

Взаимодействие программных и аппаратных средств

Цель работы: изучение механизма преобразования программных переменных в физические сигналы и наоборот.

Задача работы: построение блок схем и алгоритмов обмена данными между регистрами процессора, программной памятью и портами внешних устройств.

Введение

Понимание, каким образом программная переменная принимает значение физического сигнала и как значение программной величины превращается в конкретный сигнал, может помочь разработчикам программного обеспечения систем управления при расчете задержек, решении задач оптимизации, обеспечении устойчивости, разработке интерфейсов и подключении к процессорам и контроллерам периферии: датчиков, исполнительных механизмов, и др.

В этой работе рассматриваются вопросы, связанные с прямым и обратным преобразованием “программная переменная – сигнал”: особенности построения каналов передачи данных, подключения периферии к шинам процессора, вопросы адресной приёма-передачи данных и работа гипотетического процессора при обмене данными с внешним устройством.

ОБЩИЕ СВЕДЕНИЯ

Формирование канала передачи данных

Рассмотрим канал передачи данных, который имеет всего два состояния, например, наличие или отсутствие напряжения. Передатчик должен устанавливать напряжение в линии в соответствии с передаваемыми данными, а приемник по измеряемому напряжению выделять данные линии. Простейший передатчик включает источник напряжения и идеальный ключ, который изменением собственного сопротивления (ноль или бесконечность) замыкает или размыкает провода. В качестве идеального приемника используется измеритель напряжения (мультиметр) с бесконечным входным сопротивлением.

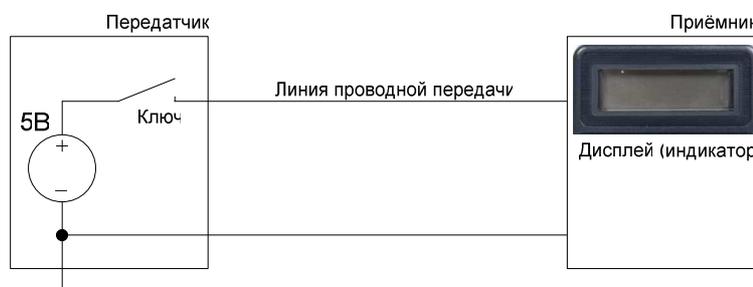


Рис. 1. Схема с неопределенным напряжением в линии передачи. В реальной схеме функцию ключа может выполнять транзистор.

В схеме Рис. 1 при замыкании ключа приёмнику передается 5В источника, однако, когда ключ разомкнут на входе приемника находится неопределенное напряжение.

Этой неопределенности нет в схеме Рис. 2, которая имеет два устойчивых состояния 5В или 0В. Сопротивление R добавлено в схему передатчика для ограничения тока через замкнутый ключ. Переходный процесс в линии пропорционален произведению ёмкости линии на выходное сопротивление передатчика, которое отличается при установлении и сбросе напряжения в линии. При установке напряжения в линии происходит заряд ёмкости через сопротивление R. Во время сброса напряжения ёмкость разряжается через сопротивление замкнутого ключа.

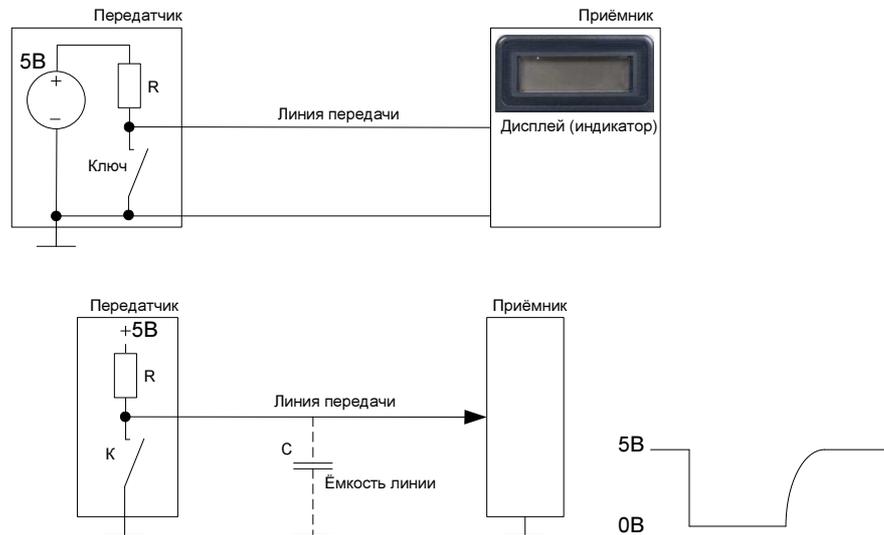


Рис. 2. Передача двух состояний 5В и 0В. Тождественные схемы. Постоянная времени ($R \cdot C$) перехода из 0В в 5В (заряда ёмкости) больше постоянной разряда линии через идеальный ключ с нулевым сопротивлением.

Для уменьшения времени установления напряжения резистор R заменён ключом K1 (Рис. 3), в схему добавлен инвертор И-НЕ, который обеспечивает работу ключей K1 и K2 в противофазе (один замкнут, другой - разомкнут).

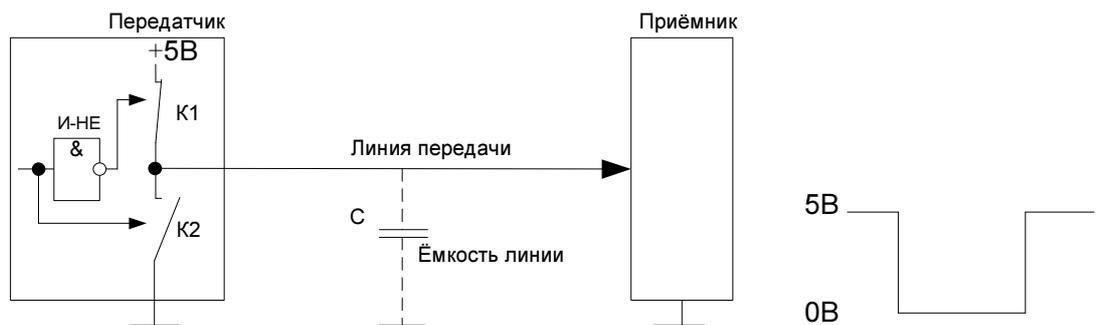


Рис. 3. Ключи K1 и K2 минимизируют время переходных процессов. Инвертор И-Не обеспечивает работу ключей в противофазе.

