DS1963S
iButton с блоком SHA

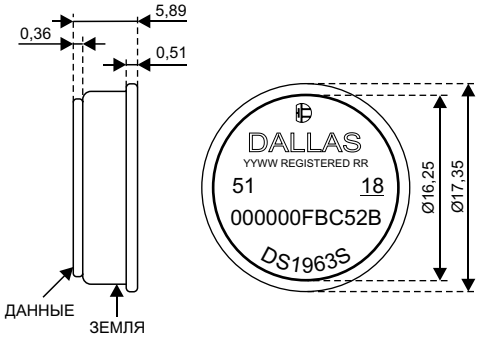
ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

- Энергонезависимая память объемом 4096 бит с возможностью записи/чтения, организованная в виде 16-ти страниц по 256 бит каждая
- Восемь страниц памяти имеют индивидуальные ключи доступа в виде 64-битных секретных кодов и 32-битные «только для чтения» счетчики количества циклов записи без возможности переполнения
- Секретные коды «только для записи» имеют собственные индивидуальные счетчики количества циклов записи
- Встроенный 512-битный блок SHA-1, предназначенный для вычисления 160-битного кода аутентификации сообщения (Message Authentication Code, или MAC-кода) и генерации ключей доступа (секретных кодов) для страниц памяти
- Устройство может использоваться как iButton® роуминга, или как сопроцессор для хост-компьютера
- Блокнотная память объемом 256 бит обеспечивает целостность данных при пересылках
- 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 –20°C до +85°C
- Сохранность данных не менее 10 лет

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
- Экономичный обмен с мастером шины с помощью единственного цифрового сигнала на скорости 15,4 Кбит в секунду
- Стандартный диаметр 16 мм и 1-проводный протокол (1-Wire® Protocol) гарантируют совместимость с семейством iButton
- 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, раздел 1, группы A, B, C и D

F5 MICROCAN



Все размеры приведены в миллиметрах.

ИНФОРМАЦИЯ ДЛЯ ЗАКАЗА

DS1963S	корпус F5 MicroCan

ПРИМЕРЫ АКСЕССУАРОВ

DS9096P	Самоклеющаяся подложка
DS9101	Универсальный зажим
DS9093RA	Крепежное кольцо
DS9093A	Держатель с защелкой
DS9092	Контактное устройство

ОПИСАНИЕ iButton

iButton для проведения электронных платежей DS1963S с блоком SHA-1 является надежным носителем данных емкостью 4 Кбит, доступ к которым может быть осуществлен с помощью минимального аппаратного обеспечения. Энергонезависимая память устройства выполняет функции локальной базы данных, которая может хранить как общедоступные, так и защищенные данные, принадлежащие пользователю устройства. Встроенный 512-битный блок SHA может быть задействован для вычисления 160-битного кода аутентификации (MAC-кода) на основе хранящихся в устройстве данных. Данные передаются последовательно с помощью 1-проводного протокола, который требует только одного вывода данных и общего провода. При использовании файлового формата TMEX (см. Application Note 114), один экземпляр DS1963S может поддерживать до четырех независимых приложений, таких как осуществление электронных платежей при расчетах за транспортные услуги, телефонные переговоры, парковку автомобилей или покупки в торговых автоматах. DS1963S также может работать в качестве сопроцессора для хост-компьютера, осуществляя вычисление кодов доступа, необходимых для записи нового баланса в роуминговое устройство после проведения платежа.

DS1963S, как и другие iButton со встроенным ОЗУ, имеет дополнительную область данных, которая называется блокнотом и работает как буфер при записи основной памяти. Блокнот DS1963S также используется для передачи сегментов данных блоку SHA-1 или для приема/сравнения кода аутентификации сообщения.

Данные вначале записываются в блокнот, откуда они могут быть считаны. После проверки правильности данных команда копирования блокнота пересылает их в основную память. Этот процесс гарантирует целостность данных даже в условиях ненадежного электрического контакта.

DS1963S 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 DS1963S практически на любые поверхности, включая пластиковые держатели, идентификационные бэджи и печатные платы.

БЕЗОПАСНОСТЬ

Система, использующая мобильные носители данных, обычно состоит из трех компонентов: 1) хост-компьютеров, которые производят запись и считывание данных с носителей, 2) самих носителей данных («подчиненных устройств») и 3) пользователей системы, которые могут

попытаться подделать данные или эмулировать поведение носителя. Конструкция DS1963S разработана так, чтобы противостоять всем таким попыткам без использования каких-либо алгоритмов, имеющих патентные ограничения. Безопасность устройства основана на стандарте SHA-1 (Secure Hash Standard), описание которого можно найти в Интернете по адресу http://www.itl.nist.gov/div897/pubs/fip180-1.htm.

