DS1961S
iButton™ с 1Кб защищенной EPROM и блоком SHA-1

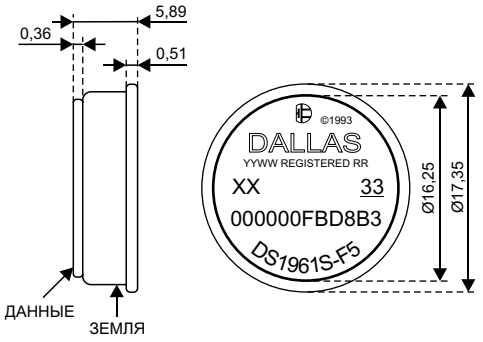
ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

- Электрически программируемая и стираемая при напряжении 5В память (EEPROM) объемом 1128 бит, разделенная на четыре страницы по 256 бит, 64-битный секретный код только для записи и до пяти регистров общего назначения для записи/чтения
- Для осуществления записи требуется знание секретного кода и возможность вычисления и передачи 160-битного MAC-кода (Message Authentication Code, кода аутентификации сообщения) для авторизации
- Память секретного кода и память данных может быть защищена от записи (целиком, или только страница 0) или переведена в режим эмуляции EPROM («запись в 0», страница 1)
- Встроенный 512-битный блок SHA-1, который предназначен для вычисления 160-битного MAC-кода и генерации секретных кодов
- Считывание и запись возможны в диапазоне напряжений питания от 2,8В до 5,25В и диапазоне температур от -40°С до +85°С
- Обмен с хостом осуществляется по одной цифровой линии на скорости 14,1 Кб в секунду с помощью 1-проводного протокола (1-Wire® Protocol)
- Встроенный генератор 16-битной контрольной суммы (CRC) обеспечивает безопасный обмен данными
- Ускоренный режим позволяет повысить скорость обмена до 125 Кбит в секунду
- Рабочий диапазон температур от -40°С до +85°С
- Длительность хранения данных не менее 10 лет при температуре +85°С

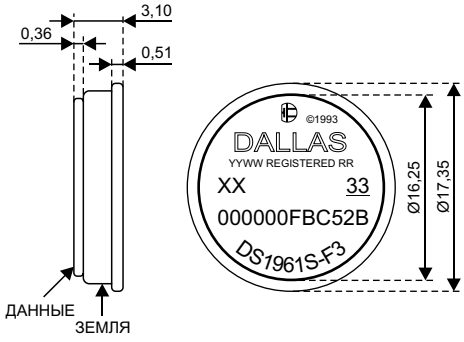
ОБЩИЕ ХАРАКТЕРИСТИКИ iButton

- Уникальный, занесенный лазером и проверенный на этапе изготовления 64-битный регистрационный номер (8-битный код семейства + 48-битный серийный номер + 8-битная контрольная сумма CRC) гарантирует абсолютный контроль, так как не существует двух устройств с одинаковыми номерами
- Встроенный контроллер многоточечной сети MicroLAN
- Цифровая идентификация и получение информации в одно касание
- Компактный носитель информации в виде кристалла микросхемы
- Данные могут быть доступными при касании объекта
- Форма в виде таблетки обеспечивает автоматическое центрирование в считывающем устройстве
- Долговечный корпус из нержавеющей стали с гравированным регистрационным номером устойчив к внешним воздействиям
- Легко прикрепляется с помощью самоклеющейся подложки, фиксируется собственным фланцем или напрессовываемым кольцом
- Детектор присутствия выдает ответ, когда считыватель в первый раз подает напряжение питания
- Соответствует UL#913 (4-я редакция); взрывобезопасное исполнение, утверждено для использования в классе I, отделение 1, группы A, B, C и D (в зависимости от приложения)

F5 MICROCAN



F3 MICROCAN



Все размеры приведены в миллиметрах.

ИНФОРМАЦИЯ ДЛЯ ЗАКАЗА

DS1961S-F5	F5 iButton
DS1961S-F3	F3 iButton

ПРИМЕРЫ АКСЕССУАРОВ

DS1963S	SHA сопроцессор в виде iButton
DS9096P	Самоклеющаяся подложка
DS9101	Универсальный зажим
DS9093RA	Крепежное кольцо
DS9093F	Держатель с защелкой
DS9092	Панелька для iButton

ОПИСАНИЕ iButton

DS1961S включает в себя 1024 бита EEPROM, память 64-битного секретного кода, 8-байтную страницу регистров/байтов управления, имеющую до пяти программируемых пользователем байт, 512-битный блок SHA-1 и полнофункциональный 1-проводный интерфейс в прочном корпусе iButton. Данные передаются последовательно с помощью 1-проводного протокола, который требует только одного вывода данных и общего провода. DS1961S имеет дополнительную область данных, которая называется блокнотом и работает как буфер при записи основной памяти, страницы регистров или при изменении секретного кода. Данные вначале записываются в блокнот, откуда они могут быть считаны. После проверки правильности данных, команда копирования блокнота пересылает их в нужное место памяти, если был принят правильный 160-битный MAC-код. При вычислении MAC-кода используется секретный код и дополнительные данные, которые сохранены в DS1961S, включая данные идентификационного регистра устройства. Без MAC-кода может быть загружен только новый секретный код. Блок SHA-1 также может использоваться для вычисления 160-битного MAC-кода при чтении страницы памяти или при вычислении нового секретного кода вместо его загрузки.

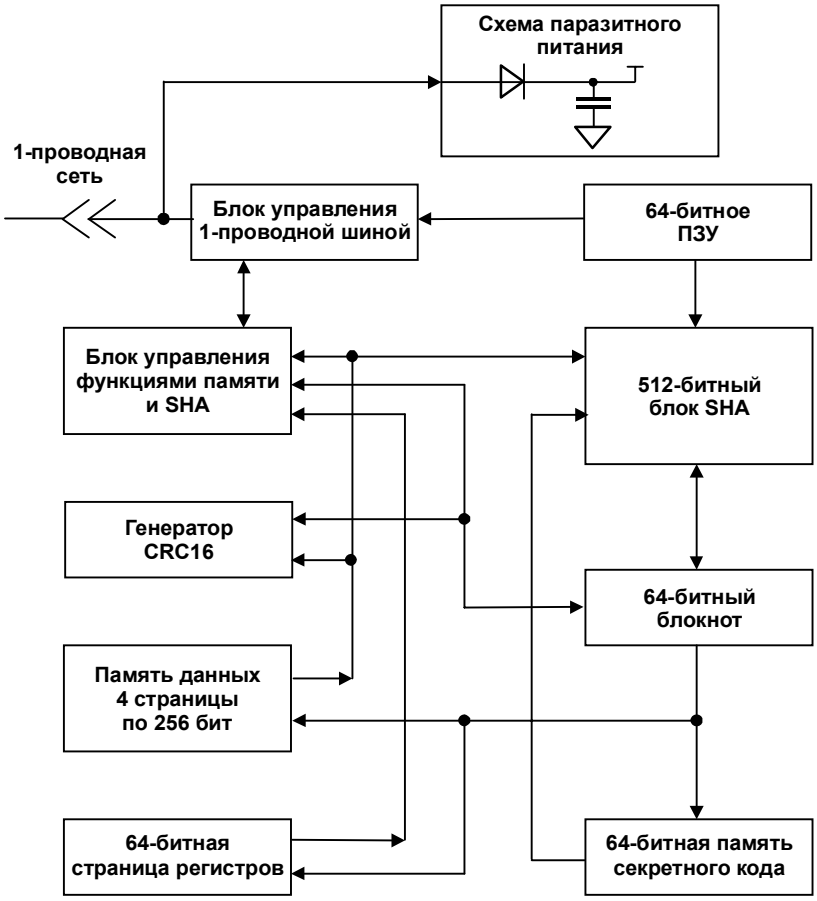
DS1961S воспринимает особую команду «обновление блокнота». Правильное использование последовательности обновления после операции копирования блокнота уменьшает количество ошибок, связанных с ненадежно запрограммированными битами в условиях контактирования касанием (см. раздел *«Запись с проверкой»*). Последовательность обновления также обеспечивает возможность восстановления функционирования устройства с ненадежно запрограммированными битами.

DS1961S имеет собственный 64-битный регистрационный номер, который записан в ПЗУ лазером в процессе изготовления, что обеспечивает гарантированную идентификацию и позволяет осуществлять абсолютный контроль. Долговечный корпус MicroCan исключительно устойчив к агрессивным внешним условиям, таким как грязь, влажность и удары. Его компактный профиль в форме таблетки автоматически центрирует прибор в считывающем устройстве, что помогает пользователям легко им оперировать. Аксессуары позволяют монтировать DS1961S практически на любые поверхности, включая пластиковые держатели и идентификационные бэджи.

ПРИМЕНЕНИЯ

DS1961S может использоваться для разных целей, таких как ограничение доступа, аутентификация предмета или пользователя, послепроизводственный контроль за продукцией и проведение электронных платежей. Как носитель электронных денег (eCash), DS1961S может сохранять до трех расчетных файлов или «кошельков» одного провайдера услуг, что делает устройство подходящим для применения в небольших фирмах, таких как кафетерии, копировальные участки, парки для отдыха или частные клубы. Для повышения безопасности, а также в случае недостаточной вычислительной мощности микроконтроллера хоста, возможно совместное использование сопроцессора DS1963S, который может служить для проверки MAC-кода, сгенерированного DS1961S или для вычисления MAC-кода, необходимого для осуществления записи в DS1961S.

Рис. 1. БЛОК-СХЕМА DS1961S



ОБЗОР

Блок-схема, показанная на рис. 1, демонстрирует связи между блоками управления и блоками памяти DS1961S. Всего DS1961S имеет шесть основных компонентов хранения и обработки данных: 1) 64-битное ПЗУ, записанное лазером, 2) 64-битный блокнот, 3) четыре 32-байтных страницы EEPROM, 4) 64-битная страница регистров, 5) 64-битная память секретного кода и 6) 512-битный блок SHA-1 (Secure Hash Algorithm).

Иерархическая структура 1-проводного протокола показана на рис. 2. Мастер шины вначале должен послать одну из семи команд функций ПЗУ: 1) Чтение ПЗУ, 2) Сравнение ПЗУ, 3) Поиск ПЗУ, 4) Пропуск ПЗУ, 5) Продолжение обмена, 6) Пропуск ПЗУ в ускоренном режиме или 7) Сравнение ПЗУ в ускоренном режиме. По окончании команд ПЗУ ускоренного режима, посланных на стандартной скорости, устройство переходит в ускоренный режим, когда обмен

данными осуществляется на повышенной скорости. Протокол, который требуется для передачи команд функций ПЗУ, показан на рис. 9. После того, как команда функции ПЗУ успешно выполнена, становятся доступными функции памяти, и мастер может передать одну из восьми команд функций памяти и SHA. Протокол для этих команд показан на рис. 7. При считывании и записи всех данных первым передается младший бит.

64-БИТНОЕ ПЗУ, ЗАПИСАННОЕ ЛАЗЕРОМ

Каждый экземпляр DS1961S содержит в ПЗУ уникальный код длиной 64 бита. Первые 8 бит являются кодом семейства. Следующие 48 бит являются уникальным серийным номером. Последние 8 бит являются контрольной сумой (CRC) первых 56 бит (см. рис. 3). Контрольная сумма получена с помощью генератора, выполненного на основе сдвигового регистра и элементов «исключающее ИЛИ», как показано на рис. 4, и использующего полином $X^8 + X^5 + X^4 + 1$. Дополнительную информацию о контрольной сумме, используемой фирмой Dallas Semiconductor, можно найти в книге *«Book of DS19xx iButton Standards»*. Биты сдвигового регистра инициализируется нулем. Затем, начиная с младшего бита кода семейства, по одному биту в сдвиговый регистр вводятся данные. После ввода 8-го бита кода семейства вводятся биты серийного номера. После ввода 48-го бита серийного номера сдвиговый регистр содержит значение CRC. Если ввести еще 8 бит CRC, то содержимое регистра вновь станет равным нулю.

Рис. 2. ИЕРАРХИЧЕСКАЯ СТРУКТУРА 1-ПРОВОДНОГО ПРОТОКОЛА

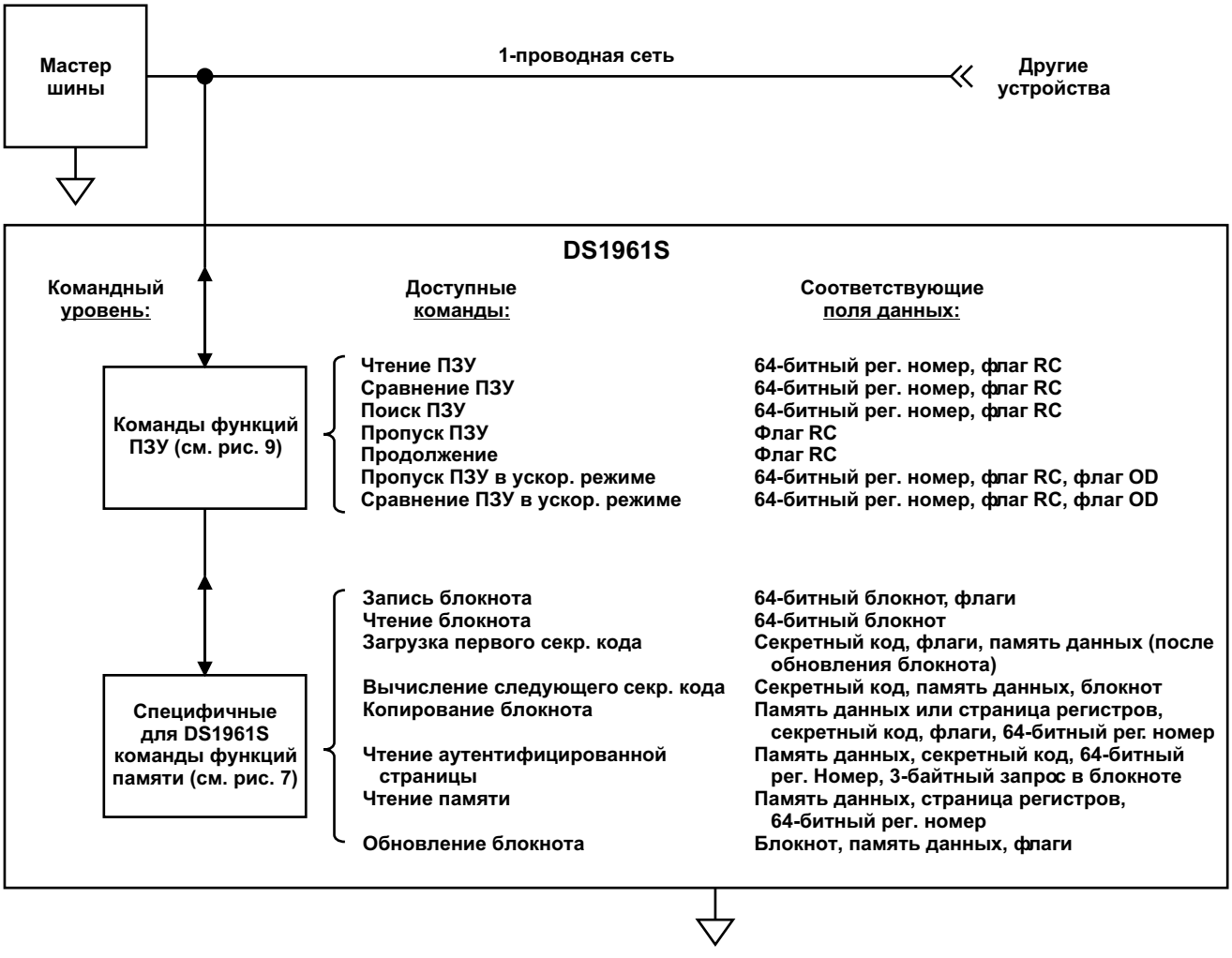


Рис. 3. 64-БИТНОЕ ПЗУ, ЗАПИСАННОЕ ЛАЗЕРОМ

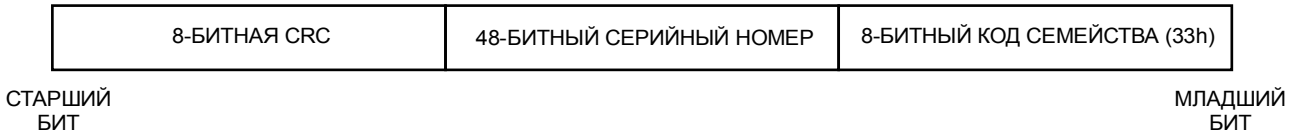
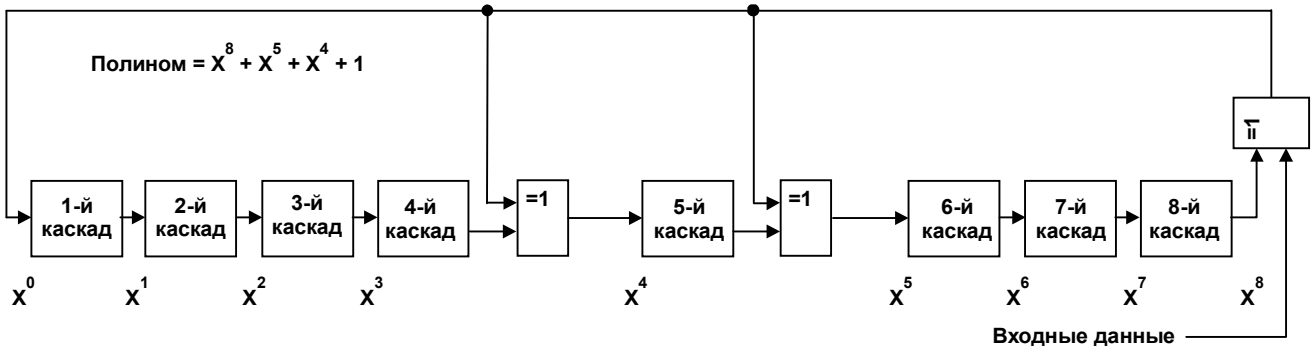


Рис. 4. ГЕНЕРАТОР CRC



КАРТА ПАМЯТИ

DS1961S имеет четыре области памяти: память данных, память секретного кода, блокнот и страница регистров с байтами пользователя и регистрами специальных функций. Память данных организована в виде страниц по 32 байта. Секретный код, страница регистров и блокнот имеют размер по 8 байт. Блокнот используется как буфер при записи памяти данных, при загрузке секретного кода и при записи страницы регистров.