Проблемы передачи данных

Поток последовательных данных состоит из нулей и единиц. Пусть, состояниям 0В и 5В соответствуют логический ноль и единица соответственно (Рис. 4). Необходимо, чтобы все данные передатчика были приняты приемником без потерь. Для этого

1. *Передатчик и приемник должны работать с одинаковым периодом (на одинаковой частоте).*

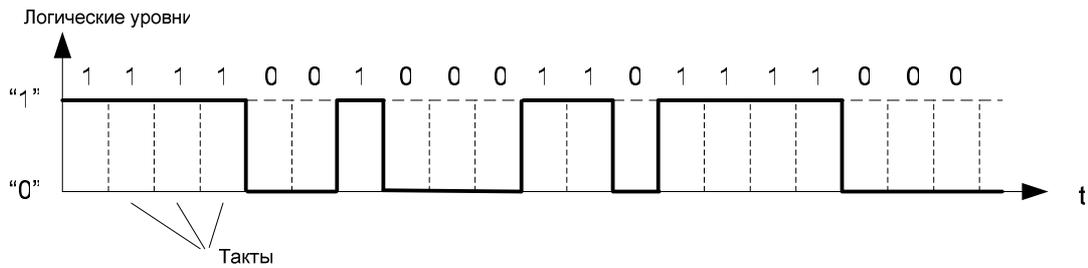


Рис. 4. Тактируемая последовательная передача данных.

Предположим, линия длительное время не меняет состояние и находится под напряжением 5В. Как определить – “Передает передатчик данные 111111111... или нет?” Для решения этого вопроса протокол передачи данных необходимо дополнить следующими положениями.

2. При отсутствии передачи линия находится под напряжением (в состоянии “1”).
3. Передача начинается переходом из 1 в ноль.
4. Первый такт после перехода не содержит данных.

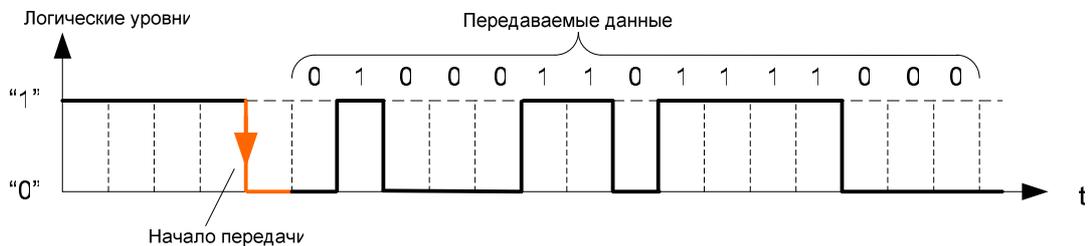


Рис. 5. Ввод стартового бита.

Чем дольше последовательность данных (длина пакета), тем выше производительность линии. Но, с другой стороны, увеличение длины пакета ужесточает требования к синхронизации работы передатчика и приемника и увеличивает чувствительность к помехам в линии, которые могут изменить передаваемую последовательность.

Для обнаружения ложных данных вводят в последовательность контрольную сумму или другие средства, защищающие данные от потерь. В общем случае, длина пакета должна устанавливаться с учетом качества сетевого оборудования. Пример передачи данных байтами показан на Рис. 6. Производительность этой линии составляет 66% (как $100\% * 8$ бит данных / 12 бит период).

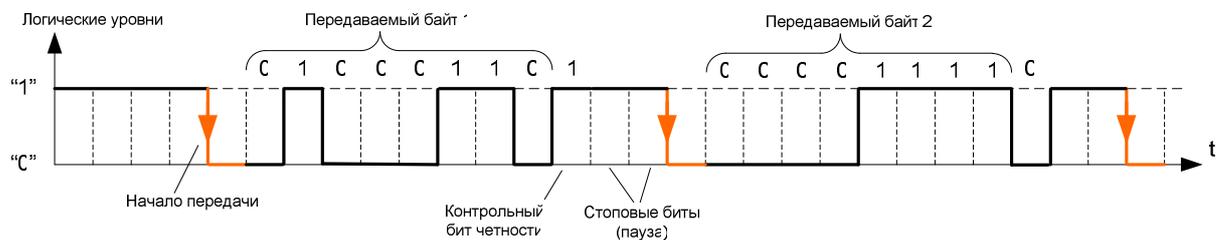


Рис. 6. Побайтовая передача данных по протоколу RS-232.

Для успешной передачи данных передатчик и приёмник должны иметь одинаковые настройки: частоту тактирования, наличие (отсутствие) контрольного бита, одинаковое количество стоповых бит и др.

Увеличение числа приемников и передатчиков

Подключение к линии дополнительных приемников (Рис. 7) не приводит к сбою передачи данных если суммарная нагрузка на передатчик не превышает предельную.

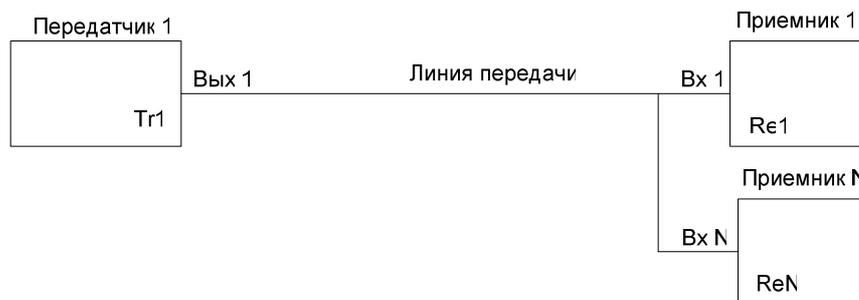


Рис. 7. Подключение к линии передачи нескольких приемников.