Ниже в виде таблицы истинности показаны комбинации возможных способов нарушения защиты. Примечания к таблице поясняют типичные методы противостояния попыткам нарушения. Более подробное описание можно найти в разделе «Обзор применений» в конце этого документа. Для получения полной информации по используемым функциям, смотрите раздел «Команды функций памяти и SHA», а также описания вычислений SHA-1 и формата посылок.

	Аутентичные данные	**Подделанные данные**	
Авторизованный хост	См. примечание 2	См. примечание 2 и 3	**Эмулированный носитель**
	Нормальная работа	См. примечание 3	**Аутентичный носитель**
Неавторизованный хост	См. примечание 1	Не важно	
	Не важно	Не важно	**Эмулированный носитель**

Примечание 1: Устройство производит авторизацию хоста на основе общего системного секретного кода, регистрационного номера устройства из ПЗУ и выбранного пользователем личного идентификационного номера, который находится в одной из страниц памяти носителя данных.

Примечание 2: Для определения того, является ли носитель аутентичным, хост записывает в блокнот устройства 3-байтный запрос перед выдачей команды вычисления MAC-кода, которое выполняется на основе этого запроса, данных страницы памяти, номера страницы, счетчика количества циклов записи страницы, регистрационного номера устройства из ПЗУ и секретного кода, установленного для данной страницы памяти. Изменяя запрос при каждом чтении, хост может проверить, содержит ли носитель правильный секретный код и может ли он выполнить требуемые вычисления SHA за отведенное время.

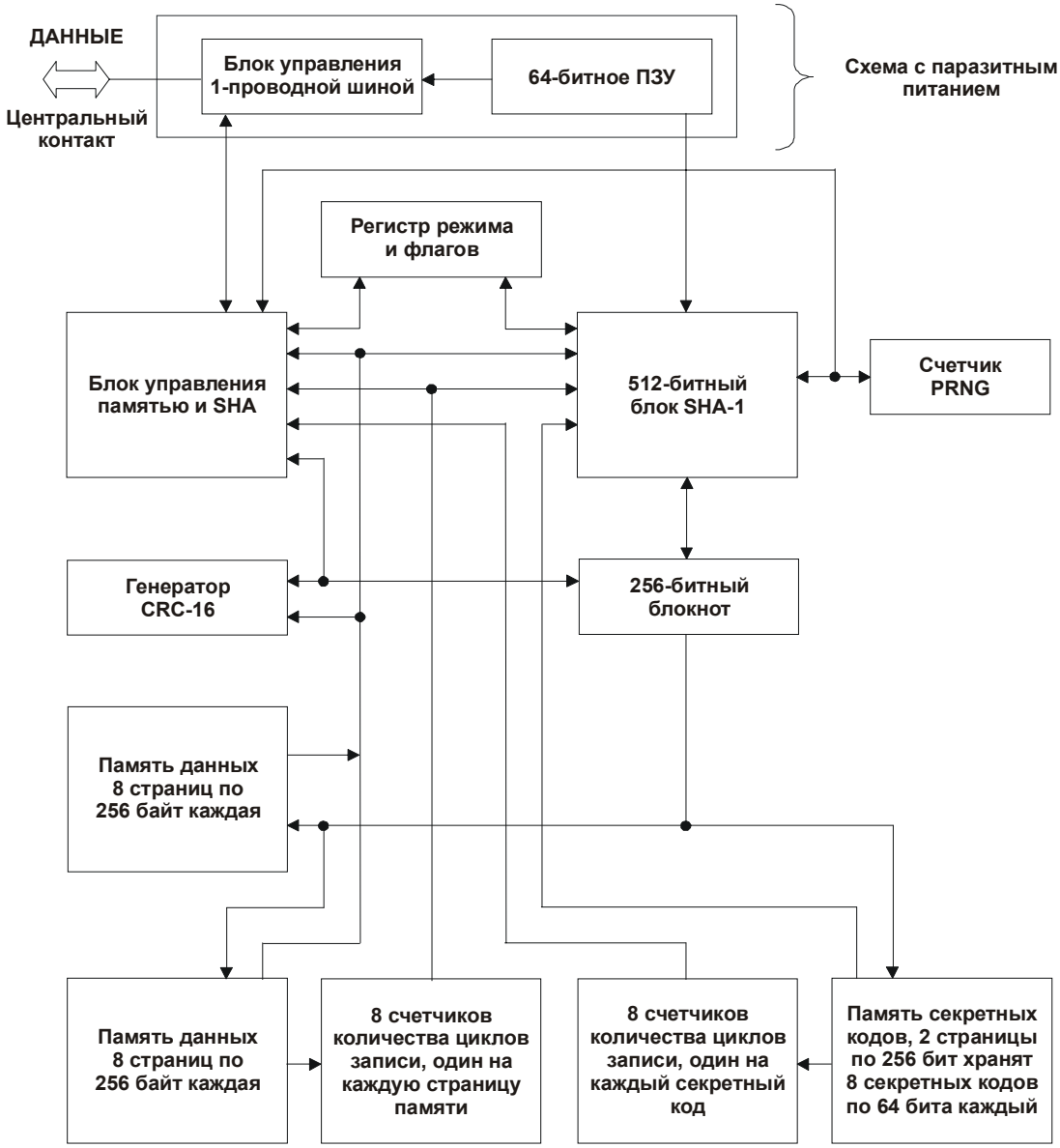
Примечание 3: Подделка данных может быть обнаружена в том случае, если данные в носителе снабжены «подписью» авторизованного хоста. Процесс подписи включает вычисление 160-битного MAC-кода на основе защищаемых данных, счетчика количества циклов записи страницы, в которой они сохранены, регистрационного номера устройства из ПЗУ и специального секретного кода, известного только авторизованному хосту. MAC-код сохраняется вместе с данными приложения (например, вместе с балансом и кодом идентификации транзакции) в соответствующей странице памяти. Для проверки аутентификации данных хост повторяет процесс подписи. Любое отличие в данных, счетчике циклов записи или несоответствующий секретный код приведут к ошибке проверки подписи.