Память данных, память секретного кода и страница регистров размещены в линейном адресном пространстве, как показано на рис. 5. Память данных и страница регистров имеют неограниченный доступ для чтения. Запись в память данных и страницу регистров требует знания секретного кода.

Секретный код может быть запрограммирован путем копирования данных из блокнота в память секретного кода или путем вычислений, использующих текущее значение секретного кода и содержимое блокнота как часть секретного кода. Секретный код не может быть непосредственно считан; доступ к нему имеет только блок SHA для вычисления кодов аутентификации сообщений.

Диапазон адресов 0088h – 008Fh, называемый страницей регистров, содержит регистры специальных функций, а также байты пользователя общего назначения и один байт производителя. Будучи однажды запрограммированными значением AAh или 55h, большинство этих байтов становятся доступными только для чтения и не могут быть изменены в дальнейшем. Любые другие коды не приводят к включению защиты от записи или активизации каких-либо функций, связанных с отдельными байтами. Специальными функциями являются: 1) защита от записи только секретного кода, 2) защита от записи одновременно всех четырех страниц памяти данных, 3) включение режима EPROM для страницы 1 памяти данных и 4) защита от записи только страницы 0 памяти данных. Когда включен режим EPROM, состояние битов в диапазоне адресов 0020h – 003Fh может быть изменено только с 1 на 0, если память данных не защищена от записи.

Рис. 5. КАРТА ПАМЯТИ DS1961S

ДИАПАЗОН АДРЕСОВ	ОПИСАНИЕ	ПРИМЕЧАНИЯ
0000h – 001Fh	Страница 0 памяти данных	Доступ для записи невозможен без секретного кода
0020h – 003Fh	Страница 1 памяти данных	Доступ для записи невозможен без секретного кода
0040h – 005Fh	Страница 2 памяти данных	Доступ для записи невозможен без секретного кода
0060h – 007Fh	Страница 3 памяти данных	Доступ для записи невозможен без секретного кода
0080h – 0087h	Память секретного кода	Доступ для чтения отсутствует; для записи знания секретного кода не требуется
0088h [1)]	Защита от записи секретного кода	Защита активизируется кодом AAh или 55h
0089h [1)]	Защита от записи страниц 0 - 3	Защита активизируется кодом AAh или 55h
008Ah [1)]	Байт пользователя, защита от записи самого себя	Защита активизируется кодом AAh или 55h
008Bh	Байт производителя (только для чтения)	Считывается AAh или 55h; см. текст
008Ch [1)]	Байт пользователя/управление режимом EPROM для страницы 1	Режим активизируется кодом AAh или 55h
008Dh [1)]	Байт пользователя/защита от записи только страницы 0	Защита активизируется кодом AAh или 55h
008Eh – 008Fh	Байт пользователя/код производителя	Функция зависит от значения байта производителя
0090h – 0097h	64-битный идентификационный регистр	Доступ только для чтения

[1)] Будучи однажды запрограммированными значением AAh или 55h, эти адреса становятся доступными только для чтения. Любые другие сохраненные коды не приводят к включению защиты от записи или активизации каких-либо функций.

Байт производителя может иметь значение 55h или AAh. Обычно он имеет значение 55h, указывая на то, что адреса 008Eh и 008Fh являются пользовательскими байтами для записи/чтения и не имеют никаких специальных функций или механизма защиты. Значение AAh указывает на то, что эти два адреса содержат запрограммированный производителем 16-битный идентификатор и защищены от записи. Идентификатор производителя может представлять собой идентификационный код, предоставленный потребителем, который позволяет прикладному программному обеспечению идентифицировать соответствующие DS1961S и быстрее выбирать подходящий секретный код. Для назначения и регистрации идентификатора производителя связывайтесь с изготовителем.

Диапазон адресов 0090h – 0097h называется идентификационным регистром. Обычно идентификационный регистр содержит копию регистрационного номера, записанного в ПЗУ устройства. Код семейства сохраняется по младшему адресу, за ним следует 48-битный серийный номер и 8-битная CRC, которая сохраняется по адресу 0097h. При чтении этих адресов (0090h – 0097h) мастер шины принимает отдельные биты регистрационного номера точно в той же последовательности, что и при выполнении соответствующей команды функций ПЗУ. В заказных версиях содержимое идентификационного регистра может представлять собой любую заданную потребителем последовательность. Для получения более подробной информации обращайтесь к изготовителю.

Рис. 6. АДРЕСНЫЕ РЕГИСТРЫ

Номер бита	7	6	5	4	3	2	1	0
Адрес назначения (TA1)	T7	T6	T5	T4	T3	T2 (0)	T1 (0)	T0 (0)
Адрес назначения (TA2)	T15	T14	T13	T12	T11	T10	T9	T8
Конечный адрес со статусом данных (E/S) (только для чтения)	AA	1	PF	1	1	E2 (1)	E1 (1)	E0 (1)

АДРЕСНЫЕ РЕГИСТРЫ И СОСТОЯНИЕ ПЕРЕСЫЛКИ

DS1961S использует три адресных регистра: TA1, TA2 и E/S (рис. 6). Такие регистры имеются и во многих других 1-проводных устройствах, но в DS1961S они работают несколько по-другому. Регистры TA1 и TA2 загружаются адресом назначения, который указывает, куда должны быть записаны или откуда считаны данные. Регистр E/S является регистром состояния пересылки и доступен только для чтения. Он используется для проверки целостности данных при выполнении команд записи. Так как блокнот DS1961S способен принимать данные только в виде блоков по 8 байт, три младших бита регистра TA1 всегда равны 0, а три младших бита регистра E/S (конечное смещение) всегда считываются как 1. Это означает, что все данные из блокнота используются для последующего копирования в основную память или память секретного кода. Бит 5 регистра E/S, называемый PF или флагом неполного байта (partial byte flag), устанавливается в 1, если количество бит данных, переданных мастером, не кратно восьми, или если данные в блокноте не являются действительными в результате пропадания питания. Успешная запись в блокнот очищает бит PF. Биты 3, 4 и 6 не несут никаких функций; они всегда считываются как 1. Флаг неполного байта поддерживает осуществляемую мастером проверку целостности данных после команды записи. Старший бит регистра E/S называется AA или флагом принятия авторизации (authorization accepted flag). Он указывает на то, что данные, сохраненные в блокноте, уже были скопированы в память по адресу назначения. Запись данных в блокнот очищает этот флаг.

ЗАПИСЬ С ПРОВЕРКОЙ

Для промежуточного хранения данных, записываемых в DS1961S, используется блокнот. Вначале мастер посылает команду записи блокнота. Заметьте, что запись в память должна производиться в рамках 8-байтного фрагмента с тремя младшими битами T2..T0 адреса назначения, равными нулю. Если биты T2..T0 будут переданы ненулевыми, устройство все равно их сбросит и будет использовать этот модифицированный адрес в качестве адреса назначения. Мастер всегда должен передавать восемь полных байт данных. Когда восемь байт данных переданы, мастер может выбрать чтение инвертированной CRC16, рассчитанной для кода команды записи блокнота, адреса и данных, переданных мастером. Мастер может сравнить эту CRC со значением, вычисленным им самим, для подтверждения правильности пересылки данных. После того, как блокнот записан, мастер должен всегда выполнять его чтение, чтобы убедиться в правильности записанных данных. Во время чтения блокнота DS1961S передает адрес назначения TA1 и TA2, а также содержимое регистра E/S. Флаг неполного байта (бит 5 регистра E/S) устанавливается в 1 в том случае, если последний байт данных, принятый DS1961S в процессе записи или обновления блокнота, был неполным, или если имело место пропадание питания во время последней процедуры записи блокнота. Флаг принятия авторизации (AA, бит 7 регистра E/S) обычно очищается при записи или обновлении блокнота; однако если обнаруживается, что он установлен в 1, значит, не была распознана команда записи или обновления блокнота. В этом случае мастер должен снова

перезаписать блокнот. После приема содержимого регистра E/S чтение блокнота заканчивается. Что происходит с данными блокнота в различных условиях, можно выяснить из описания команд записи и обновления блокнота. Вслед за данными блокнота следует инвертированная CRC, вычисленная для команды чтения блокнота, адреса назначения, регистра E/S и данных блокнота. Как и для команды записи блокнота, мастер может сравнить эту CRC со значением, вычисленным им самим, для подтверждения правильности пересылки данных. После проверки данных мастер может передать команду копирования блокнота в память. Как вариант, может быть передана команда загрузки первого секретного кода или вычисления следующего секретного кода, которые служат для изменения секретного кода. Для более подробной информации смотрите описание этих команд.

В устройствах, контактируемых касанием, надежность электрического контакта не может быть гарантирована. При плохом или прерывающемся контакте возможно завершение выполнения команды копирования блокнота при недостаточном напряжении питания, что может привести к появлению в битах EEPROM плавающих затворов с напряжением в промежуточной области между 0 и 1. Если такое произойдет, логическое состояние этих битов будет неопределенным. В зависимости от напряжения питания и окружающей температуры один и тот же бит может считываться хостом в одном состоянии, а встроенным блоком SHA-1 – в противоположном. Это приведет к полной блокировке устройства, так как хост не сможет сформировать правильный MAC-код для разрешения перезаписи этого бита. Для восстановления неудачно записанных битов и восстановления функционирования устройства была введена команда обновления блокнота. Вместе с командой загрузки первого секретного кода, команда обновления блокнота обеспечивает возможность восстановления битов EEPROM в нормальное состояние, устраняя блокировку устройства и позволяя снова производить запись.

Для предотвращения появления неудачно записанных битов после каждой команды копирования блокнота должна быть выполнена последовательность обновления. Последовательность обновления включает в себя команду обновления блокнота (с того же адреса, который использовался предыдущей командой копирования блокнота), за которой следует загрузка первого секретного кода. Команда обновления блокнота устанавливает флаг EN_LFS. Этот флаг разрешает команде загрузки первого секретного кода производить запись по адресам 0000h – 007Fh, т.е. в память данных. Использование команды загрузки первого секретного кода позволяет мастеру скопировать блокнот в память без необходимости вычисления MAC-кода. Если мастер передаст после команды обновления блокнота любую другую команду, которая изменяет данные в блокноте или адрес назначения, флаг EN_LFS будет сброшен в 0. Это исключает использование команды загрузки первого секретного кода для записи данных, отличных от данных обновленного блокнота, и запись по адресу, отличному от установленного во время обновления блокнота. Для адресов назначения 0080h и выше команда обновления блокнота ведет себя точно так же, как команда записи блокнота. В этом случае флаг EN_LFS не устанавливается, поэтому невозможно обновить данные в памяти секретного кода (0080h) или в странице регистров (0088h). Это защищает секретный код от считывания с помощью команды обновления блокнота, за которой следует чтение блокнота.

КОМАНДЫ ФУНКЦИЙ ПАМЯТИ И SHA

В соответствии с требованиями безопасности, которые учтены в конструкции, DS1961S ведет себя иначе, нежели другие устройства iButton с памятью. Несмотря на то, что большая часть памяти DS1961S может быть считана таким же образом, как и у других устройств iButton с памятью, попытки чтения секретного кода приведут к считыванию байтов, равных FFh, вместо реальных данных. *Блок-схема функций памяти и SHA* (рис. 7) описывает протоколы, необходимые для адресации памяти и работы блока SHA. Обмен между мастером и DS1961S может происходить на обычной скорости (по умолчанию, OD = 0) или в ускоренном режиме на повышенной скорости (OD = 1). Если DS1961S специально не перевести в ускоренный режим, обмен будет происходить на обычной скорости.

Запись блокнота [0Fh]

Команда записи блокнота используется во время процедуры записи в память данных, память секретного кода и по разрешенным для записи адресам страницы регистров. Если мастер шины передает адрес назначения, больший 90h, то команда не выполняется.

После выдачи команды записи блокнота мастер должен передать сначала 2-х байтный адрес назначения, а затем данные, предназначенные для записи в блокнот. Данные записываются в блокнот с его начального адреса. Заметьте, что конечное смещение (E2..E0, см. рис. 6) всегда равно 111b, и не зависит от количества переданных мастером байт. По этим соображениям мастер всегда должен передавать восемь байт, особенно в том случае, когда данные предназначаются для загрузки в память секретного кода. Если мастер посылает менее восьми байт данных и не считывает блокнот для проверки, часть нового секретного кода будет случайной и неизвестной мастеру. Принимаются только полные байты данных. Если последний байт данных является неполным, он игнорируется и устанавливается флаг неполного байта (PF).

При выполнении команды записи блокнота внутренний генератор CRC (см. рис. 12) вычисляет CRC всего потока данных, начиная с кода команды и заканчивая последним байтом данных, переданных мастером. CRC генерируется с использованием полинома CRC16. Вначале генератор CRC очищается, затем в сдвиговый регистр по одному биту вводится код команды записи блокнота (0Fh), адрес назначения (TA1 и TA2) и все байты данных. Заметьте, что вычисление CRC16 производится со значением TA1, переданным мастером, несмотря на то, что DS1961S обнуляет биты T2..T0 адреса назначения при выполнении команды записи блокнота в память. Если блокнот полностью заполнен, мастер может выдать 16 интервалов чтения и принять значение CRC, вычисленное DS1961S. Если мастер продолжит чтение после получения CRC, все последующие считанные данные будут равны FFh.

После приема адреса назначения (TA1 и TA2) DS1961S очищает флаг EN_LFS. Если включен режим EPROM и должна производиться запись в страницу 1 (0020h – 003Fh), блокнот загружается данными, представляющими собой результат операции «логическое И» между данными, передаваемыми мастером, и текущим содержимым памяти по адресам назначения. Если запись должна производиться в страницу регистров (0088h – 008Fh), то для всех защищенных от записи байтов данные в блокноте, переданные мастером, заменяются текущим содержимым этих байтов в памяти. Во всех других случаях данные, передаваемые мастером, записываются в блокнот без изменений.

Чтение блокнота [AAh]

Команда чтения блокнота позволяет произвести проверку правильности адреса назначения и данных, записанных в блокнот. После выдачи кода команды мастер приступает к чтению. Два первых байта представляют собой адрес назначения, биты T2..T0 которого обнулены. Следующий байт представляет собой конечное смещение/статус данных (E/S). За ним следуют данные, содержащиеся в блокноте, которые могут отличаться от данных, переданных туда мастером. Это возможно в том случае, если адресом назначения является память секретного кода, страница регистров, страница 1 памяти (в режиме EPROM) или если для загрузки блокнота использовалась команда обновления. В этих случаях блокнот может содержать данные, отличные от тех, что записывались туда командами записи или обновления блокнота. Мастер должен считать блокнот до конца, после чего он примет инвертированную CRC для всех считанных данных. Если мастер продолжит чтение после получения CRC, все последующие считанные данные будут равны FFh.

Блокнот может быть загружен командами записи или обновления блокнота. Данные, находящиеся в блокноте, зависят от использованной команды, адреса назначения, и от того, включен ли режим EPROM. Для более подробной информации смотрите описание команд записи и обновления блокнота.

Загрузка первого секретного кода [5Ah]

Команда загрузки первого секретного кода имеет два режима работы, которыми управляет флаг EN_LFS. Когда EN_LFS = 0, команда заменяет текущий секретный код устройства содержимым блокнота, если память секретного кода не защищена от записи. Когда EN_LFS = 1, команда позволяет перезаписать память данных (адреса 0000h – 007Fh) без необходимости вычисления SHA-1, в отличие от выполнения тех же действий командой копирования блокнота. Флаг EN_LFS всегда имеет нулевое значение, за исключением случая, когда он устанавливается в единицу при выполнении команды обновления блокнота перед загрузкой первого секретного кода.

Случай EN_LFS = 0

Перед использованием команды загрузки первого секретного кода в этом режиме мастер должен записать новое значение секретного кода в блокнот, используя начальный адрес памяти секретного кода (0080h). После выдачи команды загрузки первого секретного кода, мастер должен передать 3-байтную последовательность авторизации (TA1, TA2, E/S, в этом порядке), которая должна быть непосредственно перед этим получена с помощью команды чтения блокнота. Эта 3-байтная последовательность должна точно совпадать с данными, которые содержатся в трех адресных регистрах (см. рис. 6). Если последовательность совпадает, и память секретного кода не защищена от записи, устанавливается флаг AA и начинается копирование. В память секретного кода копируются все 8 байт блокнота.