Объединение выходов передатчиков может вызвать сквозной ток, как показано на Рис. 8. Такой ток возникает когда передатчики одновременно выдают в линию инверсные уровни (низкий и высокий).

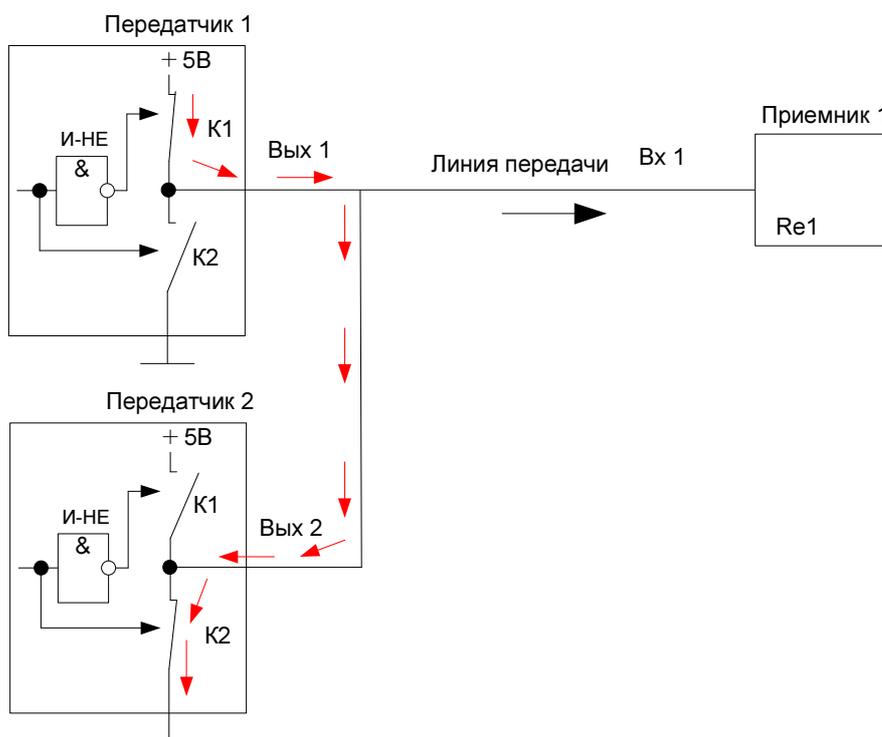


Рис. 8. Конфликт передатчиков с выходами +5В/0В. Красными стрелками показан ток короткого замыкания.

Таблица 1. Состояния линии передачи с двумя передатчиками и приёмником.

Вых 1	Вых 2	Вх 1	Комментарий
0	0	0	
0	1	н.о.	Короткое замыкание источника передатчика 2 на землю передатчика 1
1	0	н.о.	Короткое замыкание источника передатчика 1 на землю передатчика 2
1	1	1	

Передатчики с открытым коллектором (с единственным нижним ключом К2) могут подключаться к линии передачи группами. При использовании передатчиков с открытым

коллектором линия должна “подтягиваться” к источнику питания через резистор, как показано на Рис. 9.

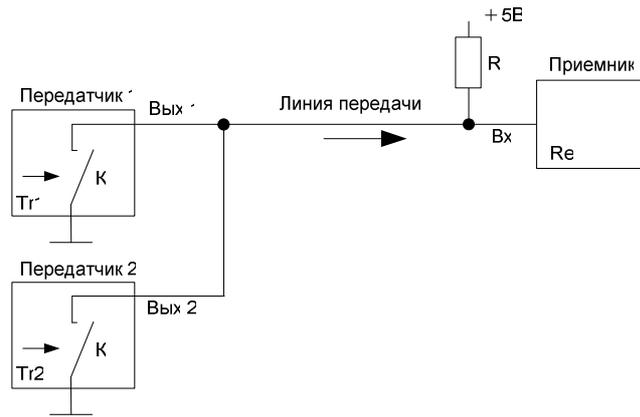


Рис. 9. Подключение нескольких передатчиков с открытым коллектором к общей линии.

Буфер и Регистр

Рассмотрим принцип действия буфера и регистра, используемых для подключения приёмников и передатчиков к шине данных процессора.

Буфер – электронное устройство с параллельными сигнальными входами и выходами. Выходы подключаются к соответствующим входам только на время действия управляющего сигнала (Рис. 10). Остальное время открытые коллекторы буфера удерживают его в отключенном состоянии от выходных линий.

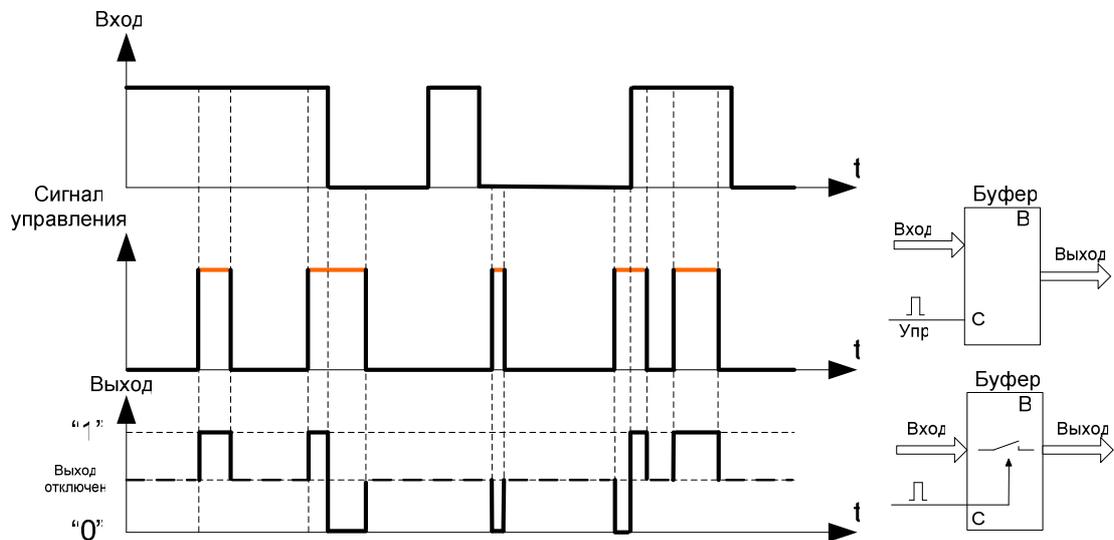


Рис. 10. Временные диаграммы работы буфера.