ОБЗОР

Блок-схема, показанная на рис. 1, демонстрирует связи между блоками управления и блоками памяти DS1963S. Всего DS1963S имеет шесть основных компонентов хранения и обработки данных: 1) 64-битное ПЗУ, записанное лазером, 2) 256-битный блокнот, 3) восемь 32-байтных страниц ОЗУ общего назначения, 4) восемь 32-байтных страниц ОЗУ, защищенных счетчиками циклов записи, 5) две 32-байтные страницы, хранящие восемь 64-битных секретных кодов с индивидуальными счетчиками количества циклов записи и 6) 512-битный блок SHA-1 (Secure Hash Algorithm).

Иерархическая структура 1-проводного протокола показана на рис. 2. Все счетчики циклов записи являются 32-битными и не переполняются (не сбрасываются в 0), когда достигается максимальное значение счета. Содержимое счетчиков считывается вместе с данными памяти с помощью специальной команды. 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. 10. После того, как команда функции ПЗУ успешно выполнена, становятся доступными функции памяти, и мастер может передать одну из восьми команд функций памяти. Протокол для этих команд показан на рис. 7. При считывании и записи всех данных первым передается младший бит.

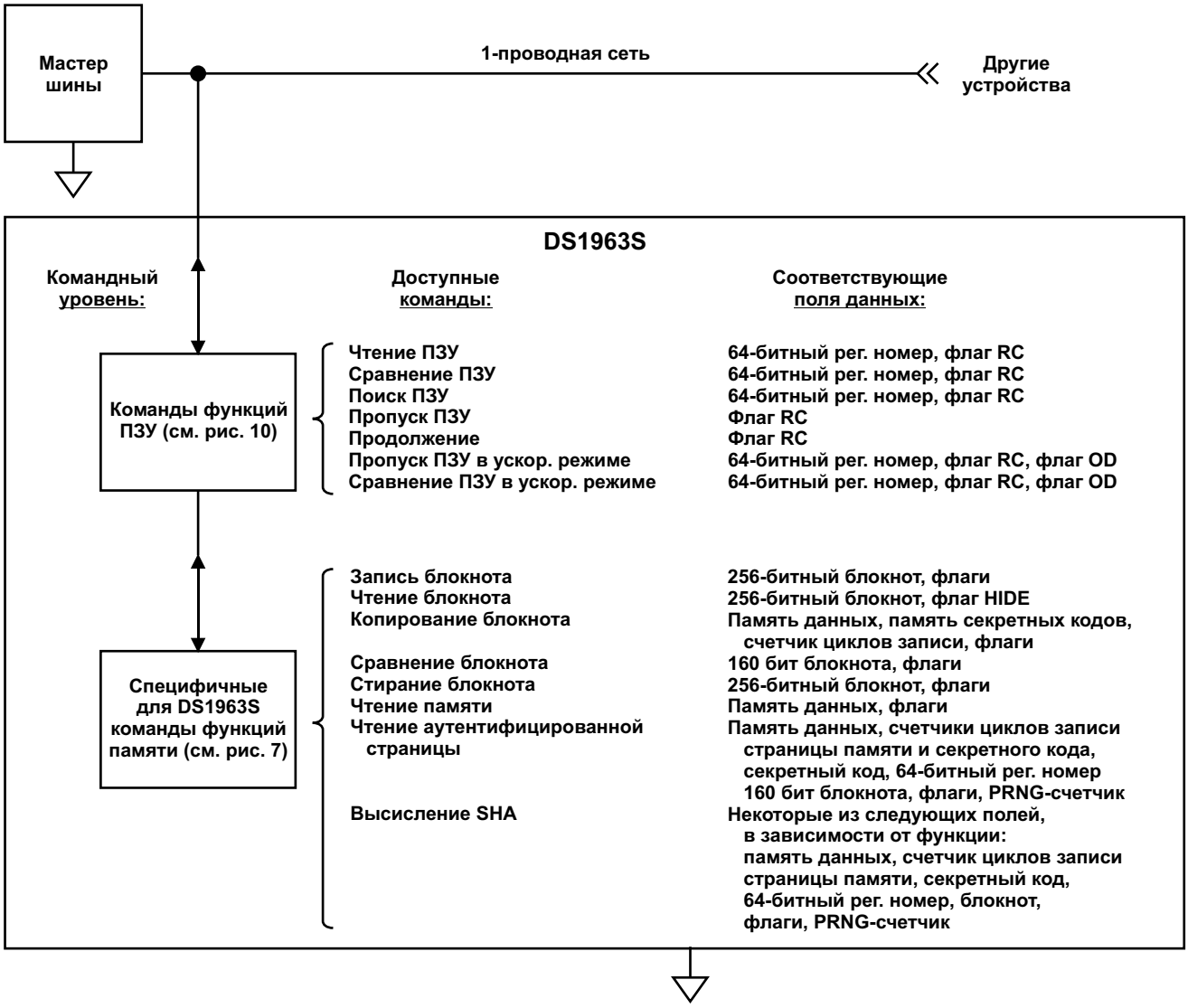
Рис. 1. БЛОК-СХЕМА DS1963S



ПАРАЗИТНОЕ ПИТАНИЕ

На блок-схеме устройства (рис. 1) показана схема, имеющая паразитное питание. Эта схема запасает энергию, когда вывод данных находится в состоянии высокого логического уровня. Запасенной энергии достаточно для питания в те моменты времени, когда вывод данных находится в состоянии низкого логического уровня, если выдерживаются требуемые временные параметры и напряжение на линии данных. Паразитное питание имеет два преимущества: 1) благодаря получению энергии с линии данных экономится внутренний литиевый источник и 2) если литиевый источник по каким-то причинам истощился, содержимое ПЗУ все равно может быть нормально считано. Остальная часть схемы DS1963S питается исключительно от литиевого источника.

Рис. 2. ИЕРАРХИЧЕСКАЯ СТРУКТУРА 1-ПРОВОДНОГО ПРОТОКОЛА



64-БИТНОЕ ПЗУ, ЗАПИСАННОЕ ЛАЗЕРОМ

Каждый экземпляр DS1963S 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. 3. 64-БИТНОЕ ПЗУ, ЗАПИСАННОЕ ЛАЗЕРОМ

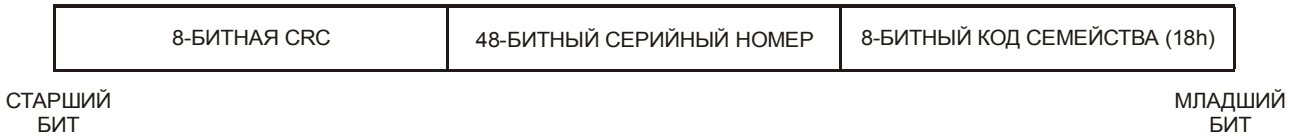
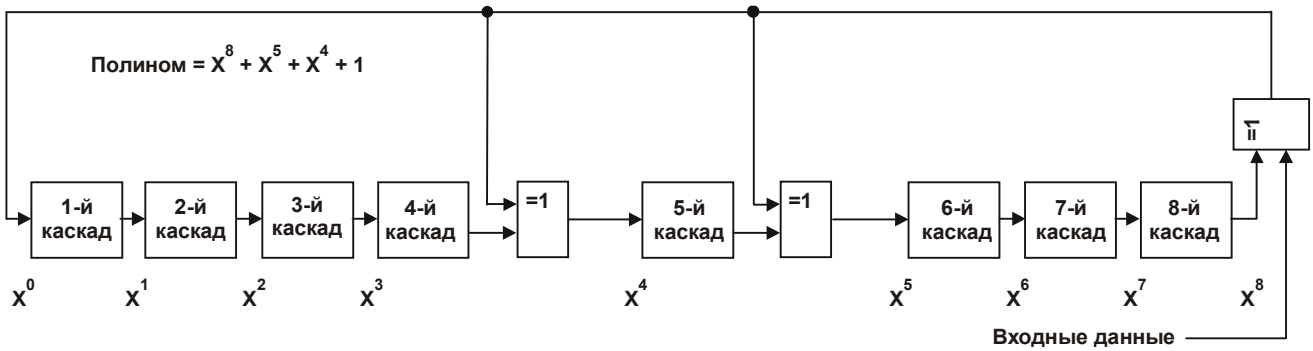


