Имитатор электронных ключей iButton

Внимание: данная статья предназначена исключительно для ознакомления с принципами работы с ключами серии iButton фирмы Dallas Semiconductor. Автор и редакция не несут ответственности за возможное незаконное использование всей или части приведенной в статье информации, схемотехнических или программных решений. Оригинальная разработка предназначалась для обеспечения возможности быстрого проникновения в помещения в случае возникновения экстремальной ситуации.

Имитатор электронных ключей фирмы Dallas Semiconductor типа DS1990A, который предлагается вашему вниманию, способен запоминать номера до 30 разных ключей, а затем имитировать любой ключ. Номера можно вводить двумя способами: непосредственным считыванием оригинала или ручным вводом в компьютер с последующей записью в имитатор по RS-232. В компьютере может быть создана база ключей, в которой указан тип, серийный номер и текстовое описание.

Touch Memory типа DS1990A представляет собой пассивное устройство (без внутреннего источника питания), которое содержит записанное с помощью лазера ПЗУ. ПЗУ содержит уникальный серийный номер. Для считывания данных с DS1990A используется 1-проводная шина фирмы DALLAS. DS1990A является подчиненным устройством, а мастером является обычно микропроцессор. Питание DS1990A во время обмена данными производится от 1-проводной шины. Эквивалентная схема интерфейсной части DS1990A показана на рис.1*а*.

Аналогичную интерфейсную часть имеет и мастер (рис. 1*б*), отличие состоит в наличии подтягивающего резистора. В состоянии ожидания 1-проводная шина имеет высокий логический уровень. Последовательность доступа к DS1990A по 1-проводной шине следующая:

- Инициализация.
- Команда чтения ПЗУ.
- Чтение данных.

Все пересылки по 1-проводной шине начинаются с инициализации. Инициализация производится в следующей последовательности (рис. 2):

- Мастер посылает импульс сброса (reset pulse) — сигнал низкого уровня длительностью не менее 480 мкс.
- За импульсом сброса следует ответ подчиненного устройства (presence pulse) — сигнал низкого уровня длительностью 60–240 мкс, который генерируется через 15–60 мкс после завершения импульса сброса.

Ответ подчиненного устройства дает мастеру понять, что на шине присутствует устройство DS1990A и оно готово к обмену. После того как мастер обнаружил ответ, он может передавать команду чтения ПЗУ. Команда чтения ПЗУ имеет код 33H. Передача данных ведется путем формирования специальных временных интервалов (time slots). Каждый временной интервал служит для передачи одного бита. Первым передается младший бит. Интервал начинается импульсом низкого уровня, длительность которого лежит в пределах 1–15 мкс. Поскольку переход из единицы в ноль менее чувствителен к емкости шины (он формируется открытым транзистором, в то время как переход из ноля в единицу формируется подтягивающим резистором), именно этот переход DS1990A использует для синхронизации с мастером. В DS1990A запускается схема временной задержки, которая определяет момент считывания данных. Номинальное значение задержки равно 30 мкс, однако оно может колебаться в пределах 15–60 мкс. За импульсом низкого уровня следует передаваемый бит. Он должен удерживаться на шине 60–120 мкс от начала интервала. Временной интервал завершается переводом шины в состояние высокого уровня на время не менее 1 мкс. Это необходимо для зарядки внутреннего конденсатора, который обеспечивает питание DS1990A. Аналогичным образом формируются временные интервалы для всех передаваемых битов (рис. 3).

Приняв команду чтения ПЗУ, DS1990A передает 8-битный код типа устройства (для DS1990A это 01H), 48-битный серийный номер и 8-битную контрольную сумму. Временные интервалы для принимаемых битов тоже формирует мастер. Интервал начинается импульсом низкого уровня длительностью 1–15 мкс. Затем мастер должен освободить шину, чтобы дать возможность DS1990A вывести бит данных. По переходу из единицы в ноль DS1990A выводит на шину бит данных и запускает схему временной задержки, которая определяет, как долго бит данных будет присутствовать на шине. Это время лежит в пределах 15–60 мкс. Для того чтобы данные на шине гарантированно установились, требуется некоторое время. Поэтому момент считывания данных мастером должен отстоять чуть меньше, чем на 15 мкс от начала временного интервала (рис. 4).

