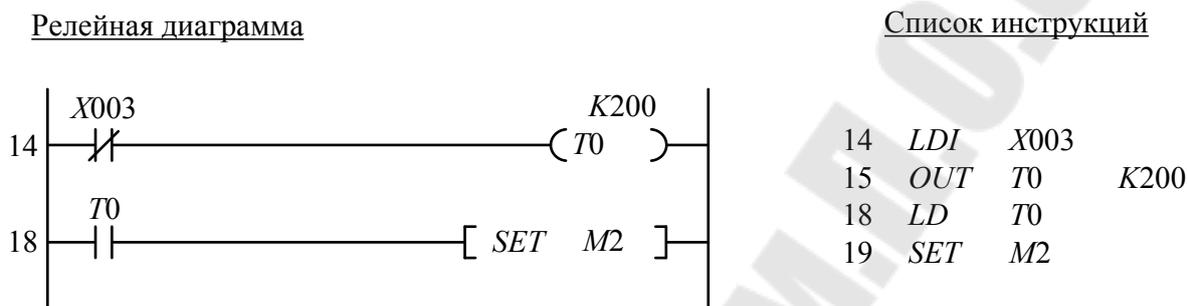


Рис. 6.3. Пример автоматического закрывания ворот через 20 секунд

Если ворота открыты, включается S3 и вход X3 выключается (по соображениям безопасности S3 имеет размыкающий контакт). Теперь начинает отсчитываться время задержки 20 с, реализованное на Т0 (K200 = 200 x 0,1 с = 20 с). По истечении этого времени устанавливается маркер M2, т. е. ворота закрываются.

Останов ворот с помощью кнопки «СТОП» (рис. 6.4).

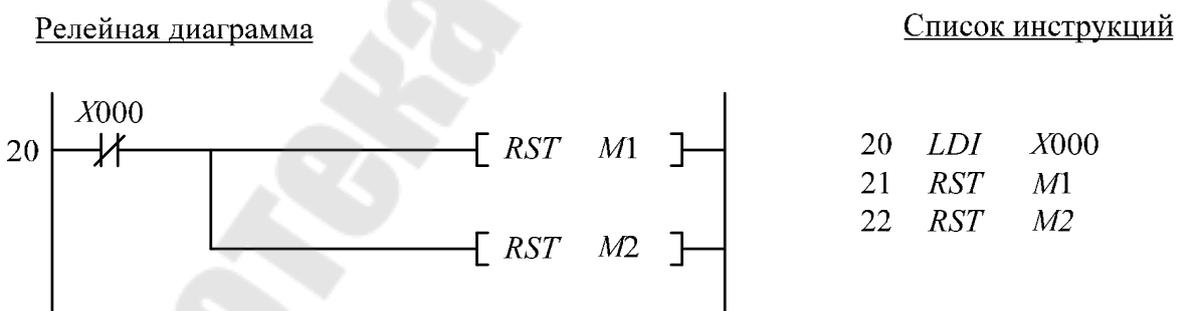
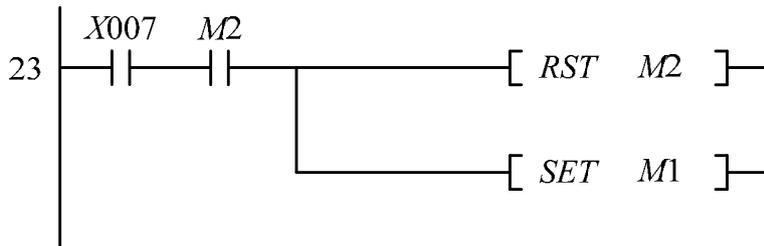


Рис. 6.4. Пример остановки ворот с помощью кнопки «СТОП»

При нажатии кнопки «СТОП» S0 оба маркера M1 и M2 сбрасываются и поэтому ворота останавливаются.

Распознавание препятствия с помощью фоторелейного барьера (рис. 6.5).

Релейная диаграмма



Список инструкций

```
23 LD X007
24 AND M2
25 RST M2
26 SET M1
```

Рис. 6.5. Пример распознавания препятствия с помощью фоторелейного барьера

Если во время закрывания фоторелейный барьер распознал препятствие, M2 сбрасывается и в результате этого процесс закрывания прерывается. Затем устанавливается M1 и ворота снова открываются.

Отключение электродвигателя концевыми выключателями (рис. 6.6).

Релейная диаграмма



Список инструкций

```
27 LDI X003
28 RST M1
29 LDI X006
30 RST M2
```

Рис. 6.6. Пример отключения электродвигателя концевыми выключателями

Открытые ворота активируют концевой выключатель S3 и выключают вход X3. В результате этого M1 сбрасывается и привод останавливается. Если ворота достигли нижнего положения, S6 включается, X6 выключается и в результате этого M2 сбрасывается и привод останавливается. По соображениям безопасности концевые выключатели имеют размыкающие контакты. Благодаря этому привод выключается (или предотвращается его включение) и в случае обрыва соединения между выключателем и входом.