Случай EN_LFS = 1

Для использования команды загрузки первого секретного кода в этом режиме должна быть выполнена команда обновления блокнота для загрузки в блокнот восьми байт памяти данных (из диапазона адресов 0000h – 007Fh), что установит флаг EN_LFS в единицу. После выдачи команды загрузки первого секретного кода мастер должен передать 3-байтную последовательность авторизации (TA1, TA2, E/S, в этом порядке), которая должна быть непосредственно перед этим получена с помощью команды чтения блокнота. Эта 3-байтная последовательность должна точно совпадать с данными, которые содержатся в трех адресных регистрах (см. рис. 6). Если последовательность совпадает, и память не защищена от записи, устанавливается флаг AA и начинается копирование. В память копируются все 8 байт блокнота.

Независимо от использованного режима, длительность процесса копирования равна t_{PROG}, во время которого напряжение на 1-проводной шине не должно опускаться ниже 2,8В. По окончании задержки на время копирования мастер должен считать, по крайней мере, один байт. Считанное значение AAh говорит о том, что копирование прошло успешно, а значение FFh говорит об ошибке копирования. Как альтернатива, вместо использования загрузки первого секретного кода при EN_LFS = 0 новый секретный код может быть загружен командой копирования блокнота. Однако этот подход требует знания текущего секретного кода и вычисления 160-битного MAC-кода.

Рис. 7-1. БЛОК-СХЕМА ФУНКЦИЙ ПАМЯТИ И SHA

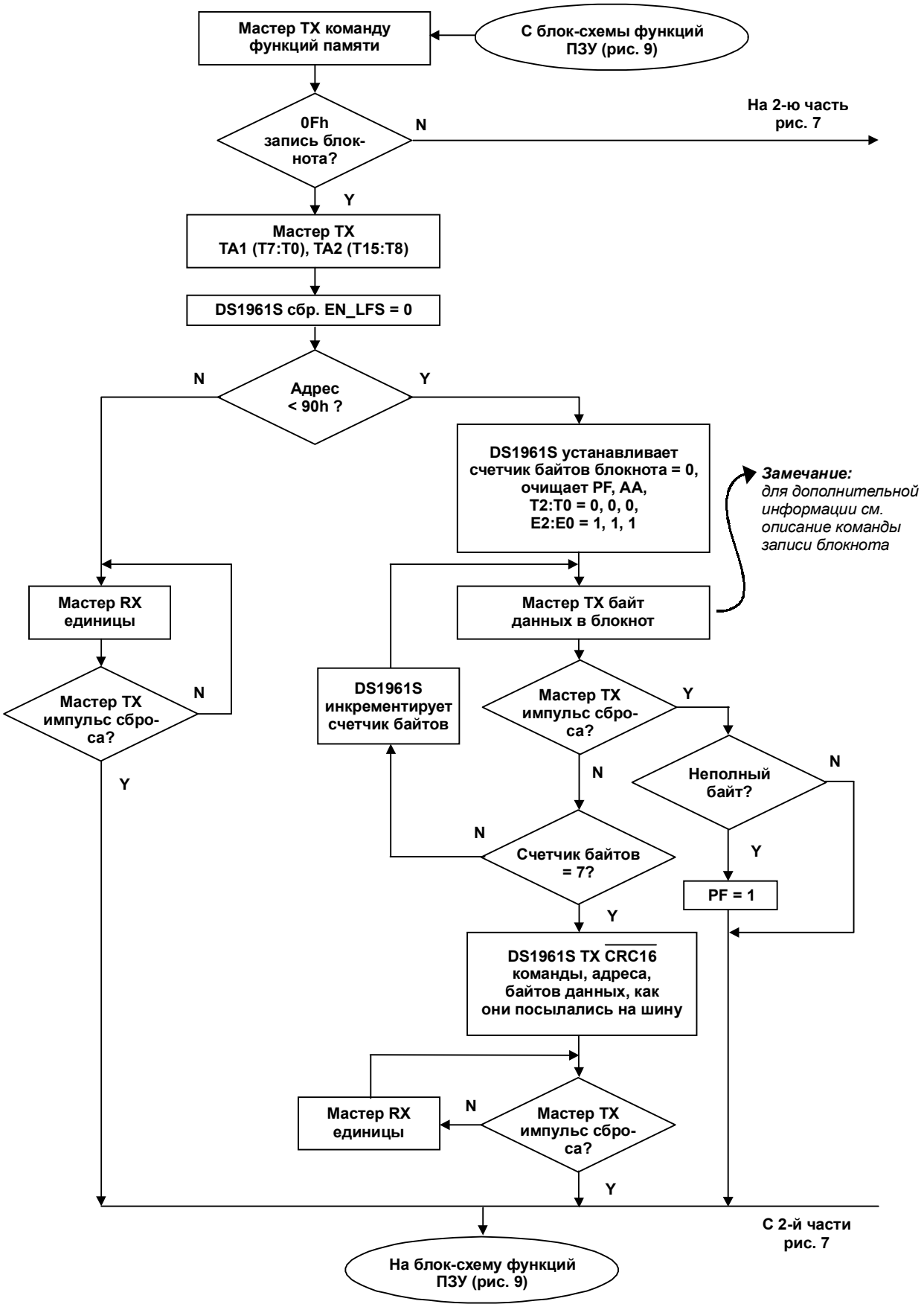


Рис. 7-2. БЛОК-СХЕМА ФУНКЦИЙ ПАМЯТИ И SHA (продолжение)

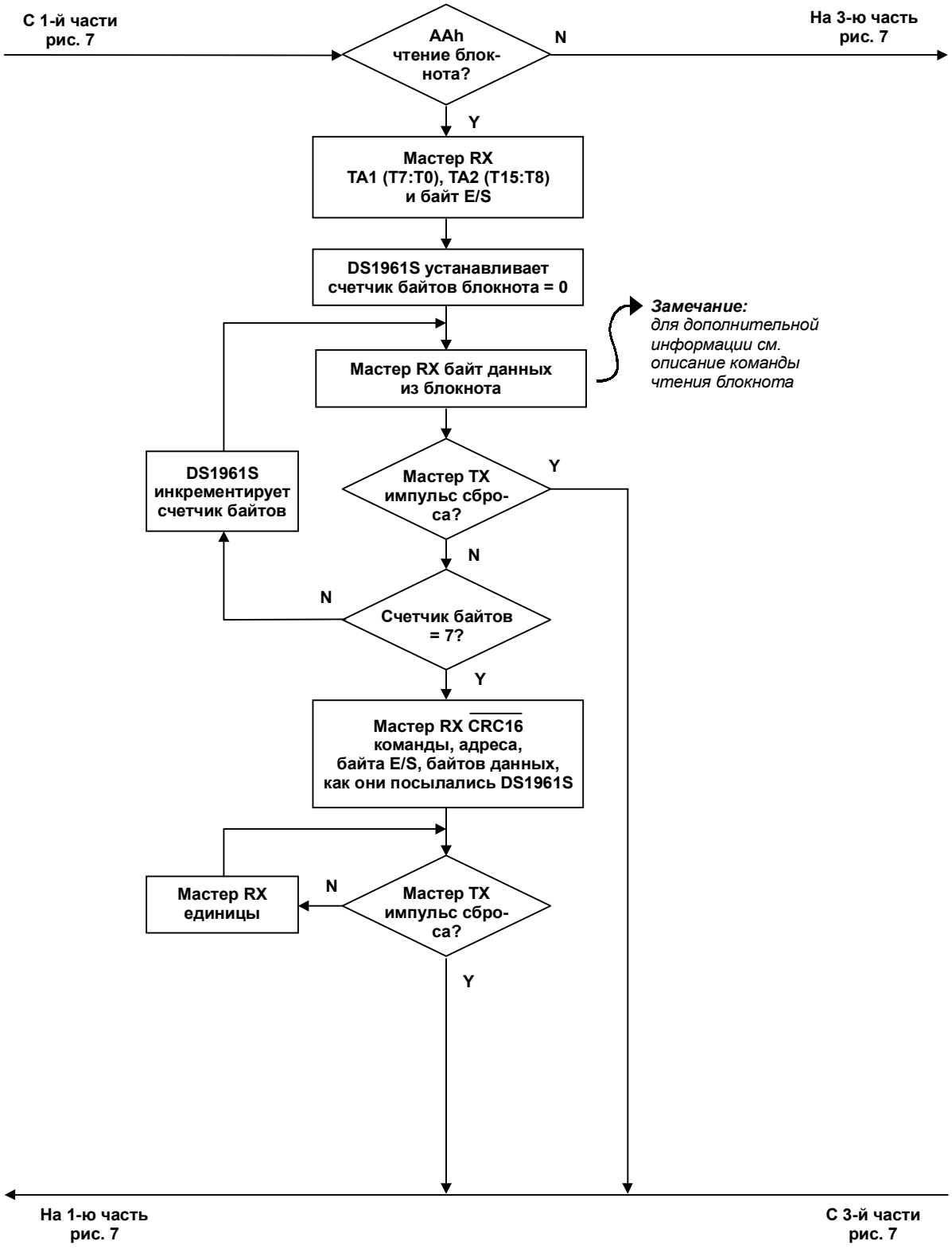


Рис. 7-3. БЛОК-СХЕМА ФУНКЦИЙ ПАМЯТИ И SHA (продолжение)

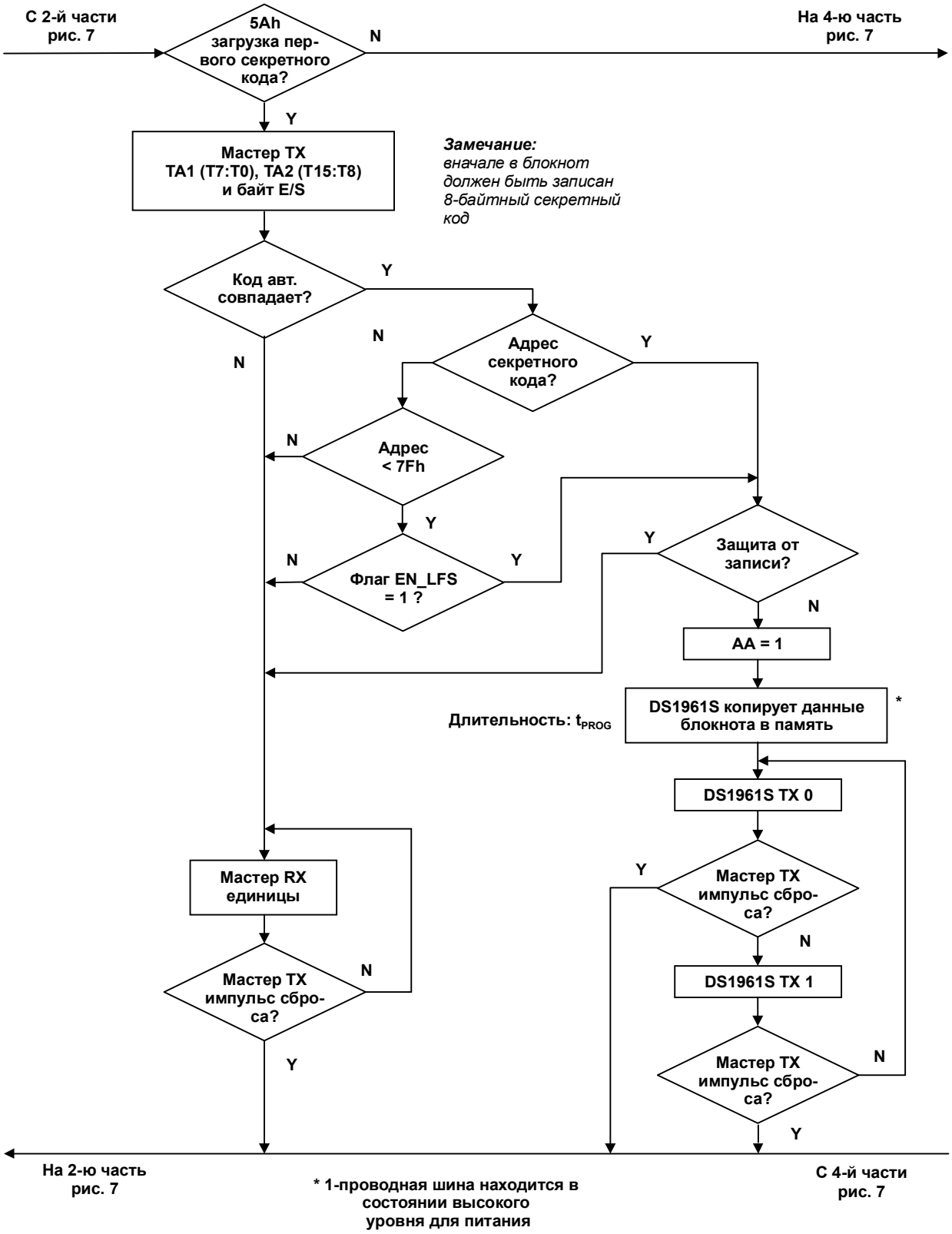


Рис. 7-4. БЛОК-СХЕМА ФУНКЦИЙ ПАМЯТИ И SHA (продолжение)

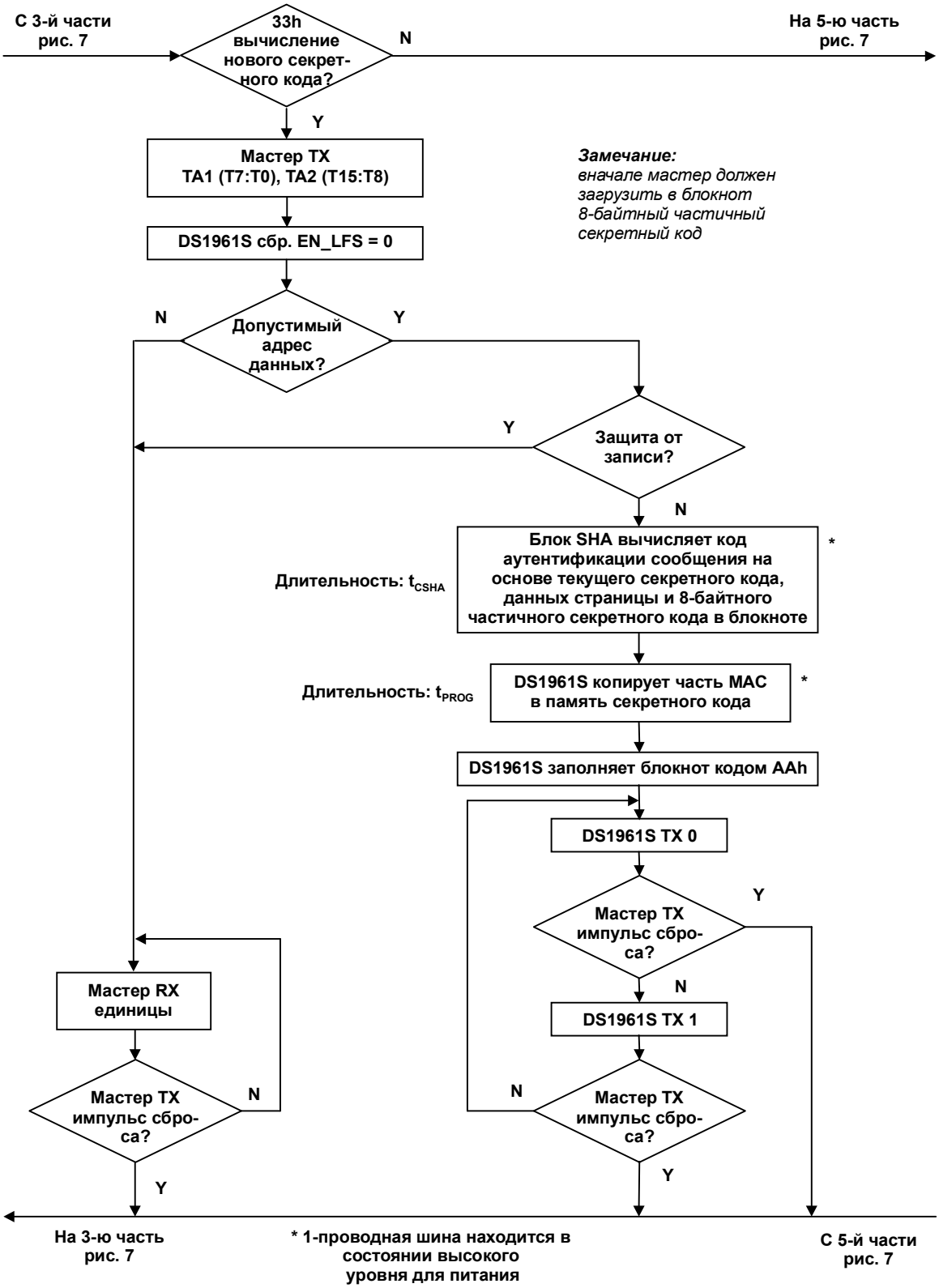


Рис. 7-5. БЛОК-СХЕМА ФУНКЦИЙ ПАМЯТИ И SHA (продолжение)