Правильность принятых данных проверяется с помощью контрольной суммы. Если подсчитать контрольную сумму всех восьми считанных байтов (включая байт считанной контрольной суммы), то в случае отсутствия ошибок должен получиться ноль. Подпрограмма вычисления контрольной суммы приведена ниже:

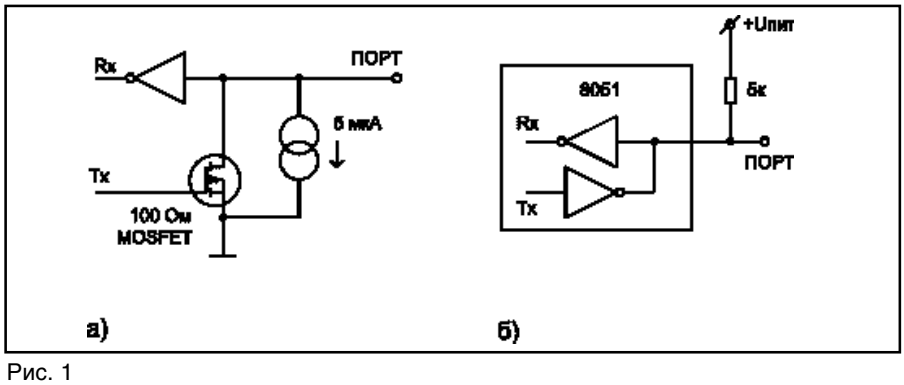

Рис. 1

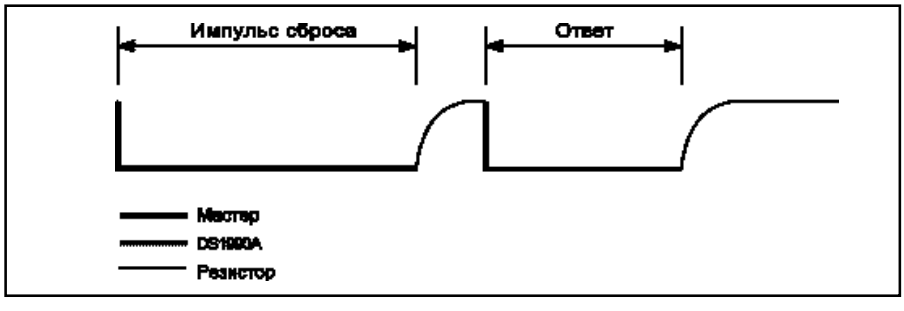

Рис. 2

```
DOW_CRC:  PUSH ACC          ;Save the Accumulator.
          PUSH B            ;Save the B register.
          PUSH ACC          ;Save bits to be shifted.
          MOV  B,#8         ;Set to shift eight bits.
CRC_LOOP: XRL  A,TEMP       ;Calculate DQIN xor CRCT0.
          RRC  A            ;Move it to the carry.
          MOV  A,TEMP       ;Get the last CRC value.
          JNC  ZERO         ;Skip if DQIN xor CRCT0 = 0.
          XRL  A,#18H       ;Update the CRC value.
ZERO:     RRC  A            ;Position the new CRC.
          MOV  TEMP,A       ;Store the new CRC.
          POP  ACC          ;Get the remaining bits.
          RR   A            ;Position next bit in LSB.
          PUSH ACC          ;Save the remaining bits.
          DJNZ B,CRC_LOOP   ;Repeat for eight bits.
          POP  ACC          ;Clean up the stack.
          POP  B            ;Restore the B register.
          POP  ACC          ;Restore the Accumulator.
          RET               ;Return.
```