Концевые выключатели должны останавливать привод и независимо от контроллера. Для этого они должны быть встроены в электропроводку (см. электросхему).

Управление электродвигателем (рис. 6.7).

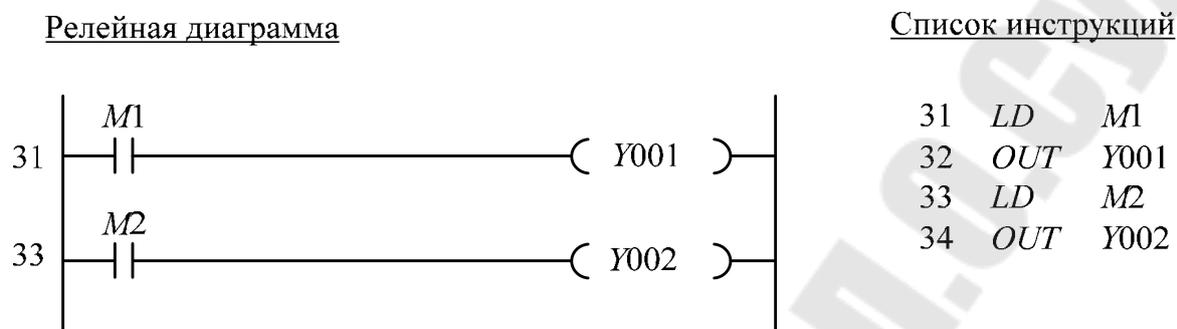


Рис. 6.7. Пример управления электродвигателем

В конце программы сигнальные состояния обоих маркеров M1 и M2 передаются выходам Y001 и Y002.

Сигнальный фонарь: «Ворота движутся» и «Ворота в неопределенном положении» (рис. 6.8).

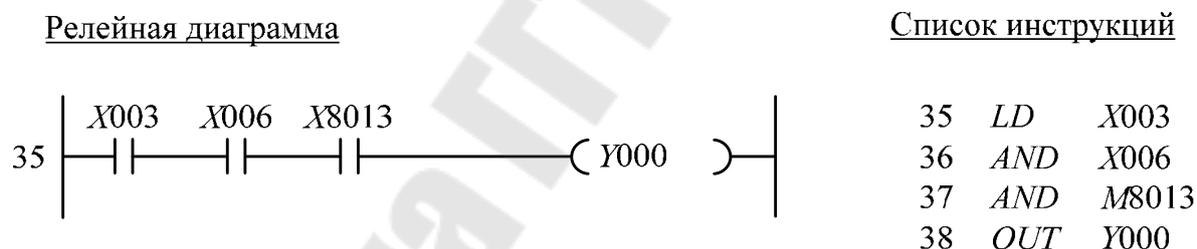


Рис. 6.8. Пример сигнала фонаря «Ворота движутся» и «Ворота в неопределенном положении»

Если ни один из двух концевых выключателей не активирован, то это означает, что ворота либо открываются, либо закрываются, либо остановлены в промежуточном положении. В этих случаях активируется мигающий сигнальный фонарь. В качестве такта мигания используется специальный маркер M8013, который автоматически устанавливается и сбрасывается с 1-секундным тактом.

Подключение контроллера

Для вышеописанной системы управления шторных ворот можно применить, например, контроллер FX1N-14MR (рис. 6.9).