Регистр – устройство с памятью (Рис. 11). В момент появления управляющего сигнала регистр устанавливает на выходах значения входных сигналов и удерживает выходные сигналы без изменения до момента появления следующего управляющего сигнала, который снова устанавливает на выходах значения входных сигналов.

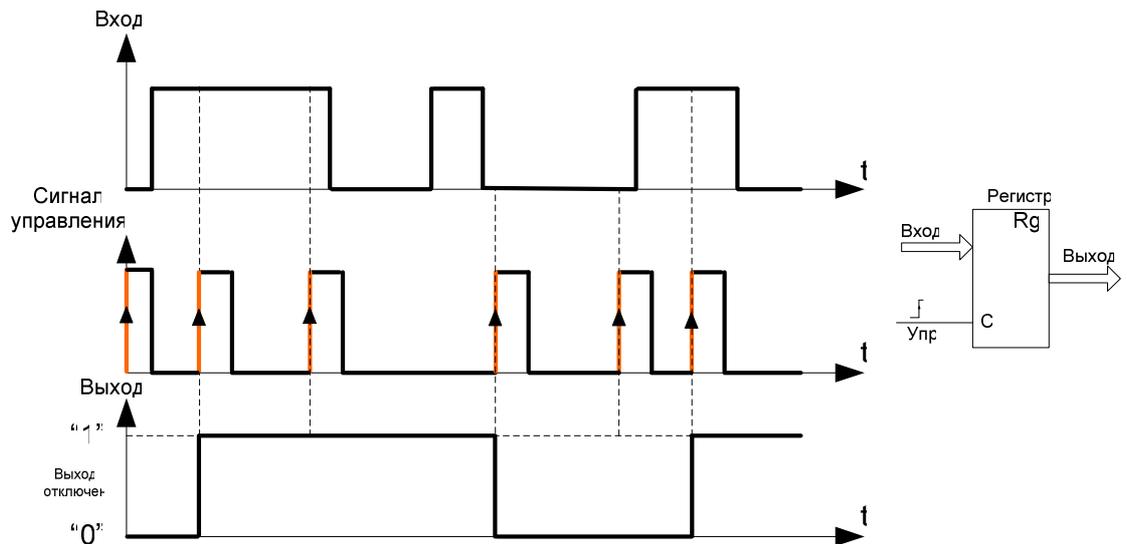


Рис. 11. Зависимость выхода от входа регистра.

Дешифратор адреса

Только два устройства (приемник и передатчик) могут участвовать в обмене данными, даже если к линии подключено несколько передатчиков и/или приемников. Для обращения к устройствам, участвующим в обмене данными, каждому устройству присваивается уникальный адрес (имя устройства). В системах параллельной передачи данных для обращения к устройствам используется адресная шина данных. В состав устройств должен входить дешифратор адреса, который следит за состоянием адресной шины и формирует сигнал на выходе при появлении требуемого адреса. Логика работы дешифратора адреса для четырехразрядной шины и одного устройства с адресом 1011 показана на Рис. 12.

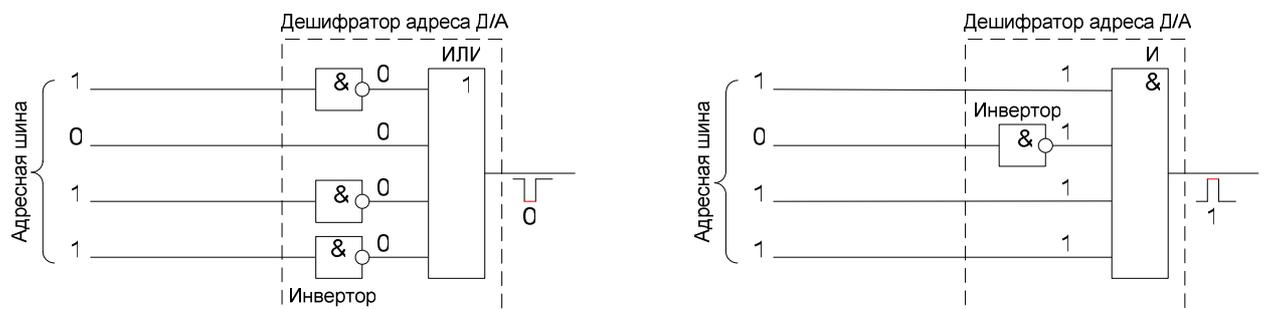


Рис. 12. Примеры логики дешифратора адреса для одного устройства с адресом 1011.

Дешифратор адреса с несколькими выходами (Рис. 13) активирует только один выход адрес которого находится на входе дешифратора.

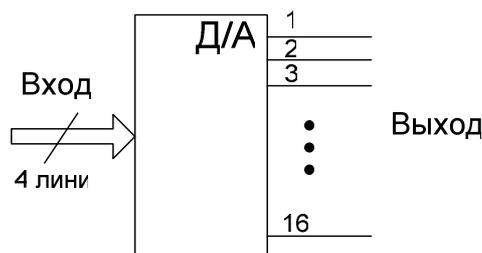


Рис. 13. Дешифратор адреса для 16 устройств.

Циклы записи, чтения и подключение портов

Чтение и запись данных рассматриваются относительно устройства, управляющего процессом обмена данными. Если это процессор, то “Запись” означает передачу от процессора, а “Чтение” – приём процессором.