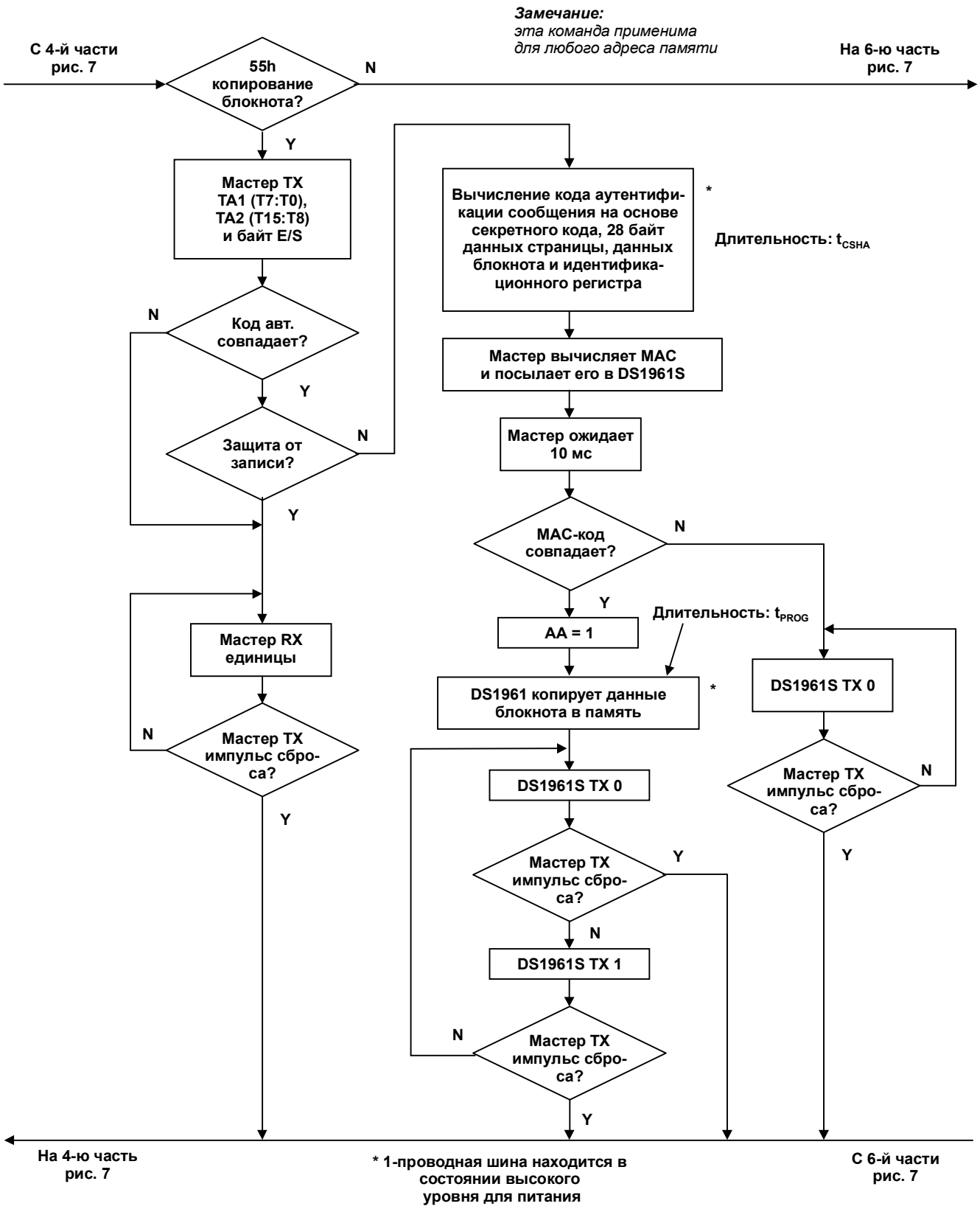


Рис. 7-6. БЛОК-СХЕМА ФУНКЦИЙ ПАМЯТИ И SHA (продолжение)

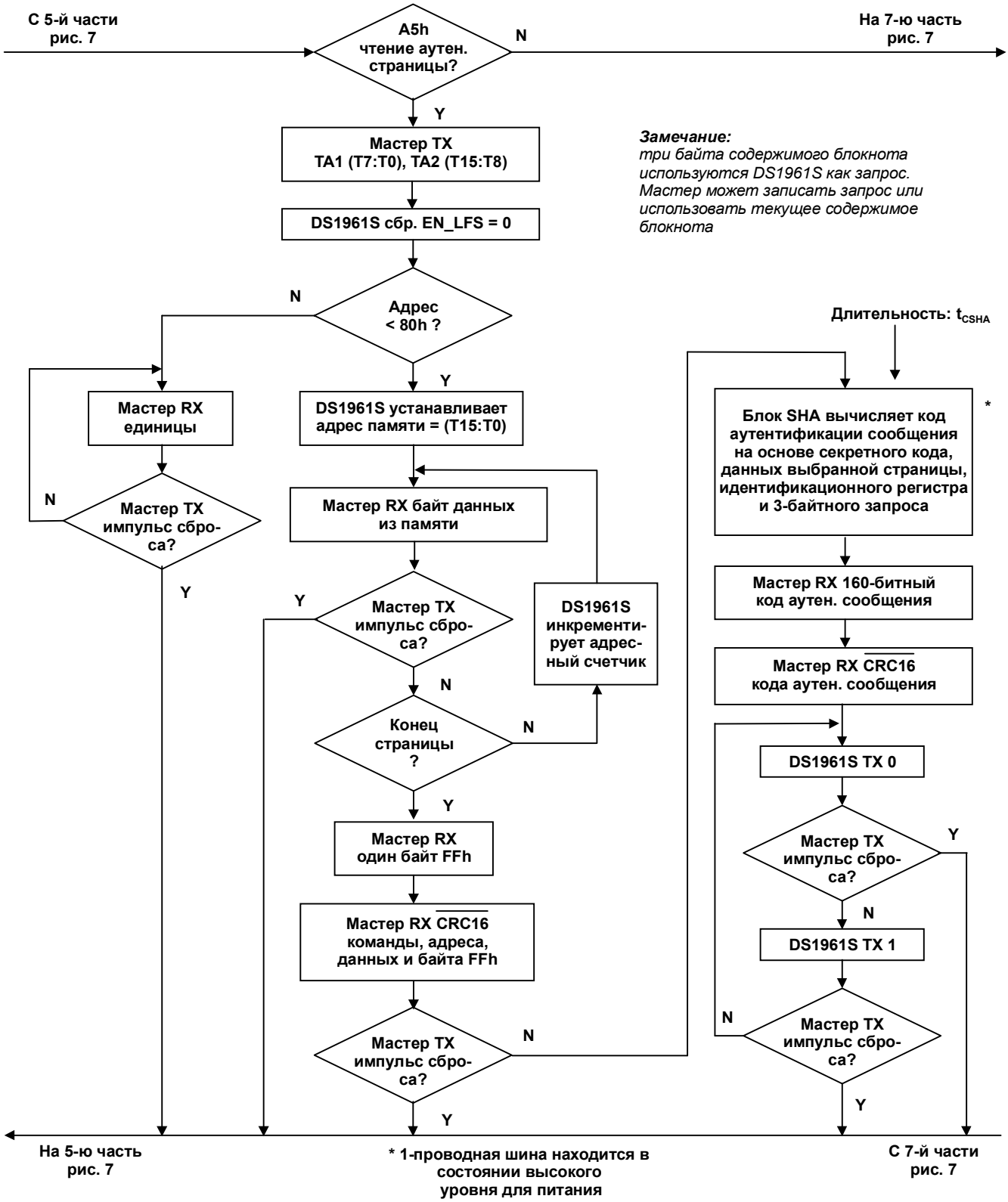


Рис. 7-7. БЛОК-СХЕМА ФУНКЦИЙ ПАМЯТИ И SHA (продолжение)

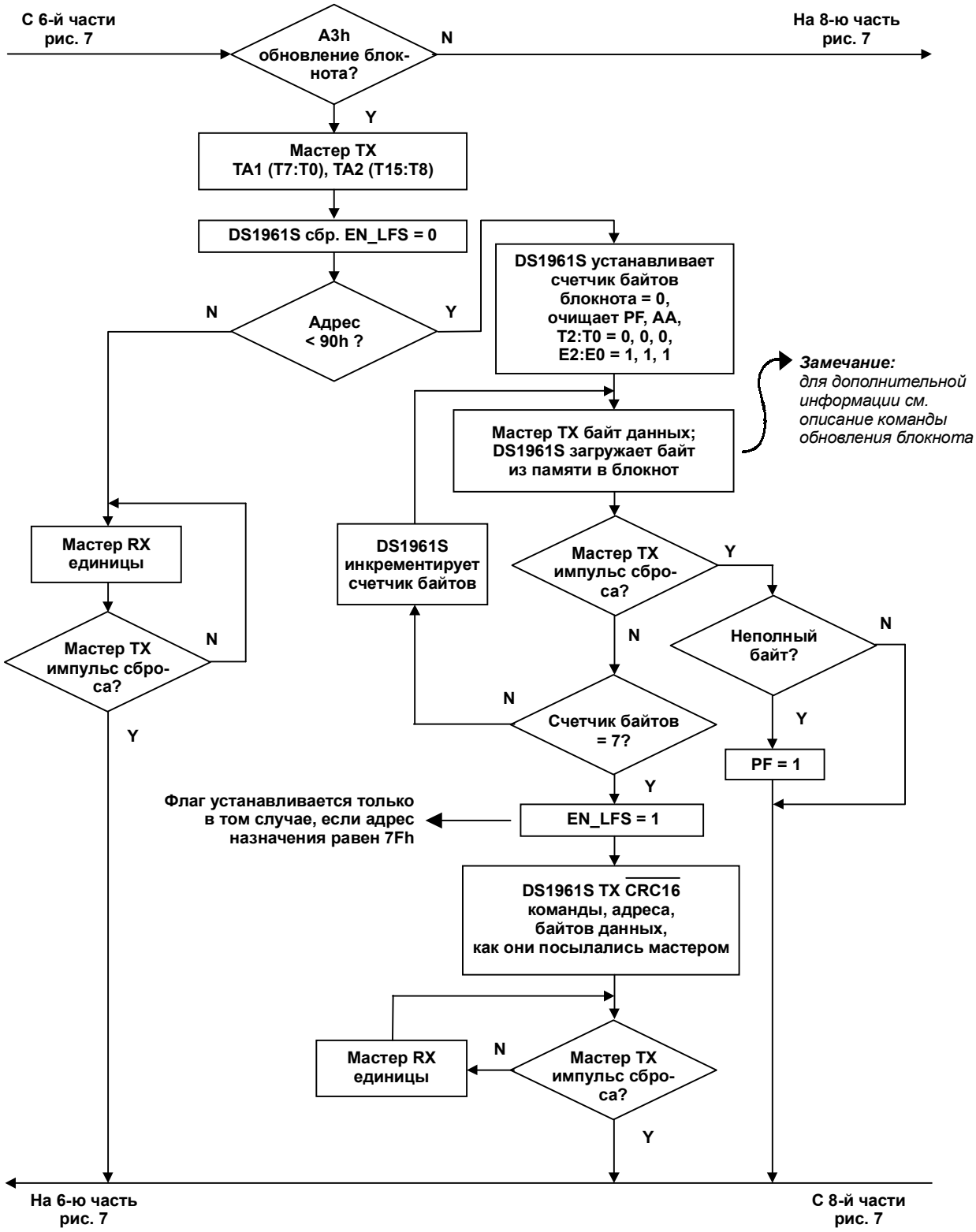
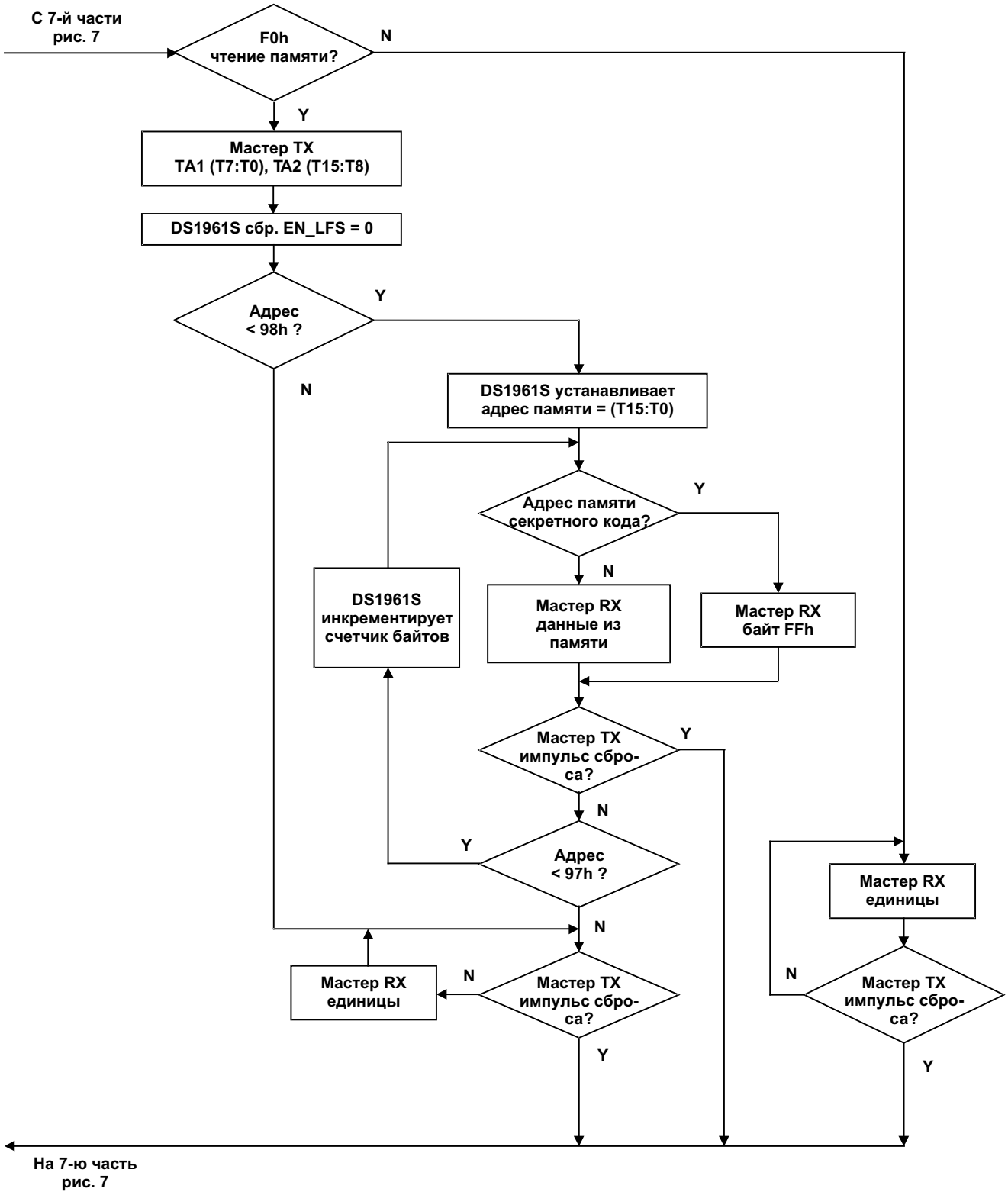


Рис. 7-8. БЛОК-СХЕМА ФУНКЦИЙ ПАМЯТИ И SHA (продолжение)



Вычисление следующего секретного кода [33h]

Некоторые приложения могут потребовать более высокого уровня безопасности, чем обеспечивает одиночный, непосредственно записанный секретный код. Для повышения уровня безопасности DS1961S имеет возможность вычисления нового секретного кода, основываясь на текущем секретном коде, содержимом выбранной страницы памяти и на частичном секретном коде, который включает все данные блокнота. Для установки вычисленного секретного кода мастер выдает команду вычисления следующего секретного кода, которая активизирует 512-битный блок SHA-1, если память секретного кода не защищена от записи. Таблица 1 показывает, как разные компоненты данных образуют входные данные для блока SHA, и как часть результата SHA загружается в память секретного кода. Алгоритм вычислений SHA рассматривается в данном документе ниже. Команда вычисления следующего секретного кода может выполняться так часто, как это требуется для повышения уровня безопасности. Для успешного вычисления следующего секретного кода мастеру шины не требуется знать текущий секретный код.