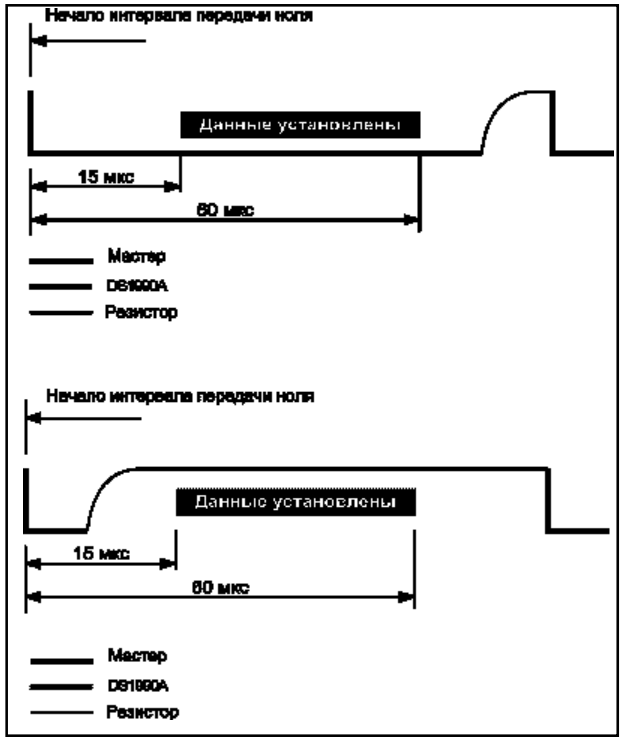

Рис. 3

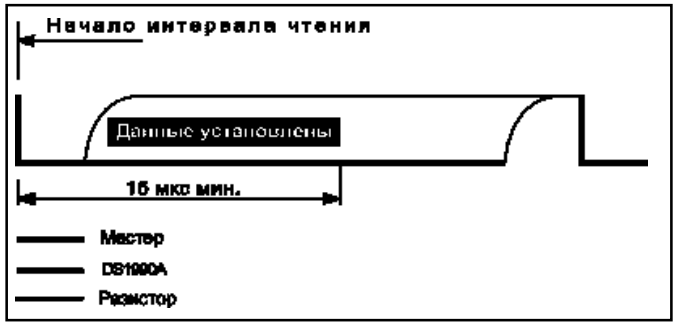

Рис. 4

рическим стиранием объемом 2 Кбайта. Микроконтроллер работает на тактовой частоте 12 МГц. Программа содержит довольно много фрагментов, критичных к времени выполнения, поэтому изменять тактовую частоту без соответствующей корректировки программы нельзя.

Для отображения номера текущего ключа в устройстве применен сдвоенный 7-сегментный светодиодный индикатор с общим анодом HG1. Для уменьшения количества элементов схемы применена динамическая индикация, реализованная программно. Катоды индикатора через токоограничивающие резисторы подключены к порту P1. Аноды управляются ключами VT1 и VT2. Ключами же управляют линии сканирования T0 и T1. Соединение катодов двух разрядов индикатора произведено в произвольном порядке, потому что так удобнее для топологии печатной платы. В связи с этим, для каждого разряда индикатора в программе использована отдельная таблица знакогенератора.

Те же линии порта P1, к которым подключены катоды индикаторов, используются и для сканирования кнопок SB1 и SB2. Линия возврата — порт P1.7. Диоды VD2 и VD3 предотвращают замыкание линий P1.0 и P1.1 (и нарушение работы индикации) при одновременном нажатии двух кнопок.

Для экономии портов контроллера микросхема энергонезависимой памяти, в которой сохраняются номера ключей, подключена к линиям сканирования индикатора T0 и T1. Процесс сканирования представляет собой чередующиеся условия «Старт» и «Стоп» шины I2C и на работу микросхемы памяти влияния не оказывает. Во время циклов записи/чтения по шине I2C процесс сканирования индикаторов приостанавливается. При этом индикаторы гасятся путем вывода в порт P1 всех единиц.

В качестве 1-проводного порта использована ножка INT0 микроконтроллера. Элементы R1, VD1 имеют защитную функцию. К выводам DQ и GND можно подключить параллельно touch probe DS9092 и touch port DS9092R. Однако их стоимость, пожалуй, больше стоимости всего устройства. Поэтому можно обойтись более простыми контактными устройствами.

Преобразователь уровней для порта RS-232 выполнен на транзисторах VT3 и VT4. Отрицательное напряжение питания поступает с порта компьютера через контакт RTS.

Питание устройства осуществляется от батареи из трех элементов (например, размера AAA) с суммарным напряжением 4,5 В. Работоспособность устройства сохраняется при снижении напряжения питания до 3 В.

Никакой настройки устройство не требует. Достаточно только, чтобы монтаж был выполнен без ошибок и из исправных деталей. Микросхему U2 можно заменить 24C08, 24C16, 24LC04, 24LC08, 24LC16. Важно только, чтобы она позволяла производить запись при снижении напряжения питания до 3 В (микросхемы некоторых производителей запрещают запись при напряжении менее 4,5 В). Вместо

Эта подпрограмма должна вызываться для каждого байта, участвующего в подсчете контрольной суммы. Байт должен быть помещен в A. Контрольная сумма получается в ячейке памяти TEMP (перед началом подсчета контрольной суммы эта ячейка должна быть обнулена).