Для получения данных от устройства процессор выставляет адрес устройства на шину адреса, затем - управляющий сигнал "Чтение" RD и, через некоторое время, считывает состояние шины данных (Рис. 14). Внешнее устройство при помощи дешифратора адреса должно определить обращение к нему и по запросу данных RD процессором, подключить собственные данные через буфер к шине данных и удерживать их на шине не дольше, чем процессор начнет новый цикл.

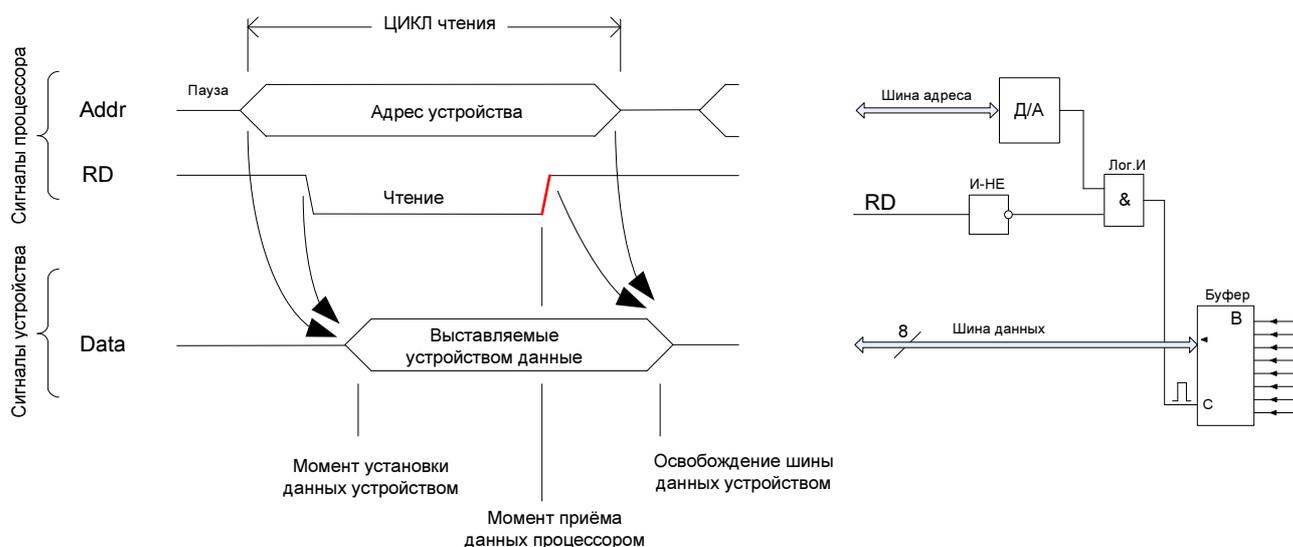


Рис. 14. Цикл чтения.

Цикл записи процессор также начинает с выставления адреса устройства на шину адреса, затем выставляет данные и сообщает об этом управляющим сигналом "Запись" WR (Рис. 15). Внешнее устройство, обнаружив собственный адрес на адресной шине и получив от процессора сигнал записи данных WR, записывает состояние шины данных в регистр.

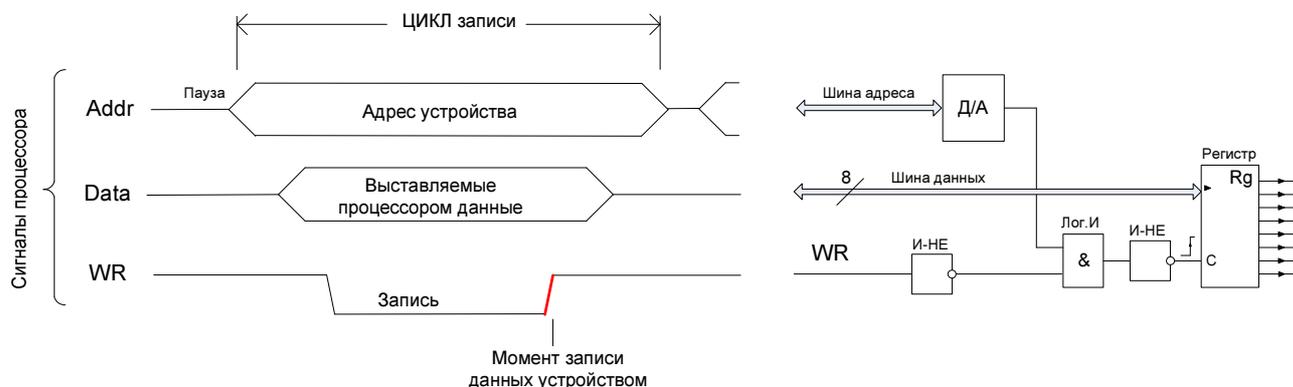


Рис. 15. Цикл записи.

Внешняя память

Если выход регистра внешнего устройства подключить ко входу буфера, то получится ячейка памяти, которая может принимать данные, хранить и выставлять данные на шину в цикле чтения. Буфер и регистр могут подключаться к общему или отдельным выходам дешифратора адреса. В первом варианте (показан на Рис. 16) запись и чтение данных будут происходить по одному адресу, во втором варианте запись выполняется по одному адресу – а чтение записанных данных – по-другому.

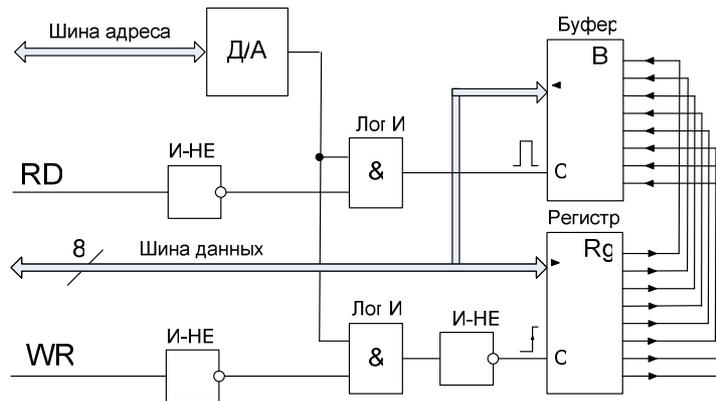


Рис. 16. Ячейка внешней памяти.