Таблица 1. ВХОДНЫЕ ДАННЫЕ SHA-1 ДЛЯ КОМАНДЫ ВЫЧИСЛЕНИЯ СЛЕДУЮЩЕГО СЕКРЕТНОГО КОДА

M0[31:24] = (SS+0)	M0[23:16] = (SS+1)	M0[15:8] = (SS+2)	M0[7:0] = (SS+3)
M1[31:24] = (PP+0)	M1[23:16] = (PP+1)	M1[15:8] = (PP+2)	M1[7:0] = (PP+3)
M2[31:24] = (PP+4)	M2[23:16] = (PP+5)	M2[15:8] = (PP+6)	M2[7:0] = (PP+7)
M3[31:24] = (PP+8)	M3[23:16] = (PP+9)	M3[15:8] = (PP+10)	M3[7:0] = (PP+11)
M4[31:24] = (PP+12)	M4[23:16] = (PP+13)	M4[15:8] = (PP+14)	M4[7:0] = (PP+15)
M5[31:24] = (PP+16)	M5[23:16] = (PP+17)	M5[15:8] = (PP+18)	M5[7:0] = (PP+19)
M6[31:24] = (PP+20)	M6[23:16] = (PP+21)	M6[15:8] = (PP+22)	M6[7:0] = (PP+23)
M7[31:24] = (PP+24)	M7[23:16] = (PP+25)	M7[15:8] = (PP+26)	M7[7:0] = (PP+27)
M8[31:24] = (PP+28)	M8[23:16] = (PP+29)	M8[15:8] = (PP+30)	M8[7:0] = (PP+31)
M9[31:24] = FFh	M9[23:16] = FFh	M9[15:8] = FFh	M9[7:0] = FFh
M10[31:24] = MPX	M10[23:16] = (SP+1)	M10[15:8] = (SP+2)	M10[7:0] = (SP+3)
M11[31:24] = (SP+4)	M11[23:16] = (SP+5)	M11[15:8] = (SP+6)	M11[7:0] = (SP+7)
M12[31:24] = (SS+4)	M12[23:16] = (SS+5)	M12[15:8] = (SS+6)	M12[7:0] = (SS+7)
M13[31:24] = FFh	M13[23:16] = FFh	M13[15:8] = FFh	M13[7:0] = 80h
M14[31:24] = 00h	M14[23:16] = 00h	M14[15:8] = 00h	M14[7:0] = 00h
M15[31:24] = 00h	M15[23:16] = 00h	M15[15:8] = 01h	M15[7:0] = B8h

РЕЗУЛЬТАТ ВЫЧИСЛЕНИЯ СЛЕДУЮЩЕГО СЕКРЕТНОГО КОДА

(SS+0) := E[7:0]	(SS+1) := E[15:8]	(SS+2) := E[23:16]	(SS+3) := E[31:24]
(SS+4) := D[7:0]	(SS+5) := D[15:8]	(SS+6) := D[23:16]	(SS+7) := D[31:24]

Условные обозначения

Mt	**Входной буфер блока SHA** $0 \leq t \leq 15$; 32-битные слова
(SS + N)	**Байт N секретного кода; секретный код начинается с адреса 0080h** (См. карту памяти)
(PP + N)	**Байт N страницы памяти; страницы памяти начинаются по адресам 0000h, 0020h, 0040h и 0060h** (См. карту памяти)
(SP + N)	**Байт N блокнота**
MPX	MPX[7] = 0; MPX[6] = 0; MPX[5:0] = (SP + 0)[5:0]
D, E	32-битные слова, часть 160-битного результата SHA

После выдачи команды вычисления следующего секретного кода мастер должен передать 2-байтный адрес назначения для выбора страницы памяти, содержимое которой составит 256 бит

входных данных SHA. После приема адреса назначения (TA1 и TA2) DS1961S очищает флаг EN_LFS. Пять младших битов адреса TA1 игнорируются, так как требуется только адрес страницы. Если переданный мастером адрес назначения является допустимым (т.е. лежит в диапазоне 0000h – 007Fh), и память секретного кода не защищена от записи, запускается блок SHA. Мастер должен выждать время t_{CSHA}, пока вычисляется новый секретный код. Сразу после этой задержки мастер должен выждать время t_{PROG}, пока новый секретный код копируется в память. Во время вычисления и копирования нового секретного кода напряжение на 1-проводной шине не должно опускаться ниже 2,8В. Если копирование прошло успешно, DS1961S заполняет блокнот кодом AAh. Если блок SHA не запускался ввиду некорректного адреса назначения или наличия у секретного кода защиты от записи, содержимое блокнота не изменяется. По окончании задержки записи мастер должен прочитать хотя бы один байт. Считанное значение AAh говорит о том, что копирование прошло успешно. Значение FFh говорит об ошибке копирования в результате некорректного адреса назначения или наличия у секретного кода защиты от записи.

Так как содержимое блокнота используется в качестве частичного секретного кода, мастер заполняет блокнот известной 8-байтной последовательностью, используя команду записи блокнота перед выдачей команды вычисления следующего секретного кода. Иначе значение нового секретного кода будет зависеть от данных, которые случайно остались в блокноте от предыдущих команд.

Копирование блокнота [55h]

Память данных DS1961S может быть считана без каких-либо ограничений. Однако выполнение команды копирования блокнота для записи новых данных в память или в страницу регистров требует знания секретного кода и возможности проведения вычислений SHA-1 для генерации 160-битного MAC-кода, который разрешит передачу данных из блокнота в память. Мастер может осуществить вычисление MAC-кода программно, или использовать для этого сопроцессор DS1963S. Подход с применением сопроцессора имеет преимущество, так как секретный код остается спрятанным внутри iButton сопроцессора. Последовательность, в которой полученный MAC-код должен передаваться в DS1961S, показана в таблице 2. Таблицы 3A и 3B показывают, как разные компоненты данных образуют входные данные для блока SHA. Алгоритм вычислений SHA рассматривается в данном документе ниже.

Таблица 2. ПОСЛЕДОВАТЕЛЬНОСТЬ ПЕРЕДАЧИ КОДА АУТЕНТИФИКАЦИИ СООБЩЕНИЯ

E[31:24]	E[23:16]	E[15:8]	E[7:0]
D[31:24]	D[23:16]	D[15:8]	D[7:0]
C[31:24]	C[23:16]	C[15:8]	C[7:0]
B[31:24]	B[23:16]	B[15:8]	B[7:0]
A[31:24]	A[23:16]	A[15:8]	A[7:0]

Направление сдвига →

В процессе пересылки первым передается младший бит. Пересылка начинается с регистра E.

После выдачи команды копирования блокнота, мастер должен передать 3-байтную последовательность авторизации, которая должна быть непосредственно перед этим получена с помощью команды чтения блокнота. Эта 3-байтная последовательность должна точно совпадать с данными, которые содержатся в трех адресных регистрах (TA1, TA2, E/S, в этом порядке). Если последовательность авторизации совпадает, и память назначения не защищена от записи, DS1961S запускает блок SHA для вычисления 160-битного MAC-кода на основе текущего секретного кода, всех данных блокнота, первых 28 байт адресованной страницы памяти и первых семи байт идентификационного регистра (байт с адресом 0097h не используется; см. таблицу 3A). Длительность вычислений равна t_{CSHA}, в это время напряжение на 1-проводной шине не должно

опускаться ниже 2,8В. Одновременно мастер вычисляет MAC-код на основе тех же данных, и по истечении промежутка времени t_{CSHA} передает его в DS1961S как доказательство своей авторизации для записи в EEPROM. После этого мастер должен выждать время t_{PROG}, в течение которого напряжение на 1-проводной шине не должно опускаться ниже 2,8В. Если MAC-код, сгенерированный DS1961S, совпадает с MAC-кодом, который вычислил мастер, DS1961S устанавливает свой флаг AA и целиком копирует содержимое блокнота в EEPROM данных. По окончании задержки копирования мастер должен прочитать хотя бы один байт. Считанное значение AAh говорит о том, что копирование прошло успешно. Значение 00h говорит об ошибке копирования в результате несовпадения вычисленного MAC-кода с MAC-кодом, присланным мастером. Значение FFh говорит об ошибке копирования в результате наличия защиты от записи или некорректной последовательности авторизации.

Таблица 3a. ВХОДНЫЕ ДАННЫЕ SHA-1 ДЛЯ КОМАНДЫ КОПИРОВАНИЯ БЛОКНОТА ПРИ ЗАПИСИ В СТРАНИЦУ ПАМЯТИ ДАННЫХ

M0[31:24] = (SS+0)	M0[23:16] = (SS+1)	M0[15:8] = (SS+2)	M0[7:0] = (SS+3)
M1[31:24] = (PP+0)	M1[23:16] = (PP+1)	M1[15:8] = (PP+2)	M1[7:0] = (PP+3)
M2[31:24] = (PP+4)	M2[23:16] = (PP+5)	M2[15:8] = (PP+6)	M2[7:0] = (PP+7)
M3[31:24] = (PP+8)	M3[23:16] = (PP+9)	M3[15:8] = (PP+10)	M3[7:0] = (PP+11)
M4[31:24] = (PP+12)	M4[23:16] = (PP+13)	M4[15:8] = (PP+14)	M4[7:0] = (PP+15)
M5[31:24] = (PP+16)	M5[23:16] = (PP+17)	M5[15:8] = (PP+18)	M5[7:0] = (PP+19)
M6[31:24] = (PP+20)	M6[23:16] = (PP+21)	M6[15:8] = (PP+22)	M6[7:0] = (PP+23)
M7[31:24] = (PP+24)	M7[23:16] = (PP+25)	M7[15:8] = (PP+26)	M7[7:0] = (PP+27)
M8[31:24] = (SP+0)	M8[23:16] = (SP+1)	M8[15:8] = (SP+2)	M8[7:0] = (SP+3)
M9[31:24] = (SP+4)	M9[23:16] = (SP+5)	M9[15:8] = (SP+6)	M9[7:0] = (SP+7)
M10[31:24] = MP	M10[23:16] = (ID+0)	M10[15:8] = (ID+1)	M10[7:0] = (ID+2)
M11[31:24] = (ID+3)	M11[23:16] = (ID+4)	M11[15:8] = (ID+5)	M11[7:0] = (ID+6)
M12[31:24] = (SS+4)	M12[23:16] = (SS+5)	M12[15:8] = (SS+6)	M12[7:0] = (SS+7)
M13[31:24] = FFh	M13[23:16] = FFh	M13[15:8] = FFh	M13[7:0] = 80h
M14[31:24] = 00h	M14[23:16] = 00h	M14[15:8] = 00h	M14[7:0] = 00h
M15[31:24] = 00h	M15[23:16] = 00h	M15[15:8] = 01h	M15[7:0] = B8h

Условные обозначения

Mt	**Входной буфер блока SHA** $0 \leq t \leq 15$; 32-битные слова
(SS + N)	**Байт N секретного кода; секретный код начинается с адреса 0080h** (См. карту памяти)
(PP + N)	**Байт N страницы памяти; страницы памяти начинаются по адресам 0000h, 0020h, 0040h и 0060h** (См. карту памяти)
(SP + N)	**Байт N блокнота**
MP	MP[7:3] = 00000b, MP[2:0] = T7:T5
(ID+N)	**Байт N идентификационного регистра** Последний байт идентификационного регистра не используется

Особое внимание требуется при копировании данных в страницу регистров. Для предотвращения случайной блокировки регистра специальной функции или байта пользователя рекомендуется сначала считать страницу регистров, а затем записать ее со всеми необходимыми изменениями в блокнот. При копировании данных в страницу регистров (или в память секретного кода с помощью команды копирования блокнота) входными данными M1..M7 для блока SHA является текущий секретный код (M1, M2), текущее содержимое страницы регистров (M3, M4), все содержимое идентификационного регистра (M5, M6) и 4 байта FFh (M7), как показано в таблице 3B. Как следствие, при использовании DS1963S в качестве сопроцессора для вычисления MAC-

кода, разрешающего пересылку данных из блокнота в страницу регистров, секретный код должен быть использован как данные страницы. Это препятствует использованию частичного (вычисленного) секретного кода, если требуется запись в страницу регистров. При практическом использовании DS1961S для проведения электронных платежей, частичные секретные коды менее желательны, чем постоянные, позволяющие защитить от записи память секретного кода или другие области устройства.

Таблица 3b. ВХОДНЫЕ ДАННЫЕ SHA-1 ДЛЯ КОМАНДЫ КОПИРОВАНИЯ БЛОКНОТА ПРИ ЗАПИСИ В СТРАНИЦУ РЕГИСТРОВ ИЛИ ПАМЯТЬ СЕКРЕТНОГО КОДА

M0[31:24] = (SS+0)	M0[23:16] = (SS+1)	M0[15:8] = (SS+2)	M0[7:0] = (SS+3)
M1[31:24] = (SS+0)	M1[23:16] = (SS+1)	M1[15:8] = (SS+2)	M1[7:0] = (SS+3)
M2[31:24] = (SS+4)	M2[23:16] = (SS+5)	M2[15:8] = (SS+6)	M2[7:0] = (SS+7)
M3[31:24] = (RP+0)	M3[23:16] = (RP+1)	M3[15:8] = (RP+2)	M3[7:0] = (RP+3)
M4[31:24] = (RP+4)	M4[23:16] = (RP+5)	M4[15:8] = (RP+6)	M4[7:0] = (RP+7)
M5[31:24] = (ID+0)	M5[23:16] = (ID+1)	M5[15:8] = (ID+2)	M5[7:0] = (ID+3)
M6[31:24] = (ID+4)	M6[23:16] = (ID+5)	M6[15:8] = (ID+6)	M6[7:0] = (ID+7)
M7[31:24] = FFh	M7[23:16] = FFh	M7[15:8] = FFh	M7[7:0] = FFh
M8[31:24] = (SP+0)	M8[23:16] = (SP+1)	M8[15:8] = (SP+2)	M8[7:0] = (SP+3)
M9[31:24] = (SP+4)	M9[23:16] = (SP+5)	M9[15:8] = (SP+6)	M9[7:0] = (SP+7)
M10[31:24] = MP	M10[23:16] = (ID+0)	M10[15:8] = (ID+1)	M10[7:0] = (ID+2)
M11[31:24] = (ID+3)	M11[23:16] = (ID+4)	M11[15:8] = (ID+5)	M11[7:0] = (ID+6)
M12[31:24] = (SS+4)	M12[23:16] = (SS+5)	M12[15:8] = (SS+6)	M12[7:0] = (SS+7)
M13[31:24] = FFh	M13[23:16] = FFh	M13[15:8] = FFh	M13[7:0] = 80h
M14[31:24] = 00h	M14[23:16] = 00h	M14[15:8] = 00h	M14[7:0] = 00h
M15[31:24] = 00h	M15[23:16] = 00h	M15[15:8] = 01h	M15[7:0] = B8h

Условные обозначения

Mt	**Входной буфер блока SHA** $0 \leq t \leq 15$; 32-битные слова
(SS + N)	**Байт N секретного кода; секретный код начинается с адреса 0080h** (См. карту памяти)
(RP + N)	**Байт N страницы регистров; страница начинается по адресу 0088h** (См. карту памяти)
(SP + N)	**Байт N блокнота**
MP	MP[7:0] = 04h
(ID+N)	**Байт N идентификационного регистра**

Чтение аутентифицированной страницы [A5h]

Команда чтения аутентифицированной страницы позволяет мастеру получить данные полной (или части) страницы памяти плюс MAC-код. MAC-код позволяет мастеру определить, является ли секретный код, сохраненный в DS1961S, действительным для данного приложения. DS1961S вычисляет MAC-код на основе собственного секретного кода, всех данных выбранной страницы памяти, первых семи байт идентификационного регистра и 3-байтного запроса, который должен быть записан мастером в блокнот перед использованием команды чтения аутентифицированной страницы. Для того чтобы сделать это, мастер может использовать команду записи блокнота с любым адресом назначения в памяти данных. Байты запроса должны размещаться в 5-м, 6-м и 7-м байтах блокнота. Как вариант, мастер может использовать в качестве запроса данные, которые остались в блокноте от предыдущей команды. MAC-код передается таким же образом, как и для команды копирования блокнота (таблица 2), только данные теперь передаются от DS1961S мастеру. Входные данные блока SHA, которые используются во время выполнения команды чтения аутентифицированной страницы, показаны в таблице 4. После передачи мастером кода команды и адреса назначения (TA1 и TA2) DS1961S первым делом очищает флаг EN_LFS. Если

адрес назначения является правильным (<0080h), мастер принимает данные страницы, начиная с адреса назначения и до конца страницы данных, затем один байт FFh, затем инвертированное значение CRC для кода команды, адреса назначения, переданной страницы данных и байта FFh. Если адрес назначения является ошибочным (≥0080h), мастер принимает вместо данных страницы байты FFh. Сразу после приема CRC мастер ожидает время t_{CSHA}, в течение которого напряжение на 1-проводной шине не должно опускаться ниже 2,8В. В это время блок SHA вычисляет код аутентификации сообщения на основании секретного кода, всех 32 байтов выбранной страницы, регистрационного номера устройства (без CRC) и 3-байтного запроса. После этого мастер считывает 160-битный MAC-код, за которым следует инвертированное значение CRC, что всегда необходимо при безопасных пересылках данных. Если мастер продолжает чтение после приема CRC, он принимает AAh.

Таблица 4. ВХОДНЫЕ ДАННЫЕ SHA-1 ДЛЯ КОМАНДЫ ЧТЕНИЯ АУТЕНТИФИЦИРОВАННОЙ СТРАНИЦЫ

M0[31:24] = (SS+0)	M0[23:16] = (SS+1)	M0[15:8] = (SS+2)	M0[7:0] = (SS+3)
M1[31:24] = (PP+0)	M1[23:16] = (PP+1)	M1[15:8] = (PP+2)	M1[7:0] = (PP+3)
M2[31:24] = (PP+4)	M2[23:16] = (PP+5)	M2[15:8] = (PP+6)	M2[7:0] = (PP+7)
M3[31:24] = (PP+8)	M3[23:16] = (PP+9)	M3[15:8] = (PP+10)	M3[7:0] = (PP+11)
M4[31:24] = (PP+12)	M4[23:16] = (PP+13)	M4[15:8] = (PP+14)	M4[7:0] = (PP+15)
M5[31:24] = (PP+16)	M5[23:16] = (PP+17)	M5[15:8] = (PP+18)	M5[7:0] = (PP+19)
M6[31:24] = (PP+20)	M6[23:16] = (PP+21)	M6[15:8] = (PP+22)	M6[7:0] = (PP+23)
M7[31:24] = (PP+24)	M7[23:16] = (PP+25)	M7[15:8] = (PP+26)	M7[7:0] = (PP+27)
M8[31:24] = (PP+28)	M8[23:16] = (PP+29)	M8[15:8] = (PP+30)	M8[7:0] = (PP+31)
M9[31:24] = FFh	M9[23:16] = FFh	M9[15:8] = FFh	M9[7:0] = FFh
M10[31:24] = MP	M10[23:16] = (ID+0)	M10[15:8] = (ID+1)	M10[7:0] = (ID+2)
M11[31:24] = (ID+3)	M11[23:16] = (ID+4)	M11[15:8] = (ID+5)	M11[7:0] = (ID+6)
M12[31:24] = (SS+4)	M12[23:16] = (SS+5)	M12[15:8] = (SS+6)	M12[7:0] = (SS+7)
M13[31:24] = (SP+4)	M13[23:16] = (SP+5)	M13[15:8] = (SP+6)	M13[7:0] = 80h
M14[31:24] = 00h	M14[23:16] = 00h	M14[15:8] = 00h	M14[7:0] = 00h
M15[31:24] = 00h	M15[23:16] = 00h	M15[15:8] = 01h	M15[7:0] = B8h

Условные обозначения

Mt	**Входной буфер блока SHA** $0 \leq t \leq 15$; 32-битные слова
(SS + N)	**Байт N секретного кода; секретный код начинается с адреса 0080h** (См. карту памяти)
(PP + N)	**Байт N страницы памяти; страницы памяти начинаются по адресам 0000h, 0020h, 0040h и 0060h** (См. карту памяти)
(SP + N)	**Байт N блокнота**
MP	MP[7:3] = 01000b, MP[2:0] = T7:T5
(ID+N)	**Байт N идентификационного регистра** Последний байт идентификационного регистра не используется

Обновление блокнота [A3h]

Команда обновления блокнота загружает данные из памяти в блокнот и устанавливает флаг EN_LFS. Этот флаг позволяет использование команды загрузки первого секретного кода для перезаписи данных, которые были только что прочитаны из памяти, не прибегая к вычислению MAC-кода.