Рис. 4. ГЕНЕРАТОР CRC



КАРТА ПАМЯТИ

Как показано на блок-схеме, DS1963S имеет четыре области памяти: память данных, память секретных кодов, счетчики и блокнот. Каждая из этих областей памяти организована в виде страниц по 32 байта, как показано на рис. 5. Блокнот используется как буфер при записи памяти данных или памяти секретных кодов. Страницы 0..15 имеют неограниченный доступ для записи/чтения. Они насчитывают 4096 бит энергонезависимого ОЗУ. Страницы 16 и 17 содержат восемь 64-битных секретных кодов, которые пользователь может только записывать. Для чтения секретные коды доступны только блоку SHA, который использует их для вычисления кода аутентификации сообщения. Шестнадцать 32-битных счетчиков подсчитывают количество циклов записи в страницы 8..15, а также в область каждого из восьми секретных кодов. Эти счетчики расположены в страницах 19 и 20 и могут быть считаны без каких-либо ограничений. Страница 21 содержит счетчик, который инкрементируется при каждом запуске блока SHA. Значение этого счетчика используется для генерации псевдослучайных чисел и поэтому он называется PRNG-счетчиком. Поскольку блок SHA потребляет примерно в 20 раз больше энергии, чем копирование всего блокнота в память, PRNG-счетчик может быть использован как индикатор оставшихся в устройстве запасов энергии. Блокнот физически расположен в странице 18.

АДРЕСНЫЕ РЕГИСТРЫ И СОСТОЯНИЕ ПЕРЕСЫЛКИ

DS1963S использует три адресных регистра: TA1, TA2 и E/S (рис. 6). Регистры TA1 и TA2 загружаются адресом назначения, который указывает, куда должны быть записаны или откуда считаны данные. Регистр E/S является счетчиком байт и регистром состояния пересылки. Он доступен только для чтения и используется для проверки целостности данных при выполнении команд записи. Пять младших битов регистра E/S содержат адрес последнего байта, записанного в блокнот для последующего копирования в основную память. Этот адрес называется конечным смещением. Бит 5 регистра E/S, называемый флагом PF, или флагом неполного байта (partial byte flag), устанавливается в 1, если количество бит данных, переданных мастером, не кратно восьми. Бит 6 не несет никаких функций; он всегда считывается как 0. Заметьте, что пять младших битов адреса назначения также определяют начальный адрес в блокноте, где осуществляется промежуточное хранение данных. Этот адрес называется смещением байта. Если адрес назначения (TA1) для команды записи равен, например, 3Ch, то в блокноте поступающие данные будут сохраняться, начиная со смещения байта 1Ch, и блокнот заполнится после приема всего 4-х байт, что даст конечное смещение 1Fh. Конечное смещение вместе с флагом неполного байта позволяют мастеру осуществлять проверку целостности данных после команды записи. Старший бит регистра E/S называется флагом 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.

Рис. 5. КАРТА ПАМЯТИ DS1963S

	Память данных общего назначения с доступом для записи/чтения				
	Номер страницы	**Диапазон адресов**	**Номер секретного кода**	**Номер счетчика**	**Инкремент счетчика**
4Кбит энергонезависимой памяти	0	0000h – 001Fh	0	0	Нет
	1	0020h – 003Fh	1	1	Нет
	2	0040h – 005Fh	2	2	Нет
	3	0060h – 007Fh	3	3	Нет
	4	0080h – 009Fh	4	4	Нет
	5	00A0h – 00BFh	5	5	Нет
	6	00C0h – 00DFh	6	6	Нет
	7	00E0h – 00FFh	7	7	Нет
	8	0100h – 011Fh	0	0	При записи
	9	0120h – 013Fh	1	1	При записи
	10	0140h – 015Fh	2	2	При записи
	11	0160h – 017Fh	3	3	При записи
	12	0180h – 019Fh	4	4	При записи
	13	01A0h – 01BFh	5	5	При записи
	14	01C0h – 01DFh	6	6	При записи
	15	01E0h – 01FFh	7	7	При записи

Рис. 5. КАРТА ПАМЯТИ DS1963S (продолжение)