Структура гипотетического процессора

Гипотетический процессор (Рис. 17) включает следующие основные части.

- Арифметико-Логическое Устройство (АЛУ). Выполняет арифметические и логические преобразования.
- Аккумулятор. Регистр, сохраняет результаты АЛУ, обменивается данными с устройствами подключенными к шине данных.
- Дешифратор команд. Анализирует инструкции. Управляет счетчиком команд и запускает работу схемы управления.
- Схема управления. Вырабатывает необходимую последовательность управляющих сигналов, обеспечивающих выполнение операций.
- Регистры процессора. Хранят адреса и данные.
- Сигналы управления “Чтение” RD, “Запись” WR и др. Служат для управления передачей данных.

Для работы процессору необходима память программ и данных, которая подключается к процессору через адресную шину, шину данных и сигналы шины управления: “Чтение памяти” и “Запись в память”. Программа, которую выполняет процессор, находится в памяти программ.

Процессор имеет сигналы “Чтение устройства ввода/вывода” и “Запись в устройство ввода/вывода”, которые используются при обращении к внешним устройствам - портам.

Адресные пространства программной памяти и портов могут перекрываться.

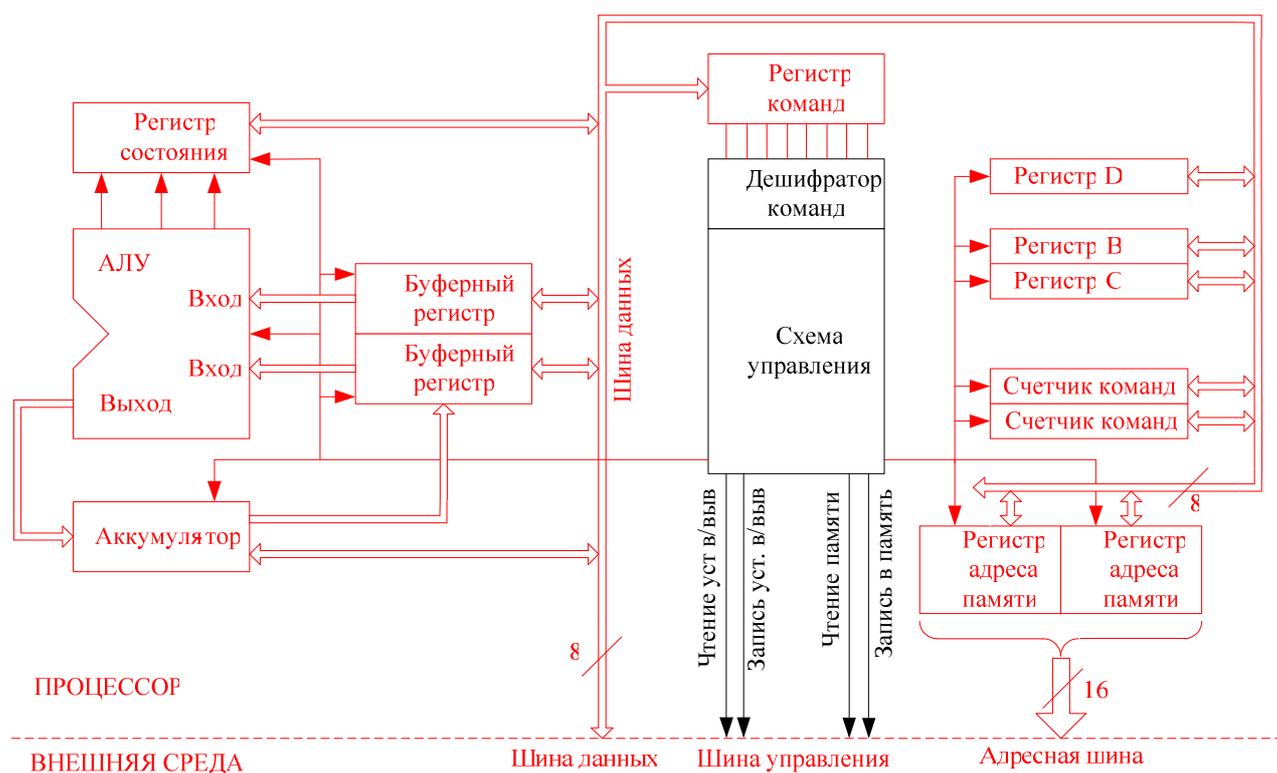


Рис. 17. Структура гипотетического процессора [0].

Принцип организации управления исполнением программ гипотетическим процессором содержит следующие положения.

- В коде команды содержится полная информация о том, что необходимо выполнить процессору и где располагается следующая команда.
- В начале работы счетчику команд процессора присваивается значение адреса первой команды программы.
- Выполнение команды включает цикл выборки команды и цикл(ы) выполнения.
- В цикле выборки содержимое счетчика команд загружается в регистр адреса памяти, затем команда из внешней памяти передается через шину данных в регистр команд процессора.
- В цикле выполнения дешифратор команд определяет размер команды (занимаемую командой область памяти в байтах), схема управления устанавливает значение адреса следующей команды в счетчике команд путём увеличения счетчика на размер выполняемой команды.
- В многобайтной команде, кроме кода команды содержатся адреса операндов и/или значения операндов. При выполнении многобайтной команды регистр адреса памяти последовательно увеличивается на единицу, указывая на адрес используемого байта команды, а схема управления подключает соответствующие ячейки памяти к регистрам процессора.
- Передача команд и данных по шине данных осуществляется от буфера-передатчика к регистру-приемнику, как было рассмотрено на Рис. 14, Рис. 15, Рис. 16.