Блок-схемы команд обновления блокнота и записи блокнота очень похожи. Если адрес назначения лежит в диапазоне 0000h – 007Fh, существуют два основных отличия. 1) Для команды обновления блокнота байты данных, которые мастер посылает вслед за адресом назначения, отбрасываются;

взамен этого блокнот загружается неизмененными данными из области памяти, согласно адресу назначения, даже если данная страница памяти находится в режиме EPROM. 2) После того, как мастер передаст восемь байт (которые не используются), флаг EN_LFS устанавливается в 1. Флаг EN_LFS сбрасывается после приема TA1 и TA2 во время выполнения команды записи блокнота, вычисления следующего секретного кода, чтения аутентифицированной страницы, обновления блокнота, чтения памяти или по сбросу при включении питания, потому что эти команды могут изменить адрес назначения и/или данные в блокноте.

Если передан адрес назначения 0080h – 008Fh, команда обновления блокнота ведет себя точно так же, как и команда записи блокнота. Это защищает секретный код от считывания последующей командой чтения блокнота.

Чтение памяти [F0h]

Команда чтения памяти может использоваться для чтения всех областей памяти, за исключением памяти секретного кода. Попытка чтения памяти секретного кода приведет к считыванию байтов FFh взамен настоящих данных. После передачи мастером кода команды и адреса назначения (TA1 и TA2) DS1961S первым делом очищает флаг EN_LFS. Если адрес назначения является правильным, мастер считывает данные, начиная с адреса назначения. Он может продолжать считывание вплоть до адреса 0097h. Если мастер продолжит чтение дальше, то получит одни логические единицы. Важно представлять, что регистры адреса назначения указывают на последний считанный байт. Байт конечного смещения/состояния данных и блокнот не изменяются.

DS1961S имеет аппаратные средства для осуществления безошибочной записи в секцию памяти. Для безопасного чтения данных и одновременного повышения скорости обмена в 1-проводных системах рекомендуется организовывать данные в пакеты размером в одну страницу памяти. Такой пакет обычно содержит вычисленную мастером 16-битную CRC, которая обеспечивает быстрый и безошибочный обмен данными, исключая необходимость многократного чтения страницы для определения того, являются ли принятые данные правильными (см. *Application Note 114*, где приведена рекомендуемая файловая структура, называемая также форматом TMEX).

АЛГОРИТМ ВЫЧИСЛЕНИЯ SHA-1

Данное описание алгоритма вычисления SHA является адаптированным вариантом документа под названием «Secure Hash Standard SHA-1», который можно найти на сайте NIST (www.itl.gov/fipspubs/fip180-1.htm). Алгоритм использует в качестве входных данных шестнадцать 32-битных слов M_t ($0 \le t \le 15$), как показано в таблицах 1, 3A, 3B и 4 для команд вычисления следующего секретного кода, копирования блокнота и чтения аутентифицированной страницы соответственно. В вычислении SHA участвуют две последовательности из восьмидесяти 32-битных слов, называемые W_t ($0 \le t \le 79$) и K_t ($0 \le t \le 79$), Булева функция f_t (B, C, D) ($0 \le t \le 79$), где B, C и D являются 32-битными словами, и еще три 32-битных слова, называемых A, E и TMP. Для вычисления SHA требуются следующие операции: арифметическое сложение без переноса («+»), логическая инверсия («\»), исключающее ИЛИ («⊕»), логическое И («∧»), логическое ИЛИ («∨»), присвоение («:=») и циклический сдвиг 32-битного слова. Выражение «$S^n(X)$» означает циклический сдвиг X на n разрядов влево, где X является 32-битным словом.

Функция f_t определена следующим образом:

$$f_t(B, C, D) = \begin{cases} (B \wedge C) \vee ((B\backslash) \wedge D) & (0 \le t \le 19) \\ B \oplus C \oplus D & (20 \le t \le 39) \\ (B \wedge C) \vee (B \wedge D) \vee (C \wedge D) & (40 \le t \le 59) \\ B \oplus C \oplus D & (60 \le t \le 79) \end{cases}$$

Последовательность W_t ($0 \le t \le 79$) определена следующим образом:

W_t := M_t ($0 \le t \le 15$)

$S^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16})$ ($16 \le t \le 79$)

Последовательность K_t ($0 \le t \le 79$) определена следующим образом:

K_t := 5A827999h ($0 \le t \le 19$)

6ED9EBA1h ($20 \le t \le 39$)

8F1BBCDCh ($40 \le t \le 59$)

CA62C1D6h ($60 \le t \le 79$)

Переменные A, B, C, D, E инициализированы следующими значениями:

A := 67452301h

B := EFCDAB89h

C := 98BADCFEh

D := 10325476h

E := C3D2E1F0h

Выходной 160-битный MAC-код представляет собой объединение переменных A, B, C, D и E после циклического выполнения следующего набора операций для t = 0 .. 79 (без учета переноса):

TMP := $S^5(A) + f_t(B, C, D) + W_t + K_t + E$

E := D

D := C

C := $S^{30}(B)$

B := A

A := TMP

Мастер может считать MAC-код в свой регистр по команде чтения аутентифицированной страницы, последовательность бит показана в таблице 3. Для команды копирования блокнота требуется такая же последовательность бит, только в этом случае MAC-код должен вычислить мастер и передать его в DS1961S. Командой вычисления следующего секретного кода MAC-код не используется. Взамен этого, в память секретного кода непосредственно копируется содержимое регистров D и E, использующихся в вычислениях SHA, как показано в таблице 1.

1-ПРОВОДНАЯ ШИНА

1-проводная шина представляет собой систему, в которой имеется один мастер шины и одно или несколько подчиненных устройств. Во всех случаях DS1961S является подчиненным устройством. Мастером шины обычно является микроконтроллер. Для небольших систем сигналы 1-проводного протокола могут генерироваться программно, используя один вывод порта микроконтроллера. Для более крупных систем рекомендуется использовать микросхему драйвера однопроводной линии DS2480B или адаптеры для последовательного порта, построенные на основе этой микросхемы (DS9097U). Это упрощает аппаратную часть и избавляет микропроцессор от необходимости выполнения операций реального времени.

Обсуждение 1-проводной шины можно разбить на три части: аппаратная конфигурация, последовательность пересылки и 1-проводные сигналы (типы сигналов и их временные параметры). Протокол 1-проводной шины определяет пересылки с помощью понятия специальных

временных интервалов, которые начинаются спадом импульса синхронизации, выдаваемого мастером. Более детальное описание протокола приведено в главе 4 книги *«Book of DS19xx iButton Standards»*.

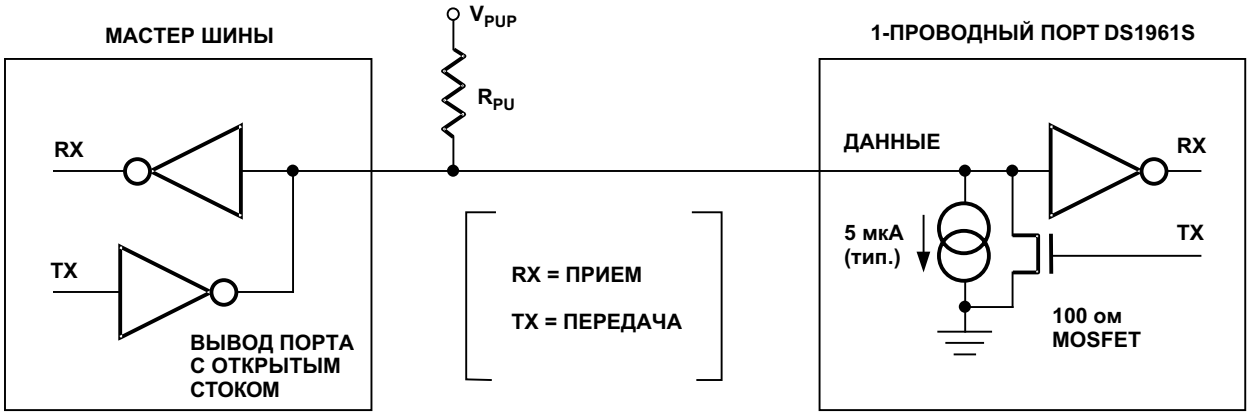
АППАРАТНАЯ КОНФИГУРАЦИЯ

По определению 1-проводная шина имеет только одну линию; поэтому важно обеспечить для каждого устройства, подключенного к шине, возможность в соответствующие моменты времени ею управлять. Для этого каждое устройство, подключенное к 1-проводной шине, должно иметь выход с открытым стоком или с тремя состояниями. DS1961S имеет выход с открытым стоком, его внутренняя схема эквивалентна показанной на рис. 8.

Многоточечная шина представляет собой 1-проводную шину, к которой подключено несколько подчиненных устройств. В стандартном режиме передача данных по 1-проводной шине идет со скоростью максимум 16,3 Кбит в секунду. При включении ускоренного режима скорость может быть увеличена до 142 Кбит в секунду. DS1961S не гарантирует полной совместимости со стандартом iButton, так как для DS1961S максимальная скорость обмена составляет 14,1 Кбит в секунду в стандартном режиме и 125 Кбит в секунду в ускоренном режиме. Для успешного выполнения команд функций памяти и SHA на любой скорости DS1961S требует подтягивающего резистора с номиналом максимум 2,2 Ком. Если обмен ведется одновременно с несколькими DS1961S, например, для записи в несколько устройств одинакового секретного кода, то в те моменты времени, когда идет копирование блокнота в EEPROM, подтягивающий резистор должен быть закорочен.

В состоянии покоя на линии 1-проводной шины присутствует высокий уровень. Если по каким-либо причинам пересылка должна быть приостановлена, линию нужно оставить в состоянии покоя, чтобы впоследствии пересылка могла быть продолжена. Если этого не сделать и оставить линию в состоянии низкого уровня дольше, чем на 16 мкс при повышенной скорости, или 120 мкс при обычной скорости, одно или несколько устройств на шине могут быть сброшены. Для DS1961S при повышенной скорости линия не должна находиться в состоянии низкого уровня дольше, чем 15,2 мкс, чтобы быть уверенным в том, что ни одно устройство не будет сброшено. Несмотря на такую неполную совместимость, DS1961S корректно работает в паре с драйвером 1-проводной шины DS2480B и с адаптерами последовательного порта, которые построены на основе этой микросхемы.

Рис. 8. АППАРАТНАЯ КОНФИГУРАЦИЯ



ПОСЛЕДОВАТЕЛЬНОСТЬ ПЕРЕСЫЛКИ

Последовательность действий для доступа к DS1961S через 1-проводный порт должна быть следующей:

- Инициализация
- Команда функций ПЗУ
- Команда функций памяти или SHA
- Передача данных

ИНИЦИАЛИЗАЦИЯ

Все пересылки по 1-проводной шине начинаются с последовательности инициализации. Последовательность инициализации содержит импульс сброса, выдаваемый мастером шины, за которым следует импульс (импульсы) присутствия, передаваемый подчиненным устройством (устройствами).

Импульс присутствия говорит мастеру шины о том, что подчиненное устройство представлено на шине и оно готово к работе. Более подробную информацию можно найти в разделе *«Сигналы 1-проводной шины»*.

КОМАНДЫ ФУНКЦИЙ ПЗУ

Когда мастер шины обнаруживает импульс присутствия, он может подать одну из семи команд функций ПЗУ, которые поддерживаются DS1961S. Все команды функций ПЗУ имеют длину 8 бит. Список этих команд приведен ниже (см. блок-схему на рис. 9).

Чтение ПЗУ [33h]

Эта команда позволяет мастеру шины считать из DS1961S 8-битный код семейства, уникальный 48-битный серийный номер и 8-битную CRC. Команда может быть использована только в том случае, когда на шине присутствует всего одно подчиненное устройство. Если имеется несколько подчиненных устройств, то произойдет искажение данных, так как все они попытаются одновременно передать данные (открытые стоки реализуют функцию «монтажное И»). В результате принятый мастером код семейства и 48-битный серийный номер будет неправильным.

Сравнение ПЗУ [55h]

Команда сравнения ПЗУ, за которой следует 64-битный регистрационный номер, позволяет мастеру шины адресовать отдельное устройство на многоточечной шине. Только тот экземпляр DS1961S, содержимое ПЗУ которого полностью совпадет с переданным мастером 64-битным регистрационным номером, будет отвечать на последующие команды функций памяти или SHA. Все остальные подчиненные устройства будут ожидать импульса сброса. Эта команда может использоваться при наличии на шине как одного, так и нескольких устройств.