Память секретных кодов с доступом пользователя только для записи

Номер страницы	Диапазон адресов	Описание
16	0200h – 0207h	Секретный код 0
	0208h – 020Fh	Секретный код 1
	0210h – 0217h	Секретный код 2
	0218h – 021Fh	Секретный код 3
17	0220h – 0227h	Секретный код 4
	0228h – 022Fh	Секретный код 5
	0230h – 0237h	Секретный код 6
	0238h – 023Fh	Секретный код 7

Память счетчиков с доступом пользователя только для чтения

Номер страницы	Диапазон адресов	Описание
19	0260h – 0263h	Счетчик 0 (циклы записи для страницы 8)
	0264h – 0267h	Счетчик 1 (циклы записи для страницы 9)
	0268h – 026Bh	Счетчик 2 (циклы записи для страницы 10)
	026Ch – 026Fh	Счетчик 3 (циклы записи для страницы 11)
	0270h – 0273h	Счетчик 4 (циклы записи для страницы 12)
	0274h – 0277h	Счетчик 5 (циклы записи для страницы 13)
	0278h – 027Bh	Счетчик 6 (циклы записи для страницы 14)
	027Ch – 027Fh	Счетчик 7 (циклы записи для страницы 15)
20	0280h – 0283h	Счетчик циклов записи секретного кода 0
	0284h – 0287h	Счетчик циклов записи секретного кода 1
	0288h – 028Bh	Счетчик циклов записи секретного кода 2
	028Ch – 028Fh	Счетчик циклов записи секретного кода 3
	0290h – 0293h	Счетчик циклов записи секретного кода 4
	0294h – 0297h	Счетчик циклов записи секретного кода 5
	0298h – 029Bh	Счетчик циклов записи секретного кода 6
	029Ch – 029Fh	Счетчик циклов записи секретного кода 7
21	02A0h – 02A3h	PRNG-счетчик

Рис. 6. АДРЕСНЫЕ РЕГИСТРЫ

Номер бита	7	6	5	4	3	2	1	0
Адрес назначения (TA1)	T7	T6	T5	T4	T3	T2	T1	T0
Адрес назначения (TA2)	T15	T14	T13	T12	T11	T10	T9	T8
Конечный адрес со статусом данных (E/S) (только для чтения)	AA	0	PF	E4	E3	E2	E1	E0

ЗАПИСЬ С ПРОВЕРКОЙ

Для промежуточного хранения данных, записываемых в DS1963S, используется блокнот. Вначале мастер посылает команду записи блокнота и задает желаемый адрес назначения, за которым следуют данные, предназначенные для записи в блокнот. При некоторых условиях (см. описание команды записи блокнота) в конце команды записи блокнота мастер принимает инвертированную CRC16, рассчитанную для кода команды, адреса и данных. Зная значение CRC, мастер может сравнить его со значением, вычисленным им самим, чтобы убедиться в правильности пересылки данных и приступить к команде копирования блокнота. Если мастер не получает CRC16, он может выдать команду чтения блокнота для проверки правильности данных. Как преамбулу к данным блокнота, DS1963S повторяет адрес назначения TA1 и TA2, а также передает содержимое регистра E/S. Если флаг PF установлен, значит, данные были переданы в блокнот с ошибками. В этом случае мастер может не продолжать чтения; он может начать новую попытку записи данных в блокнот. Подобным образом установка флага AA говорит о том, что устройством была не распознана команда записи. Если все прошло успешно, оба флага очищены, а конечное смещение указывает на адрес последнего байта, записанного в блокнот. В этом случае мастер может продолжить чтение и проверить каждый байт данных. После проверки данных, мастер может выдать команду копирования блокнота. За кодом этой команды должно следовать точное содержимое регистров TA1, TA2 и E/S. Мастер может получить содержимое этих регистров путем чтения блокнота или вычислить их исходя из адреса назначения и количества записываемых данных. Как только DS1963S правильно принимает эти байты, происходит копирование данных в нужную область памяти, начиная с адреса назначения.

КОМАНДЫ ФУНКЦИЙ ПАМЯТИ И SHA

В соответствии с требованиями безопасности, которые учтены в конструкции, DS1963S ведет себя иначе, нежели другие устройства iButton с памятью. Несмотря на то, что память данных DS1963S может быть считана таким же образом, как и у других устройств iButton, попытки чтения страниц 16 и 17, где хранятся секретные коды, и страницы 18, где физически расположен блокнот, приведут к считыванию байтов, равных FFh, вместо реальных данных. Другие функции, которые имеются как у DS1963S, так и у обычных устройств iButton с памятью, управляются флагом под названием HIDE. Когда флаг HIDE сброшен, эти функции работают так же, как и у других устройств iButton с памятью. Флаг HIDE обычно управляется (устанавливается и сбрасывается) функциями, которые выполняет блок SHA. Для предохранения данных блокнота от несанкционированного считывания флаг HIDE автоматически устанавливается, как только схема с паразитным питанием выполняет начальный сброс, что происходит всякий раз, когда DS1963S начинает работать со считывателем. После этого флаг HIDE очищается по команде стирания блокнота, которая, кроме того, стирает все данные, оставшиеся в блокноте.