Принципиальная схема устройства приведена на рис. 5. Основой является микроконтроллер фирмы Atmel AT89C2051. Этот микроконтроллер имеет встроенное ПЗУ программ с элект-

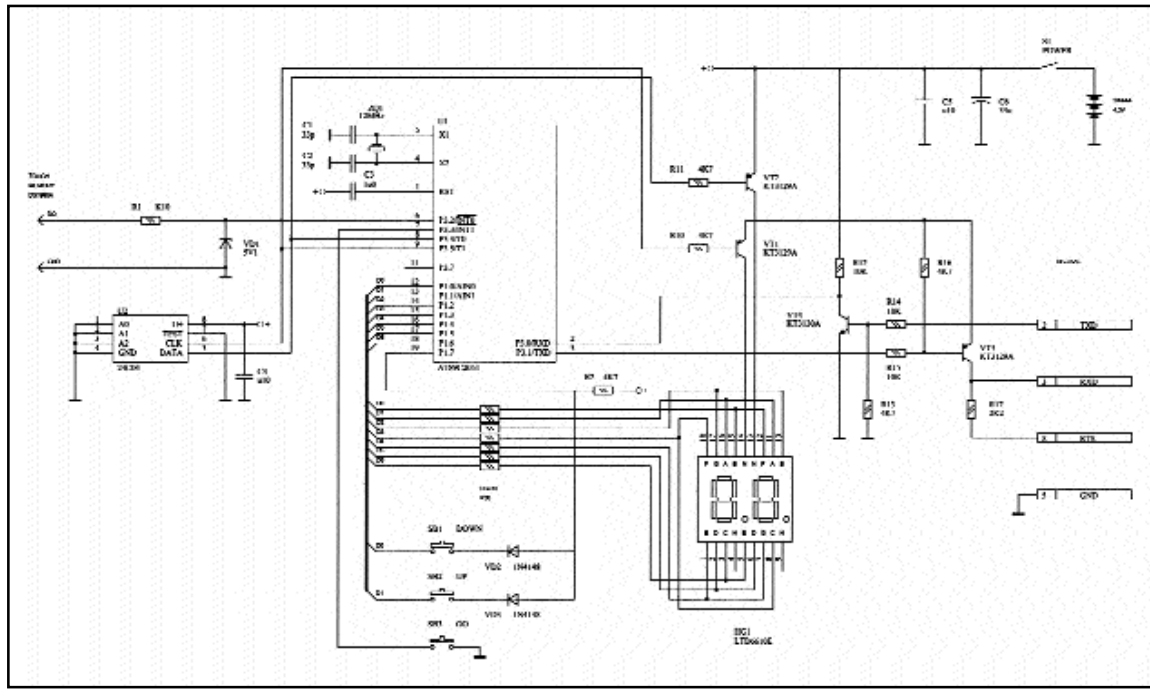

Рис. 5

индикатора HG1 можно применить два отдельных индикатора любого типа, необходимо только, чтобы они имели общий анод и, конечно, достаточную яркость свечения. Транзисторы можно применить любые маломощные соответствующей проводи-

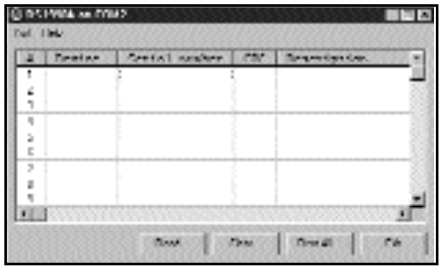

Рис. 6

мости с максимальным током коллектора не менее 50 мА. Для порта RS-232 применена вилка разъема D-SUB-9, которая предназначена для объемного монтажа. Плата входит между рядами контактов, которые припаиваются к соответствующим ламелям. Кнопки (без фиксации) и выключатель питания можно применить любые малогабаритные, однако при этом может потребоваться корректировка печатной платы.