Рис. 9-1. БЛОК-СХЕМА ФУНКЦИЙ ПЗУ

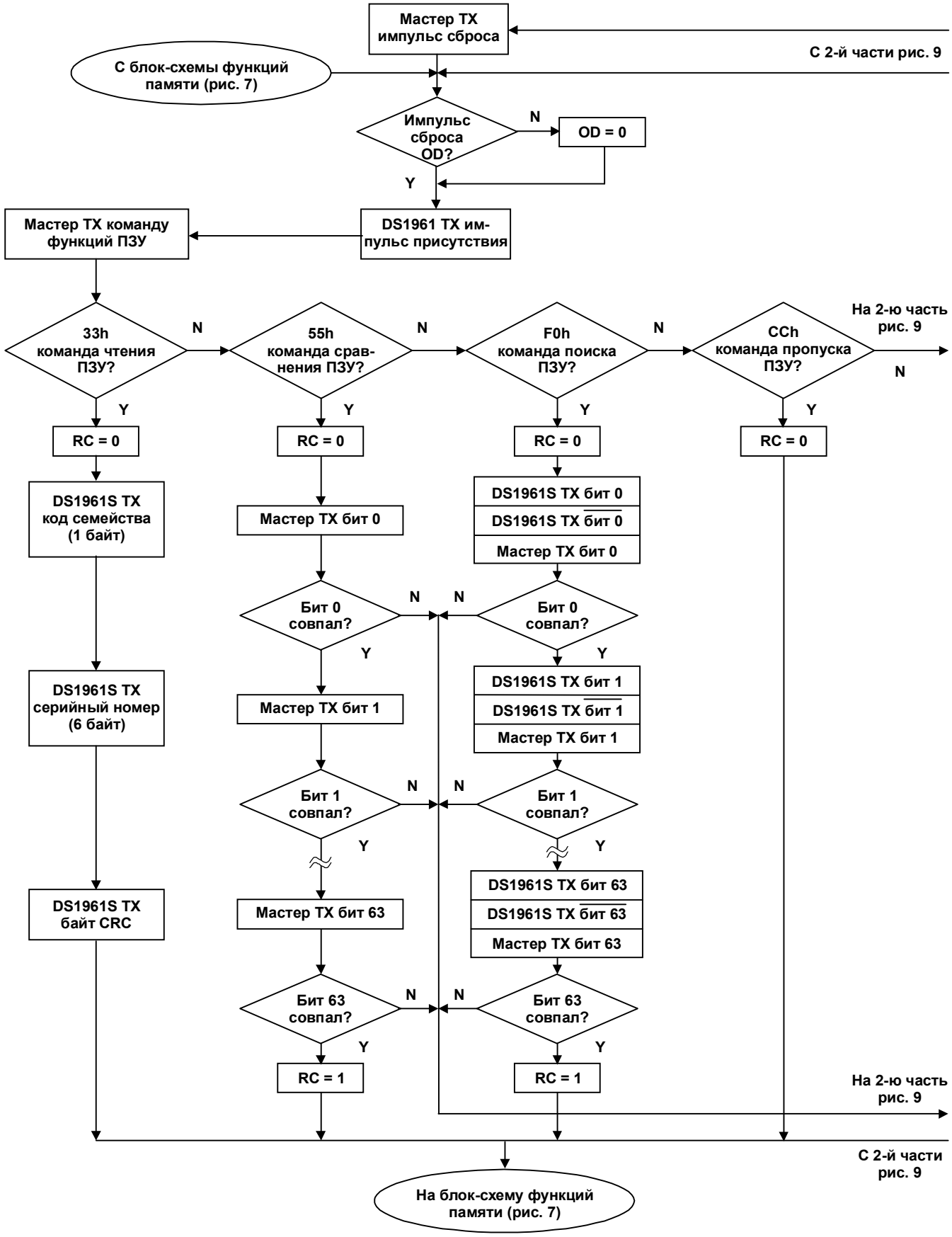
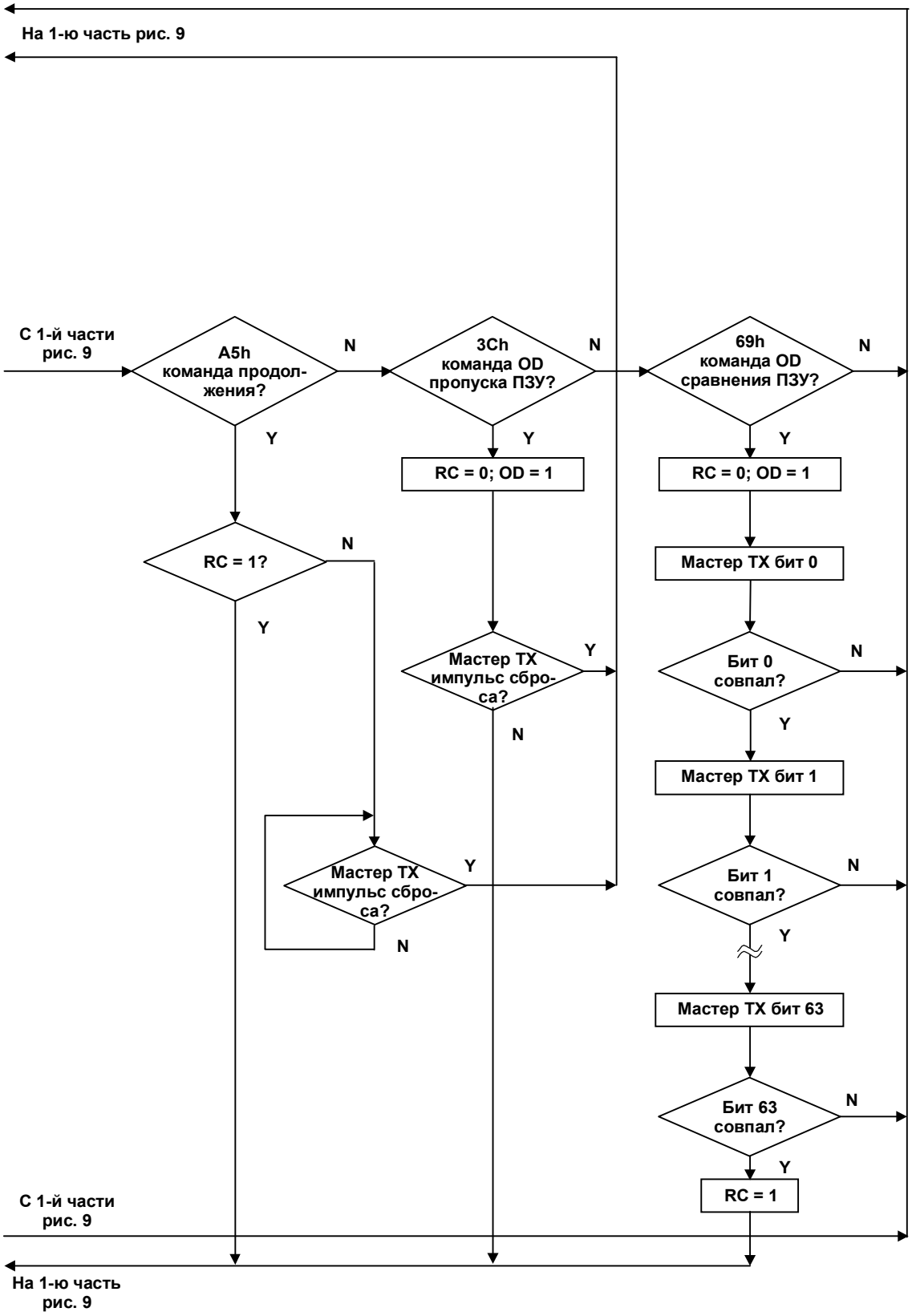


Рис. 9-2. БЛОК-СХЕМА ФУНКЦИЙ ПЗУ (продолжение)



Поиск ПЗУ [F0h]

Когда система включается в первый раз, мастер шины может не знать количества присутствующих на шине устройств или их 64-битных регистрационных номеров. Команда поиска ПЗУ позволяет мастеру шины воспользоваться процессом идентификации 64-битных номеров всех подчиненных устройств, подключенных к шине. Процесс поиска ПЗУ представляет собой повторение простой процедуры, выполняемой в три приема: чтение бита, чтение инверсии бита, затем записи значения этого бита. Мастер шины выполняет эту процедуру для каждого бита регистрационного номера. После одного полного прохода мастер шины определяет 64-битный номер одного из устройств. Регистрационные номера остальных устройств можно определить с помощью дополнительных проходов. См. главу 5 книги *«Book of DS19xx iButton Standards»*, где приведено исчерпывающее описание процесса поиска ПЗУ, включая конкретный пример.

Пропуск ПЗУ [CCh]

Эта команда позволяет экономить время в случае наличия на шине всего одного устройства, позволяя мастеру шины обращаться к функциям памяти и SHA без привлечения 64-битного регистрационного номера. Если на шине присутствует более одного подчиненного устройства, а вслед за командой пропуска ПЗУ посылается, например, команда чтения, произойдет искажение данных, так как несколько подчиненных устройств попытаются передать данные одновременно (открытые стоки реализуют функцию «монтажное И»).

Команда продолжения [A5h]

Обычно для записи полной 32-байтной страницы требуется получить доступ к DS1961S несколько раз. В случае наличия на шине более одного устройства это предполагает, что при каждой операции доступа при выполнении команды сравнения ПЗУ должна повторяться передача 64-битного регистрационного номера. Для получения в такой ситуации максимальной пропускной способности шины, была введена специальная команда продолжения. Эта команда проверяет состояние бита RC, и если он установлен, управление сразу передается функциям памяти и SHA, как в случае выполнения команды пропуска ПЗУ. Бит RC устанавливается только при успешном выполнении команды сравнения ПЗУ, поиска ПЗУ или сравнения ПЗУ в ускоренном режиме. Когда бит RC установлен, к устройству может быть осуществлен повторный доступ с помощью команды продолжения. Осуществление доступа к другому устройству на шине очищает бит RC, предотвращая одновременный ответ на команду продолжения нескольких устройств.

Пропуск ПЗУ в ускоренном режиме [3Ch]

Эта команда позволяет экономить время в случае наличия на шине всего одного устройства, позволяя мастеру шины обращаться к функциям памяти и SHA без привлечения 64-битного регистрационного номера. В отличие от обычной команды пропуска ПЗУ, команда пропуска ПЗУ в ускоренном режиме переводит DS1961S в ускоренный режим (overdrive mode, OD = 1). Любой обмен после этой команды должен производиться на повышенной скорости, пока импульс сброса длительностью минимум 480 мкс не сбросит все устройства на шине и не переведет их в режим обычной скорости (OD = 0). На многоточечной шине эта команда переводит в ускоренный режим все устройства, которые этот режим поддерживают. Для последующей адресации отдельного устройства, поддерживающего ускоренный режим, должен быть выдан импульс сброса на повышенной скорости, за которым должна следовать команда сравнения ПЗУ или поиска ПЗУ. Это ускоряет процесс поиска. Если на шине присутствует несколько подчиненных устройств, поддерживающих ускоренный режим, а за командой пропуска ПЗУ в ускоренном режиме следует команда чтения, произойдет искажение данных, так как несколько подчиненных устройств попытаются передать данные одновременно (открытые стоки реализуют функцию «монтажное И»).

Сравнение ПЗУ в ускоренном режиме [69h]

Сравнение ПЗУ в ускоренном режиме, за которым следует 64-битный регистрационный номер, передаваемый на повышенной скорости, позволяет мастеру шины адресовать отдельное устройство на многоточечной шине. Только тот экземпляр DS1961S, содержимое ПЗУ которого полностью совпадет с переданным мастером 64-битным регистрационным номером, будет отвечать на последующие команды функций памяти или SHA. Подчиненные устройства, которые уже находятся в ускоренном режиме после предыдущей команды пропуска ПЗУ в ускоренном режиме или после успешного выполнения команды сравнения ПЗУ в ускоренном режиме, остаются в этом режиме. Все подчиненные устройства, поддерживающие ускоренный режим, возвращаются в режим обычной скорости при следующем импульсе сброса длительностью минимум 480 мкс. Команда сравнения ПЗУ в ускоренном режиме может использоваться при наличии на шине как одного, так и нескольких устройств.

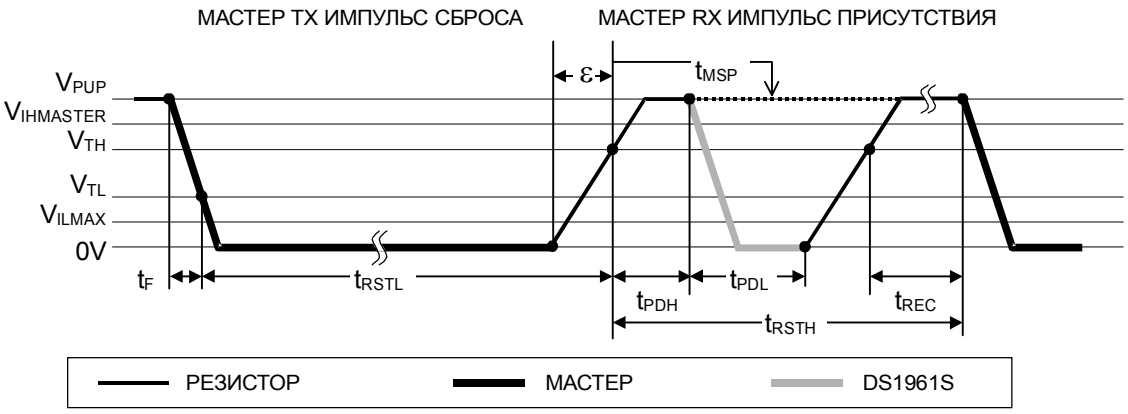
СИГНАЛЫ 1-ПРОВОДНОЙ ШИНЫ

DS1961S требует строгого соблюдения протокола для гарантии целостности данных. Протокол содержит четыре типа сигналов: последовательность сброса с импульсом сброса и импульсом присутствия, запись нуля, запись единицы и чтение данных. Все эти сигналы, за исключением импульса присутствия, инициируются мастером шины. DS1961S имеет возможность вести обмен на двух скоростях: стандартной скорости и повышенной скорости в ускоренном режиме. Если устройство специально не переведено в ускоренный режим, DS1961S работает на стандартной скорости. В ускоренном режиме все сигналы имеют меньшую длительность.

Чтобы перейти из состояния покоя в активный режим, напряжение на линии 1-проводной шины должно упасть с V_{PUP} ниже порогового значения V_{TL}. Для перехода с активного режима в состояние покоя, напряжение должно подняться с V_{ILMAX} выше порога V_{TH}. Напряжение V_{ILMAX} используется для определения логического уровня, но с ним не связана инициация каких-либо действий.

Последовательность инициализации, которая требуется для начала любого обмена с DS1961S, показана на рис. 10. За импульсом сброса следует импульс присутствия, который говорит о готовности DS1961S принять данные, представляющие собой корректные команды функций ПЗУ или памяти. В сети, содержащей разнородные устройства, длительность низкого уровня импульса сброса t_{RSTL} должна быть достаточной для того, чтобы самое медленное устройство восприняло его как импульс сброса. Для DS1961S эта длительность составляет 480 мкс на стандартной скорости и 62 мкс в ускоренном режиме. Если мастер шины использует управление скоростью нарастания на спаде импульсов, он должен удерживать низкий уровень на линии в течение времени $t_{RSTL} + t_F$ для компенсации времени спада. При длительности t_{RSTL} 480 мкс или более, устройство переходит из ускоренного режима в режим обычной скорости. Если DS1961S находится в ускоренном режиме, и длительность t_{RSTL} не превышает 80 мкс, устройство остается в ускоренном режиме.

Рис. 10. ПРОЦЕДУРА ИНИЦИАЛИЗАЦИИ (ИМПУЛЬСЫ СБРОСА И ПРИСУТСТВИЯ).



После того, как мастер освобождает линию, он переходит в режим приема (RX). Теперь 1-проводная шина находится в состоянии высокого уровня, что обеспечивается подтягивающим резистором, или в случае применения драйвера DS2480B, активной схемой. Когда достигается порог V_{TH}, DS1961S формирует задержку t_{PDH}, а затем посылает импульс присутствия путем удержания линии в состоянии низкого уровня в течение времени t_{PDL}. Для обнаружения импульса присутствия мастер должен проверить состояние линии в момент t_{MSP}.

Промежуток t_{RSTH} должен быть как минимум равен сумме t_{PDHMAX}, t_{PDLMAX} и t_{RECMIN}. Сразу после окончания интервала t_{RSTH} может производиться обмен данными. В сети, содержащей разнородные устройства, длительность t_{RSTH} должна быть увеличена как минимум до 480 мкс на стандартной скорости и до 48 мкс в ускоренном режиме для согласования с другими 1-проводными устройствами.

Временные интервалы записи и чтения

Обмен данными с DS1961S происходит с помощью временных интервалов, каждый из которых служит для передачи одного бита. Временные интервалы записи предназначены для передачи данных от мастера к подчиненному устройству, а временные интервалы чтения – от подчиненного устройства к мастеру. Определение интервалов записи и чтения проиллюстрировано на рис. 11.

Любой интервал начинается с того, что мастер переводит линию в состояние низкого уровня. Как только напряжение на линии упадет ниже порога V_{TL}, в DS1961S начинается формирование внутреннего временного интервала. Разброс длительности этого интервала определяет окно опроса подчиненного устройства, которое длится от t_{SLSMIN} до t_{SLSMAX}. Напряжение на линии данных в момент опроса определяет, каким воспринимает DS1961S этот временной интервал: как 1 или как 0. Для обеспечения надежного обмена напряжение в течение всего окна опроса должно быть или ниже V_{ILMAX}, или выше максимального значения V_{TH}.

Передача данных от мастера к подчиненному устройству

Для временного интервала записи единицы время удержания мастером низкого уровня ($t_{MPD1} = t_{W1L} - \varepsilon + t_F$) должно быть достаточно малым, чтобы позволить напряжению на линии достичь значения V_{TH} к моменту t_{SLSMIN}, ближайшей точки опроса DS1961S. После самой дальней точки опроса (t_{SLSMAX}) перед началом следующего временного интервала требуется время восстановления (t_{REC}).

Для временного интервала записи нуля время удержания мастером низкого уровня ($t_{MPD0} = t_{W0L} + t_F$) должно быть достаточно большим, чтобы сохранить напряжение на линии ниже значения V_{ILMAX} до самой дальней точки опроса DS1961S в момент t_{SLSMAX}. Перед началом следующего

временного интервала напряжение на линии данных должно подняться выше V_{TH} и оставаться таким в течение времени восстановления t_{REC}.

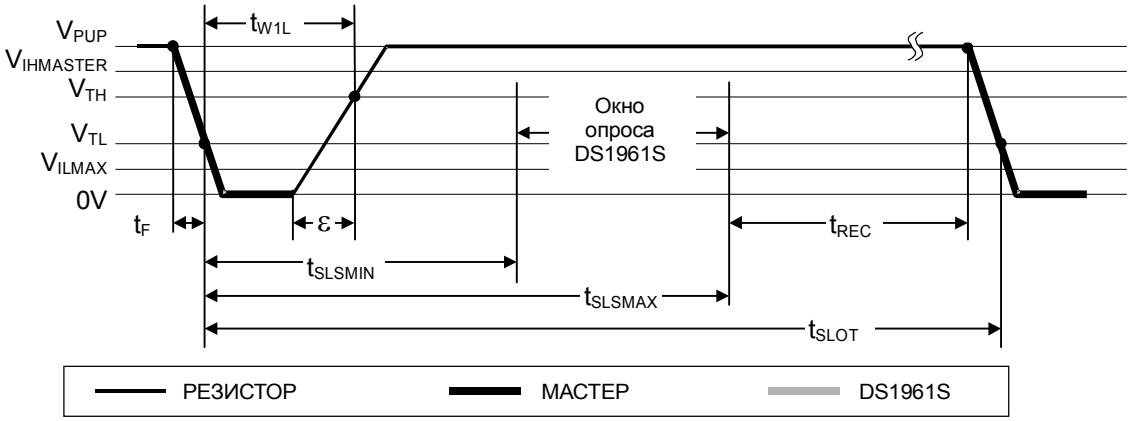
Передача данных от подчиненного устройства к мастеру

Временной интервал чтения очень похож на временной интервал записи единицы. Мастер начинает интервал чтения с того, что переводит линию в состояние низкого уровня. Как только напряжение на линии упадет ниже порога V_{TL}, в DS1961S начинается формирование внутреннего временного интервала. Время удержания мастером низкого уровня ($t_{MPDR} = t_{RL} + t_F$) должно быть достаточным, чтобы перекрыть время установления t_{SU}, после которого DS1961S выдает бит данных на 1-проводный порт. Если передается 0, DS1961S удерживает линию данных в состоянии низкого уровня в течение времени t_{SPD}. Если бит данных равен 1, DS1961S вообще не переводит линию данных в состояние низкого уровня.