Команды ввода/вывода (обращения к портам)

В список команд процессора входят команды обращения к портам, например, для процессора x86 это команды OUT и IN. При обмене данными с внешним портом участвуют регистры общего назначения DX и AX. В DX записывается адрес порта, регистр AX используется для приёма-передачи данных.

Пример записи в порт по адресу PORT_NUMBER данных DATA:

```
MOV DX, PORT_NUMBER; // Запись адреса порта в регистр DX
```

```
MOV AL, DATA; // Запись байта данных в младший байт аккумулятора
```

```
OUT DX, AL; // Запись в порт. На адресную шину выдается содержимое регистра DX, на шину данных – содержимое регистра AL. Посылка данных синхронизируется сигналом управления “Запись в устройство ввода/вывода”.
```

Пример чтения данных из порта с адресом PORT_NUMBER с сохранением их в регистре AL.

```
MOV DX, PORT_NUMBER; // Запись адреса порта в регистр DX
```

```
IN AL, DX; // Чтение порта. На адресную шину выдается содержимое регистра DX, затем выставляется сигнал управления “Чтение устройства ввода/вывода”. Состояние шины данных записывается в регистр AL в момент окончания сигнала управления.
```

- Примечание. 1. При отсутствии устройства или данных устройства в момент окончания сигнала “Чтение устройства ввода/вывода” процессор записывает в AL состояние шины данных. Существуют различные схемы и алгоритмы повышения надежности приема/передачи, которые здесь не рассматриваются.
2. На языках высокого уровня для обмена данными с портами существуют собственные команды или используются вставки ассемблерного кода.

Пример пошагового обмена данными между процессором и портами внешнего устройства

Взаимодействие программных и аппаратных средств показано на примере выполнения программы гипотетическим процессором, который обменивается данными с внешним устройством (Рис. 18).

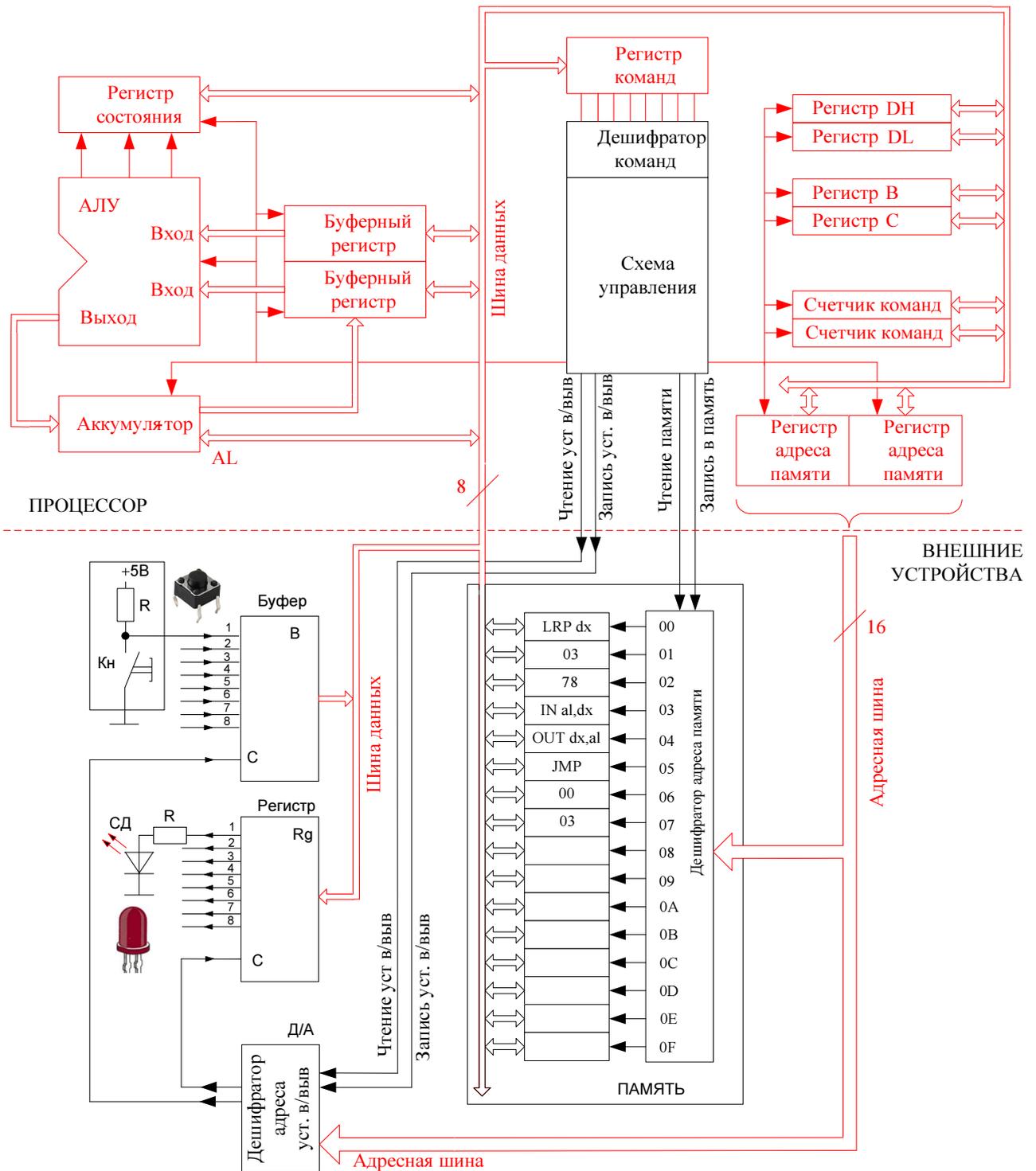


Рис. 18. Процессор с внешней памятью и портами.

Программная переменная должна находиться в аккумуляторе процессора. Внешнему устройству присвоен адрес 378h, который используется как при чтении так и при записи. Сигнал младшего разряда на буфере внешнего устройства зависит от состояния кнопки. К младшему разряду выхода регистра подключен светодиод.