При включении питания на индикаторах отображается текущий порядковый номер ключа. Порядковый номер запоминается в энергонезависимой памяти при входе в режим имитации. Всего может быть до 30 ключей. Пролистывать порядковый номер можно с помощью кнопок «Up» и «Down». Если к 1-проводному порту подключить DS1990A, то его серийный номер будет считан и записан в память под текущим порядковым номером. В случае удачного считывания на индикаторах на 2 секунды появляется надпись «Pr». Теперь устройство может имитировать считанный ключ. Для этого необходимо войти в режим имитации с помощью кнопки «Go». В этом режиме на индикатор выводится символ «P», а кнопки «Up» и «Down» не действуют. В режиме имитации 1-проводный порт можно подключить к считывателю touch memory, и он отреагирует на это как на подключение оригинальной DS1990A. Выйти из режима имитации можно вторичным нажатием кнопки «Go».

В имитаторе DS1990A возникла необходимость программно реализовать не только мастера, но и подчиненное устройство. Реализовать подчиненное устройство несколько сложнее ввиду того, что инициатором на шине является мастер. Однако это реализуемо, если во время имитации микроконтроллер не будет отвлекаться на другие задачи. Самая критичная по времени операция — это определение начала временного интервала (time slot). Для этого использована внутренняя логика прерываний. Роль порта 1-проводной шины выполняет ножка INT0 контроллера. Прерывание настроено таким образом, что флаг устанавливается при переходе входного сигнала из единицы в ноль. Само прерывание при этом запрещено, а флаг анализируется программно. Это позволяет устанавливать на шине данные с наименьшей задержкой относительно начала временного интервала:

```
JNB IE0,$          ;wait for interrupt flag
MOV DATA,C         ;DATA <- C
```

Вход в режим имитации осуществляется нажатием кнопки «Go». При этом программа начинает выполнять критичный по времени цикл и отвлекаться на сканирование индикатора и на опрос кнопок не может. В этом режиме индикация осуществляется статически, поэтому гореть может только один разряд (с несколько увеличенной яркостью). На индикаторе зажигается символ «P», для чего в порт P1 выводится соответствующая информация. Линия сканирования T0 устанавливается в ноль, при этом транзистор VT2 все время открыт, а второй разряд HG1 — все время включен. Выход из цикла имитации выполнен нестандартно. Кнопка «Go» подключена к ножке INT1 контроллера. При входе в цикл это прерывание разрешается. Его обработчик выглядит следующим образом:

```
GoKey:  MOV SP,#STACK
        MOV A,#LO(EDIT)
        PUSH ACC
        MOV A,#HI(EDIT)
        PUSH ACC
        RETI                ;return to Edit
```

Этот текст аналогичен по выполняемым действиям команде «LJMP Edit». Он осуществляет переход на метку EDIT в основной программе, где проводится начальная инициализация. Однако применять команду LJMP нельзя, так как в этом случае логика прерываний не восстановит своего состояния (она останется в состоянии обработки прерывания, и прерывание больше не возникнет).

Кроме автономной работы, устройство может работать с PC, к которому оно подключается через порт RS-232 (поддерживаются порты COM1 — COM4). Программа на PC (рис. 6) отображает параметры ключей, записанных в память имитатора.

В колонке # отображается серийный номер, Device — тип устройства, например DS1990A, DS1992, DS1994 и т. д. Нужно отметить, что устройство может имитировать только DS1990A, но это не мешает распознавать другие типы touch memory. В колонке Serial number отображается серийный номер ключа, в колонке CRC — результат проверки контрольной суммы («Ok» или ничего). Колонка Description предназначена для текстового описания ключа. Текстовые описания всех ключей сохраняются в ini-файле. А вот серийные номера ключей в ini-файл не записываются и хранятся только в энергонезависимой памяти устройства. Поэтому, имея доступ только к компьютеру, нельзя получить серийные номера.

Перемещение по строчкам таблицы эквивалентно пролистыванию порядковых номеров ключей с помощью кнопок «Up» и «Down». При этом на индикаторах устройства также меняется текущее значение. Поля колонки Serial number доступны для редактирования. Поэтому в память устройства можно ввести серийный номер ключа, просто набрав его в таблице и нажав Enter.

Кнопка Read служит для считывания содержимого памяти имитатора в таблицу. С помощью кнопки Clear можно стереть текущий ключ, а с помощью Clear All — стереть все ключи.