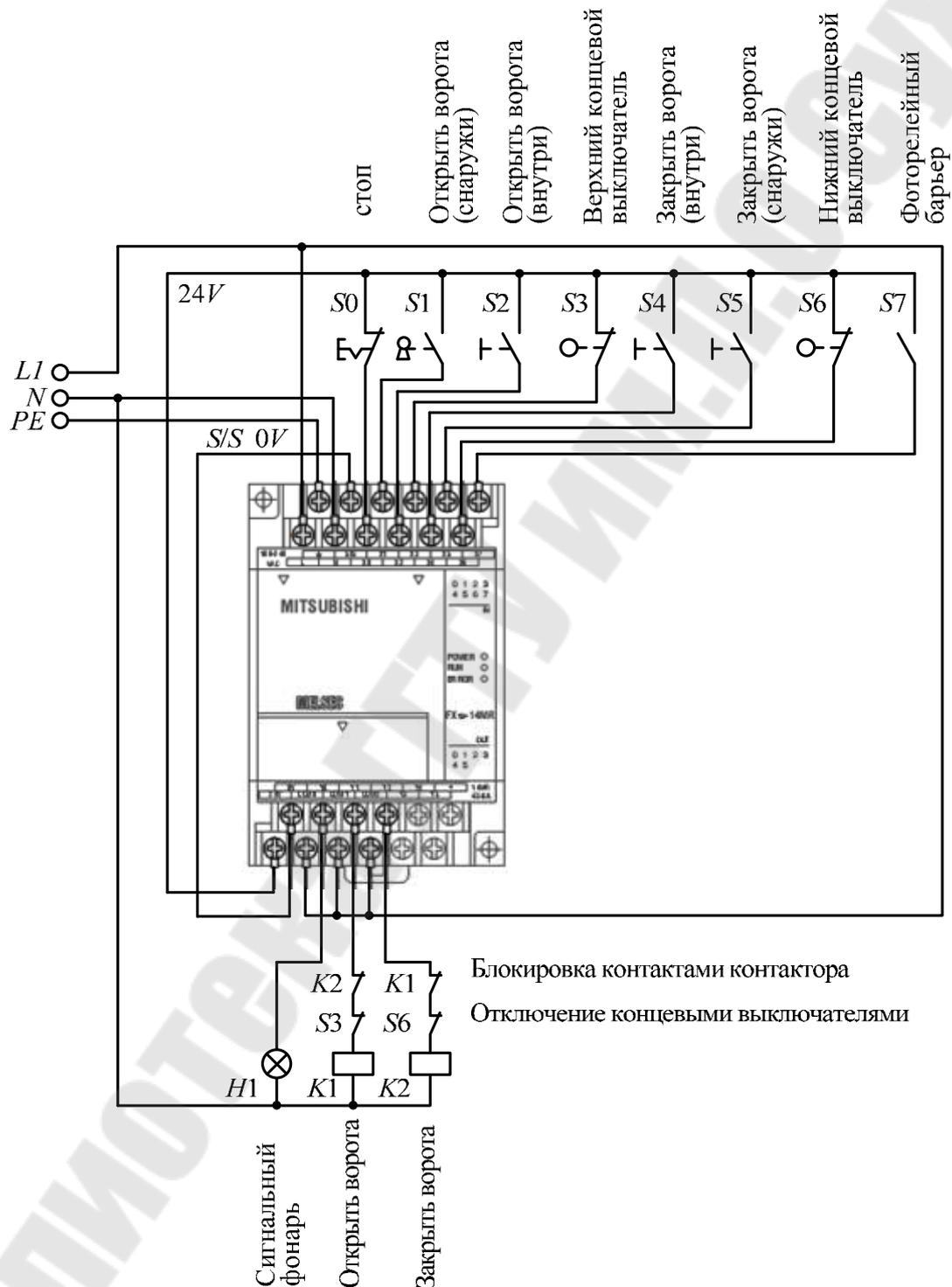


Рис. 6.9. Схема управления электроприводом шторных ворот на основе контроллера FX1N-14MR

Литература

1. Гребнев В.В. Микроконтроллеры семейства AVR фирмы Atmel. — М.: ИП РадиоСофт, 2002 — 176с.
2. Евстифеев А.В. Микроконтроллеры AVR семейства Mega. Руководство пользователя. — М.: Издательский дом «Додэка-XXI», 2007. — 592 с.
3. Ревич Ю.В. Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера. — М.: БХВ-Петербург, 2014. — 368 с.
4. Хартов В.Я. Микроконтроллеры AVR. Практикум для начинающих. 2-е издание. — М.: МГТУ им. Н. Э. Баумана, 2012. — 280 с.
5. Руководство по программированию ПЛК Mitsubishi. Mitsubishi Electric - руководство по использованию ПЛК серии Melsec FX, 2009. — 89 с.
6. Минаев И.Г., Самойленко В.В. Программируемые логические контроллеры. Практическое руководство для начинающего инженера. Учебное пособие. - Ставрополь: АГРУС, 2009. - 100 с.

Оглавление

Тема 1. Управление светодиодами	3
Тема 2. Запоминание последовательности сигналов и воспроизвести ее с соблюдением интервалов между сигналами.....	9
Тема 3. Программы с использованием таблицы данных	16
Тема 4. Управление шаговым двигателем	22
Тема 5. Управление охранной сигнализацией	27
Тема 6. Управление электроприводом шторных ворот	34
Литература	42

МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА В АВТОМАТИЗИРОВАННОМ ЭЛЕКТРОПРИВОДЕ

**Практикум
по одноименной дисциплине
для студентов специальности 1-53 01 05
«Автоматизированные электроприводы»
дневной формы обучения**

Составитель **Савельев** Вадим Алексеевич

Подписано к размещению в электронную библиотеку
ГГТУ им. П. О. Сухого в качестве электронного
учебно-методического документа 18.03.19.

Рег. № 48Е.
<http://www.gstu.by>