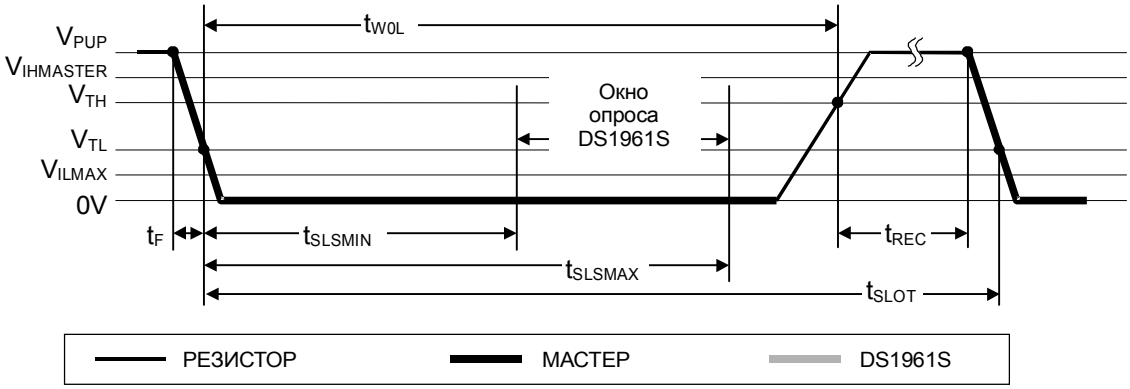
Мастер опрашивает линию данных в момент времени t_{MSR}, который лежит внутри окна, ограниченного суммой времени t_{RL} и времени нарастания (δ) с одной стороны, и временем t_{SPDMIN} с другой. Оптимальное положение точки опроса в случае чтения нуля находится не позднее момента t_{SPDMIN}. В случае чтения единицы напряжение на 1-проводной линии в момент t_{MSR} должно успеть достигнуть значения $V_{IHMASTER}$. Это условие определяет максимальную длительность удержания мастером низкого уровня. Для обеспечения надежного обмена длительность удержания мастером низкого уровня должна быть как можно меньше, чтобы предоставить максимум времени для достижения линией значения V_{IHMIN}. Перед началом следующего временного интервала по истечению t_{SPDMAX} напряжение на линии данных должно подняться выше V_{TH} и оставаться таким в течение времени восстановления t_{REC}.

Рис. 11. ВРЕМЕННАЯ ДИАГРАММА ЗАПИСИ/ЧТЕНИЯ

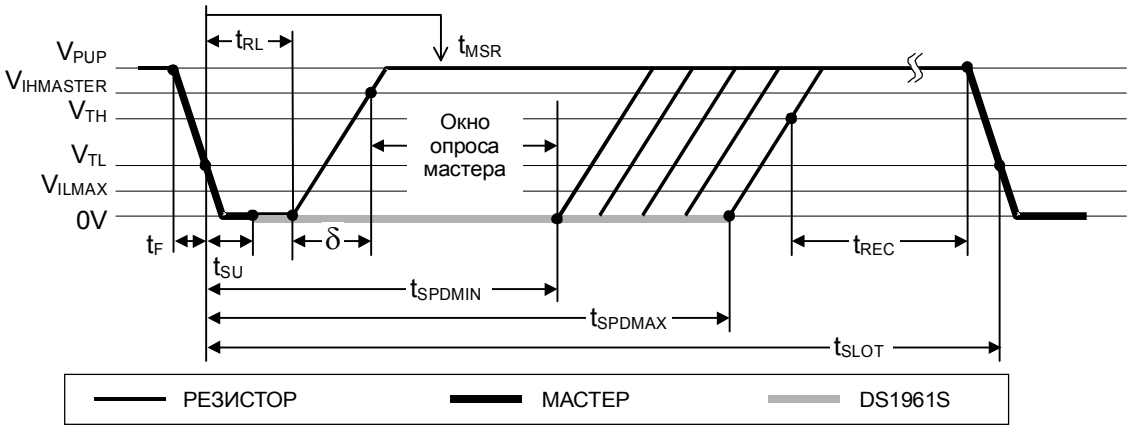
Временной интервал записи единицы



Временной интервал записи нуля



Временной интервал чтения данных



ВЫЧИСЛЕНИЕ CRC

DS1961S использует два разных типа контрольной суммы (CRC). Первым типом является 8-битная CRC. Она вычисляется при изготовлении и записывается лазером в старший байт 64-битного ПЗУ. Эквивалентный полином для этой CRC имеет следующий вид: $X^8 + X^5 + X^4 + 1$. Для проверки правильности считывания данных из ПЗУ, мастер шины может вычислить значение CRC для первых 56 бит 64-битного ПЗУ и сравнить его со значением, считанным из DS1961S. Эта 8-битная CRC принимается при чтении ПЗУ в нормальном виде (без инверсии).

Вторым типом является 16-битная CRC, вычисляемая с помощью стандартизованного полинома $X^{16} + X^{15} + X^2 + 1$. Эта CRC используется для обнаружения ошибок при чтении аутентифицированной страницы, чтении блокнота и для быстрой проверки правильности пересылки данных при записи или обновлении блокнота. Это тот же тип CRC, что используется в расширенной файловой структуре iButton. В отличие от 8-битной CRC, 16-битная CRC всегда считывается и передается в инвертированном виде. Внутренний генератор CRC в DS1961S (рис. 12) вычисляет новое значение 16-битной CRC в соответствии с блок-схемой, показанной на рис. 7. Мастер шины может сравнить значение CRC, считанное из устройства, со значением, вычисленном им самим для тех же данных. На основании результата сравнения мастер может принять решение продолжить операцию или повторить чтение той части данных, для которой обнаружена ошибка CRC.

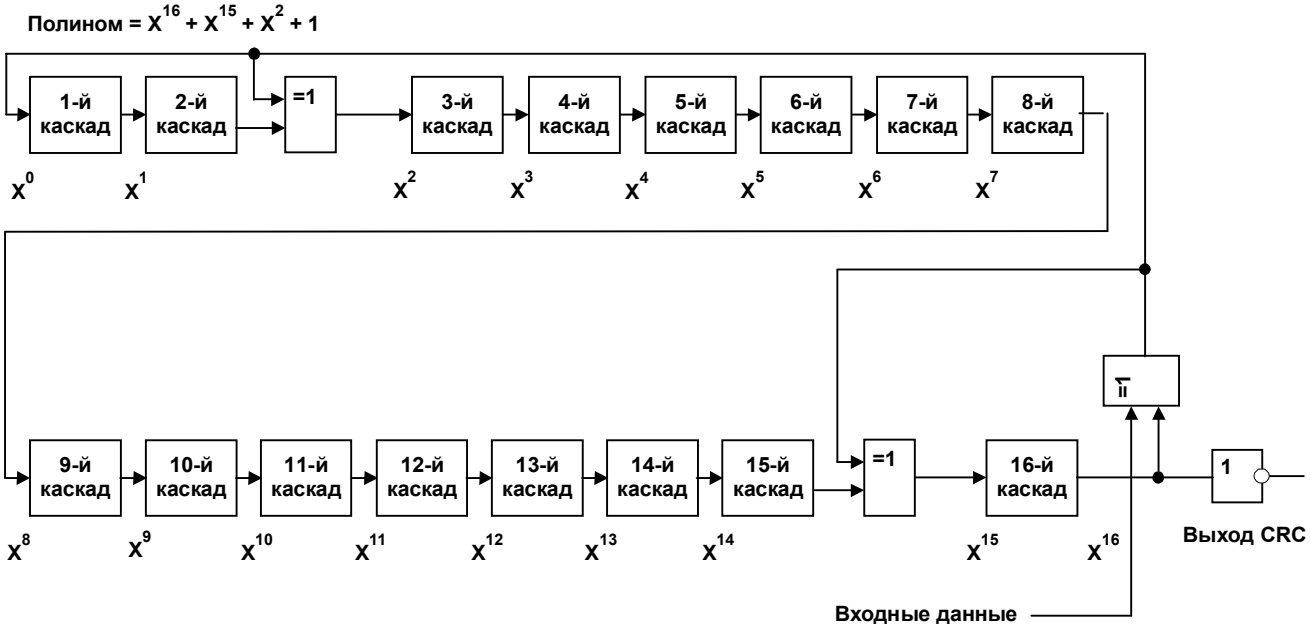
При записи блокнота, так же как и при обновлении блокнота, генерация CRC начинается очисткой сдвигового регистра генератора CRC. Затем по одному биту в сдвиговый регистр вводится код команды, адрес назначения TA1 (со сброшенными в 0 разрядами T2..T0) и TA2, а также все байты данных, переданные мастером. DS1961S передает эту CRC только в том случае, если мастер передал точно восемь байт данных.

При чтении блокнота генерация CRC также начинается очисткой сдвигового регистра генератора CRC. Затем по одному биту в сдвиговый регистр вводится код команды, адрес назначения TA1 и TA2, байт E/S, а также данные блокнота, которые были модифицированы DS1961S (см. описание команды записи блокнота). DS1961S передает эту CRC только в том случае, если мастер продолжает чтение после окончания чтения блокнота.

При чтении аутентифицированной страницы 16-битная CRC является результатом сдвига в предварительно очищенный генератор CRC байта команды, за которым следуют два байта адреса, байты данных и байт, равный FFh. CRC, которая следует за передачей MAC-кода, получается очисткой генератора CRC, затем сдвигом туда 160-битного MAC-кода с тем же порядком следования бит, что и при считывании MAC-кода мастером.

Более подробное описание процесса вычисления CRC, включая пример аппаратной и программной реализации, приведено в книге *«Book of DS19xx iButton Standards»*.

Рис. 12. АППАРАТНАЯ РЕАЛИЗАЦИЯ И ПОЛИНОМ ВЫЧИСЛЕНИЯ CRC-16



МАКСИМАЛЬНО ДОПУСТИМЫЕ УСЛОВИЯ*

Напряжение на входе/выходе относительно земли	–0,5В .. +6В
Втекающий ток входа/выхода	20 мА
Рабочая температура	–40°C .. +85°C
Температура перехода	+150°C
Температура хранения	–55°C .. +85°C

* *Функционирование устройства при этих или любых других условиях, выходящих за приведенные в спецификации рамки, не предполагается. Работа при максимально допустимых условиях в течение длительного периода времени может привести к снижению надежности.*

ЭЛЕКТРИЧЕСКИЕ ХАРАКТЕРИСТИКИ

ПАРАМЕТР	СИМВ.	УСЛОВИЯ	МИН.	ТИП.	МАКС.	ЕД.	ПРИМ.
Рабочая температура	T_A	Любые, за исключением программирования EEPROM	-40		85	°C	1
		Любые	-20		85		
Напряжение подтяжки	V_{PUP}	Стандартная скорость	2,8		5,25	В	1
		Повышенная скорость	3,3		5,25		
ОСНОВНЫЕ ПАРАМЕТРЫ ВХОДА/ВЫХОДА							
Сопротивление подтягивающего резистора	R_{PUP}				2,2	Ком	1, 2
Входная емкость	C_{IO}			100	800	Пф	3
Входной ток	I_L	При напр. на входе V_{PUP}	1		10	мкА	4
Входное напряжение низкого уровня	V_{IL}				0,30	В	1, 5, 8
Выходное напряжение низкого уровня при токе 4 мА	V_{OL}				0,4	В	5, 10
Время восстановления	t_{REC}	Стандартная скорость, R_{PUP} = 2,2 Ком	5			мкс	1
		Повышенная скорость, R_{PUP} = 2,2 Ком	2				
		Повышенная скорость, непосредственно перед импульсом сброса, R_{PUP} = 2,2 Ком	5				
Длительность временного интервала	t_{SLOT}	Стандартная скорость	65			мкс	1, 13
		Повышенная скорость, V_{PUP} > 4,5 В	7				
		Повышенная скорость	9				
ВХОД/ВЫХОД, ЦИКЛ СБРОСА И ПОЛУЧЕНИЯ ИМПУЛЬСА ПРИСУТСТВИЯ							
Длительность низкого уровня сброса	t_{RSTL}	Стандартная скорость, V_{PUP} > 4,5 В	480		640	мкс	1, 13
		Стандартная скорость	720		960		
		Повышенная скорость	68		80		
Длительность высокого уровня импульса присутствия	t_{PDH}	Стандартная скорость	15		60	мкс	13
		Повышенная скорость, V_{PUP} > 4,5 В	1		5		
		Повышенная скорость	1		6,7		

Длительность низкого уровня импульса присутствия	t_{PDL}	Стандартная скорость	60		240	мкс	13
		Повышенная скорость, V_{PUP} > 4,5 В	7,3		24		
		Повышенная скорость	7,3		28		
Время опроса импульса присутствия	t_{MSP}	Стандартная скорость	60		75	мкс	1
		Повышенная скорость, V_{PUP} > 4,5 В	5		8,3		
		Повышенная скорость	6,7		8,3		
ВХОД/ВЫХОД, ЦИКЛ ЗАПИСИ							
Длительность низкого уровня при записи 0	t_{W0L}	Стандартная скорость	60		120	мкс	13
		Повышенная скорость, V_{PUP} > 4,5 В	5		14		
		Повышенная скорость	7		14		
Длительность низкого уровня при записи 1	t_{W1L}	Стандартная скорость	5		15 - ε	мкс	1, 11, 13
		Повышенная скорость, V_{PUP} > 4,5 В	1		2 - ε		
		Повышенная скорость	1		1,85 - ε		
Окно опроса при записи (для подчиненного устройства)	t_{SLS}	Стандартная скорость	15		60	мкс	13
		Повышенная скорость, V_{PUP} > 4,5 В	2		5		
		Повышенная скорость	1,85		7		
ВХОД/ВЫХОД, ЦИКЛ ЧТЕНИЯ							
Длительность низкого уровня при чтении	t_{RL}	Стандартная скорость	5		15 - ε	мкс	1, 12, 13
		Повышенная скорость, V_{PUP} > 4,5 В	1		2 - ε		
		Повышенная скорость	1		1,85 - ε		
Длительность низкого уровня при чтении 0 (для подчиненного устройства)	t_{SPD}	Стандартная скорость	15		60	мкс	13
		Повышенная скорость, V_{PUP} > 4,5 В	2		5		
		Повышенная скорость	1,85		7		
Окно опроса при чтении	t_{MSR}	Стандартная скорость	t_{RL} + δ		15	мкс	1, 12, 13
		Повышенная скорость, V_{PUP} > 4,5 В	t_{RL} + δ		2		
		Повышенная скорость	t_{RL} + δ		1,85		
EEPROM							
Ток программир.	I_{LPROG}				700	мкА	14
Время программир.	t_{PROG}				10	мс	14
Количество циклов стирания/записи	N_{CYCLE}		50 000			–	
Время хранения данных	t_{RET}	+85°C, без питания	10			лет	
БЛОК SHA-1							
Ток при вычислении	I_{LCSHA}				4,5	мА	
Длительность вычисления	t_{CSHA}				1,5	мс	

ПРИМЕЧАНИЯ:

1) Системное требование.
2) Максимально допустимое сопротивление подтягивающего резистора является функцией количества 1-проводных устройств в системе и времени восстановления. Приведенное здесь значение дано для одного устройства и минимального времени восстановления. Для более сложных систем требуется активная подтяжка, обеспечиваемая, например, драйвером DS2408B.
3) При первом включении емкость вывода данных может достигать 800 пФ. Если используется подтягивающий резистор 5 Ком с линии данных на V_{PUP}, то достаточно времени 5 мс после включения питания, чтобы паразитная емкость перестала влиять на нормальный обмен.
4) Нагрузка на землю, представленная входом.
5) Все напряжения указаны относительно земли.
6) V_{TL}, V_{TH} являются функциями внутреннего напряжения питания.
7) Напряжение на выводе данных, ниже приведенного, воспринимается как логический 0.
8) Когда мастер удерживает линию в состоянии низкого логического уровня, напряжение на выводе данных должно быть меньше или равно V_{ILMAX}.
9) Напряжение на выводе данных, выше приведенного, воспринимается как логическая 1.
10) Вольт-амперная характеристика линейна для напряжений меньше 1В.
11) ε представляет собой время, требуемое схеме подтяжки для увеличения напряжения на линии с V_{IL} до V_{TH}.
12) δ представляет собой время, требуемое схеме подтяжки для увеличения напряжения на линии с V_{IL} до входного порога высокого логического уровня мастера.
13) Выделенные цветом значения не соответствуют опубликованному стандарту на iButton. Смотрите сравнительную таблицу, приведенную ниже.
14) Гарантируется конструкцией, не тестируется в процессе производства.

Название параметра	Стандартные значения				Значения для DS1961S			
	Стандартная скорость		Повышенная скорость		Стандартная скорость		Повышенная скорость	
	мин.	макс.	мин.	макс.	мин.	макс.	мин.	макс.
t_{SLOT} (включая t_{REC})	61 мкс	-	7 мкс	-	65 мкс	-	9 мкс	-
t_{RSTL}	480 мкс	-	48 мкс	80 мкс	720 мкс	960 мкс	68 мкс	80 мкс
t_{PDH}	15 мкс	60 мкс	2 мкс	6 мкс	15 мкс	60 мкс	1 мкс	6,7 мкс
t_{PDL}	60 мкс	240 мкс	8 мкс	24 мкс	60 мкс	240 мкс	7,3 мкс	24 мкс
t_{W0L}	60 мкс	120 мкс	6 мкс	16 мкс	60 мкс	120 мкс	7 мкс	14 мкс
t_{SLS}, t_{SPD}	15 мкс	60 мкс	2 мкс	6 мкс	15 мкс	60 мкс	1,85 мкс	7 мкс