Последовательный порт настраивается следующим образом: скорость 4800 бод, 8 бит данных, 1 стоп-бит. Бит паритета является признаком передачи адреса. Обмен всегда инициирует PC. Порядок обмена следующий: в ОЗУ контроллера имитатора организован буфер обмена. PC может прочитать или записать любой байт этого буфера. Для этого PC должен вначале передать адрес байта. Признак передачи адреса — бит паритета, равный единице. Адрес имеет длину 5 бит, хотя в данном контроллере буфер имеет размер всего 10 байт. Поэтому имеют смысл только первые 10 адресов. Вслед за передачей адреса PC должен передать байт данных, который будет записан в буфер по этому адресу. Для чтения данных PC должен передать адрес, у которого бит D7 равен единице. Это признак запроса чтения. После получения такого адреса, контроллер сам передаст байт данных из буфера. Формат передаваемых данных наглядно представлен в табл.1.

Табл. 1(а)

P	D7	D6	D5	D4	D3	D2	D1	D0	
0				Data Byte					- запись данных
1	0	0	X			Address			- запись адреса
1	1	0	X			Address			- запись адреса и запрос чтения

Буфер обмена имеет следующую структуру:

Табл. 1(б)

Адрес	Имя	Описание
0	Buff	Команда
1	Chan	Порядковый номер ключа
2	Fam_Code	Код семейства ключа
3	Ser_Num1	Серийный номер ключа, байт 1
4	Ser_Num2	Серийный номер ключа, байт 2
5	Ser_Num3	Серийный номер ключа, байт 3
6	Ser_Num4	Серийный номер ключа, байт 4
7	Ser_Num5	Серийный номер ключа, байт 5
8	Ser_Num6	Серийный номер ключа, байт 6
9	CRC_Byte	Контрольная сумма

Подробное описание параметров:

Табл. 1(*c*)

0	Buff	Команда

Этот параметр представляет собой код команды, которая будет выполнена контроллером. Возможные коды команд приведены в табл. 2:

Табл. 2(а)

Код в Buff	Команда
0	Нет операции
1	Установить порядковый номер
2	Записать буфер обмена в энергонезависимую память

Команда «Установить порядковый номер» включает текущий номер в соответствии со значением в ячейке Chan буфера обмена. Поэтому прежде чем передавать код этой команды, необходимо записать в Chan требуемый номер.

Команда «Записать буфер обмена в энергонезависимую память» осуществляет копирование серийного но-

мера из буфера обмена в энергонезависимую память. Перед тем, как передать код этой команды, необходимо правильно заполнить буфер и установить требуемый порядковый номер с помощью предыдущей команды.

После выполнения любой команды контроллер возвращает ее код в PC как подтверждение выполнения.

Табл. 2(б)

1	Chan	Порядковый номер ключа

Этот параметр задает порядковый номер ключа и может находиться в диапазоне от 1 до 30.

Табл. 2(c)

2	Fam_Code	Код семейства ключа

Этот параметр является кодом семейства ключа. Имеет смысл записывать только 01H (код семейства DS1990A).

Табл. 2(d)

3	Ser_Num1	Серийный номер ключа, байт 1
4	Ser_Num2	Серийный номер ключа, байт 2
5	Ser_Num3	Серийный номер ключа, байт 3
6	Ser_Num4	Серийный номер ключа, байт 4
7	Ser_Num5	Серийный номер ключа, байт 5
8	Ser_Num6	Серийный номер ключа, байт 6

В эти ячейки записывается серийный номер ключа.

Табл. 2(e)

9	CRC_Byte	Контрольная сумма

Этот параметр должен быть равен контрольной сумме предыдущих семи ячеек (Fam_Code, Ser_Num1 .. Ser_Num6).

Исходный текст программы на Delphi находится на сервере журнала.

Программа использует динамическую библиотеку ComAPI32.DLL, которую можно найти там же.

Исходный текст программы для микроконтроллера DS1990A.Asm транслируется с помощью TASM 2.76. Использованы библиотеки LibReg.Asm, LibRTC.Asm, LibCom.Asm, LibMac.Asm. Все исходные тексты находятся на сервере журнала. Там же можно найти файлы топологии печатной платы и принципиальной схемы устройства.

Программу прошивки смотрите на www.compitech.ru/data/shem/01_00/#3

Леонид Ридико, Минск