Светодиод должен отслеживать состояние кнопки: включаться при отжатой кнопке и выключаться при замкнутых контактах кнопки, а программная переменная отслеживать состояние светодиода. Для этого процессору в цикле необходимо считывать в аккумулятор сигналы внешнего устройства со входа буфера, а затем данные аккумулятора записывать в регистр внешнего устройства.

Программа, расположенная с нулевого адреса в подключенной к процессору внешней памяти, содержит следующие команды.

- ЗАГРУЗКА РЕГИСТРОВОЙ ПАРЫ ПРЯМАЯ (LRP dx). Записывает в регистр DX процессора данные команды.
- СЧИТЫВАНИЕ ПОРТА (IN al,dx). Передает данные внешнего устройства в младший регистра аккумулятора. Адрес внешнего устройства берется из регистра DX процессора.
- ЗАПИСЬ В ПОРТ (OUT dx,al) передает данные младшего байта аккумулятора внешнему устройству. Адрес внешнего устройства находится в регистре DX процессора.
- ПЕРЕХОД (JMP). Переход к следующей команде по указанному адресу.

В начале работы счетчику команд присваивается значение начального адреса (точки входа) программы 0000.

Трехбайтная команда ЗАГРУЗКА РЕГИСТРОВОЙ ПАРЫ ПРЯМАЯ (LRP dx) включает один цикл выборки команды и два цикла выполнения. В цикле выборки содержимое счетчика команд загружается в регистр адреса памяти значение которого появляется на адресной шине и декодируется дешифратором адреса памяти, который указывает на нулевую ячейку памяти. При получении сигнала “Чтение памяти” устройство внешней памяти выставляет на шину данных содержимое нулевой ячейки – первую команду, которая записывается схемой управления в регистр команд процессора. В цикле выполнения, при декодировании команды, устройство управления увеличивает содержимое счетчика команд на три – величину занимаемой командой области памяти в байтах. Теперь счетчик указывает на адрес следующей команды 0003. Для обращения ко второму байту команды схема управления процессора увеличивает регистр адреса памяти на единицу. Адрес 0001 подается на адресную шину. Получив адрес и управляющий сигнал процессора устройство внешней памяти подключает содержимое ячейки 0001 (число 03h) к шине данных. Схема управления записывает число 03h в старший байт DH регистровой пары общего назначения DX. Вторая половина фазы выполнения начинается с увеличения регистра адреса памяти на единицу до 0002, в этой ячейке памяти содержится третий байт команды (число 78h), который как и два предыдущих попадает на шину данных, значение третьего байта записывается в младший байт DL пары DX процессора. На этом заканчивается выполнение команды LRP dx, в результате адрес порта внешнего устройства находится в регистре DX процессора.

При выполнении следующей команды IN al,dx в цикле выборки схема управления процессора загружает содержимое счетчика команд в регистр адреса памяти и выставляет сигнал “Чтение памяти”, команда из третьей ячейки памяти записывается в регистр команд. В цикле выполнения команды схема управления увеличивает содержимое счетчика команд на один байт - величину размера команды, счетчик указывает на адрес следующей команды 0004. Затем байты регистровой пары DX последовательно записываются в регистры адреса памяти, таким образом адрес порта внешнего устройства появляется на адресной шине. Дешифраторы и внешней памяти и внешнего устройства указывают на ячейку 378h, но, поскольку схема управления процессора в соответствии с кодом команды выдает сигнал “Чтение уст. в/выв” а не “Чтение памяти”, то только буфер внешнего устройства с адресом 378h подключается к шине данных и сигналы устройства, включая состояние кнопки, через буфер поступают на шину данных. Схема управления процессором записывает содержимое шины данных в младшем регистре аккумулятора.

После считывания из памяти команды OUT dx,al содержимое счетчика команд увеличивается на 1. Его содержимое теперь 0005 – адрес следующей команды JMP. Содержимое регистровой пары DX (0378h) пересылается в регистр адреса памяти процессора. Содержимое младшего байта аккумулятора AL помещается на шину данных. При поступлении сигнала “Запись уст. в/выв” эти данные с шины записываются в регистр внешнего устройства к которому подключен светодиод.

По команде ПЕРЕХОД (JMP) в счетчик команд из второго и третьего байта команды JMP загружается новое значение 0003 - адрес первой команды следующего цикла. Далее выполняется команда IN al,dx и другие команды. Описанный процесс повторяется до тех пор, пока не произойдет останов процессора.

Физический сигнал принимает значение программной переменной в результате выполнения процессором команды “Запись в порт”, которая сопровождается передачей данных по адресной шине, приема-передачей данных по шине данных и формированием соответствующего сигнала шины управления.

“Чтение порта” позволяет присвоить программной переменной значение физического сигнала. Здесь также участвует передача данных по адресной шине и шине данных с формированием сигнала управления – чтения устройства ввода/вывода.

Аналогичным образом происходят преобразование переменной в контроллерах или в мини-компьютерах, в которых внешняя память и/или порты объединены с процессором на одном кристалле или на общей плате.

ВАРИАНТЫ ДЛЯ САМОКОНТРОЛЯ

Задание 1. Разработайте на базе 8-разрядного гипотетического процессора устройство управления 17-ю светодиодами. Опишите алгоритм его работы.

Задание 2. Разработайте на базе гипотетического процессора устройство считывания и отображения (при помощи светодиодов) состояний 9 ключей (кнопок).

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Опишите механизм преобразования программных переменных в физические сигналы.
2. Какое устройство управляет циклом чтения и записи данных?
3. Назначение буфера и регистра?
4. Для чего необходим дешифратор адреса? Нужен ли он для передачи данных от одного передатчика одному приёмнику?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Введение в микропроцессорную технику Автор: Ч. Гилмор Издательство: Мир Год: 1984 Страниц: 314. Язык: русский.
2. Dr. Bob Davidov. Компьютерные технологии управления в технических системах <http://portalnp.ru/author/bobdavidov>.