Блок-схема функций памяти и SHA (рис. 7) описывает протоколы, необходимые для доступа к памяти и работы с блоком SHA. Обмен между мастером и DS1963S может происходить как на обычной скорости (по умолчанию, OD = 0), так и в ускоренном режиме на повышенной скорости (OD = 1). Если DS1963S специально не перевести в ускоренный режим, обмен будет происходить на обычной скорости.

Команда записи блокнота [0Fh]

HIDE = 0, адрес назначения может лежать только в диапазоне 0000h – 01FFh:

После выдачи команды записи блокнота мастер должен сначала передать 2-байтный адрес назначения, а затем данные, предназначенные для записи в блокнот. Данные записываются в блокнот, начиная со смещения байта (T4:T0). В тот момент, когда мастер закончит запись данных, конечное смещение (E4:E0) будет равно смещению байта. 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. 13) 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), который был передан мастером, и все байты данных. Мастер может завершить выполнение команды записи блокнота в любой момент времени. Однако если конечное смещение равно 11111b, мастер может выдать 16 интервалов чтения и принять значение CRC, вычисленное DS1963S.

HIDE = 1, адрес назначения может лежать только в диапазоне 0200h – 023Fh:

Функционирование команды ограничено выбором секретного кода, который будет перезаписан данными, хранящимися в блокноте. Эти данные обычно являются результатом выполненной перед этим команды вычисления первого секретного кода или команды вычисления следующего секретного кода. Адреса восьми секретных кодов показаны на рис. 5. Адрес, передаваемый после кода команды, может указывать в любое место диапазона адресов, принадлежащего выбранному секретному коду. Вслед за адресом назначения мастер может передать байты данных, как и в случае записи блокнота. Как только передано столько данных, сколько может вместиться в блокнот, начиная с выбранного адреса назначения, мастер может выдать 16 интервалов чтения и принять значение CRC, вычисленное DS1963S. Переданные байты данных используются при вычислении CRC, но реально в блокнот не записываются.

Команда чтения блокнота [AAh]

HIDE = 0:

Команда чтения блокнота позволяет произвести проверку адреса назначения, конечного смещения и целостности данных, записанных в блокнот. После выдачи кода команды мастер приступает к чтению. Два первых байта представляют собой адрес назначения. Следующий байт представляет собой конечное смещение/статус данных (E/S). За ним следуют данные, содержащиеся в блокноте, начиная со смещения байта (T4:T0). Мастер может считать блокнот до конца, после чего он примет инвертированное значение CRC, вычисленное DS1963S. Если мастер продолжит чтение после получения CRC, все последующие считанные данные будут представлять собой логические единицы.

HIDE = 1:

Функционирование команды ограничено чтением адреса назначения и конечного смещения. Взамен данных блокнота мастер будет считывать логические единицы, пока не будет достигнут

конец блокнота. После этого мастер примет значение CRC, вычисленное DS1963S. Если мастер продолжит чтение после получения CRC, все последующие считанные данные будут представлять собой логические единицы.

Команда копирования блокнота [55h]

HIDE = 0, адрес назначения может лежать только в диапазоне 0000h – 01FFh:

Команда копирования блокнота используется для записи данных из блокнота в страницу памяти. После выдачи кода этой команды 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, устанавливается флаг AA (флаг принятия авторизации) и начинается копирование. Во время процесса копирования данных мастер будет считывать логические единицы. После завершения процесса копирования мастеру будет передаваться последовательность чередующихся нулей и единиц, вплоть до выдачи мастером импульса сброса. Любая попытка сбросить устройство в процессе копирования данных будет игнорирована. Процесс копирования обычно длится 30 мкс.

Данные, которые должны быть скопированы, определяются тремя адресными регистрами. Содержимое блокнота, начиная со смещения байта и заканчивая конечным смещением, будет скопировано в память, начиная с адреса назначения. В любом случае, этой командой может быть скопировано в память от 1 до 32 байт. Флаг AA очищается только при выполнении команды копирования блокнота.

HIDE = 1, адрес назначения может лежать только в диапазоне 0200h – 023Fh:

Функция выполняется так же, как было описано выше, если адрес назначения и конечное смещение совпадают с адресом секретного кода. Если адрес назначения указывает на основную память, а флаг HIDE установлен (например, начальным сбросом схемы с паразитным питанием), копирование данных блокнота осуществляться не будет. Тем не менее, если записать данные в блокнот, затем установить флаг HIDE, затем выполнить команду записи блокнота для выбора номера секретного кода, а затем выполнить команду копирования блокнота, то возможна запись известных данных («пароля») в область секретного кода. Однако выполнение этой процедуры не рекомендуется, так как это снижает уровень безопасности.

Команда чтения памяти [F0h]

Команда чтения памяти может использоваться для чтения страниц памяти 0..15, счетчиков циклов записи, размещенных в страницах 19 и 20 и PRNG-счетчика в начале страницы 21. Попытка чтения памяти секретных кодов, хранящихся в страницах 16 и 17, не приведет к получению настоящих данных. Попытка чтения страницы памяти 18 приведет к считыванию данных блокнота, если флаг HIDE очищен (HIDE = 0), и значений FFh, если этот флаг установлен (HIDE = 1). После кода команды мастер должен передать 2-байтный адрес назначения. После этих двух байт мастер считывает данные, начиная с адреса назначения. Он может продолжать считывание вплоть до конца PRNG-счетчика и далее. За PRNG-счетчиком следуют 12 неопределенных байт. Если мастер продолжит чтение дальше, то получит одни логические единицы. Важно представлять, что регистры адреса назначения будут указывать на последний считанный байт. Байт конечного смещения/состояния данных не изменяется.

DS1963S имеет аппаратные средства для осуществления безошибочной записи в память. 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).

Команда стирания блокнота [C3h]

Предназначением этой команды является очистка флага HIDE, а также стирание данных, которые могли остаться в блокноте после осуществления предыдущей операции. После выдачи кода команды мастер передает адрес назначения, как и для команды записи блокнота, но не передает данных. После этого весь блокнот будет автоматически заполнен байтами FFh, независимо от адреса назначения. Этот процесс длится примерно 32 мкс, в течение которых мастер принимает единицы. После завершения процесса мастер начнет принимать последовательность чередующихся нулей и единиц, что будет свидетельствовать о завершении выполнения команды.

Команда сравнения блокнота [3Ch]

MAC-код, вычисляемый блоком SHA DS1963S, записывается в блокнот. В то же время некоторые вычисления, которые производятся при авторизации хоста или при выполнении функции проверки страницы данных, требуют установки флага HIDE. Команда сравнения блокнота позволяет проверить данные, которые при этом остаются недоступными для чтения. Команда сравнивает 160-битный MAC-код, который находится после вычислений SHA в блокноте по адресам 8..27, как описано в разделах «Алгоритм вычислений SHA» и «Форматы выходных данных SHA-1», с результатом, полученным мастером при его собственных вычислениях. После того, как мастер выдаст код команды сравнения блокнота, он передает данные, байт за байтом, начиная с байта 8 и заканчивая байтом 27. Если все байты совпадают, мастер считывает последовательность чередующихся нулей и единиц. Если был дополнительно установлен флаг AUTH, устанавливается флаг MATCH. Если сравнение выявило различия, мастер считывает все единицы.

Чтение аутентифицированной страницы [A5h]

Эта команда, применимая только к страницам 0..15, позволяет мастеру получить данные полной (или части) страницы памяти и вычисленный MAC-код. После передачи мастером кода команды и правильного адреса назначения, он принимает данные страницы, начиная с адреса назначения и до конца страницы данных, затем значение счетчика циклов записи для этой страницы, значение счетчика циклов записи для секретного кода, ассоциированного с этой страницей и инвертированное значение CRC для кода команды, адреса назначения, переданной страницы данных и значений счетчиков. Сразу после приема мастером значения CRC, блок SHA начинает вычисление MAC-кода на основании секретного кода, ассоциированного с этой страницей, всех 32 байт данных выбранной страницы, значения счетчика циклов записи страницы, номера страницы, регистрационного номера устройства (без CRC) и 3-байтного запроса, который выбирается из блокнота по адресам 20..22. Результат вычислений SHA помещается в блокнот по адресам 8..27, для того, чтобы его мог считать мастер. В то время, когда идут вычисления SHA, мастер считывает все единицы. Когда вычисления завершаются, последовательность сменяется чередующимися нулями и единицами. После этого мастер обычно должен прочитать все данные страницы и т.д., вычислить MAC-код (см. описание команды вычислений SHA, функцию проверки страницы данных), затем сравнить его с данными, находящимися в блокноте для определения того, содержит ли DS1963S правильный секретный код, ассоциированный с выбранной страницей данных.

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

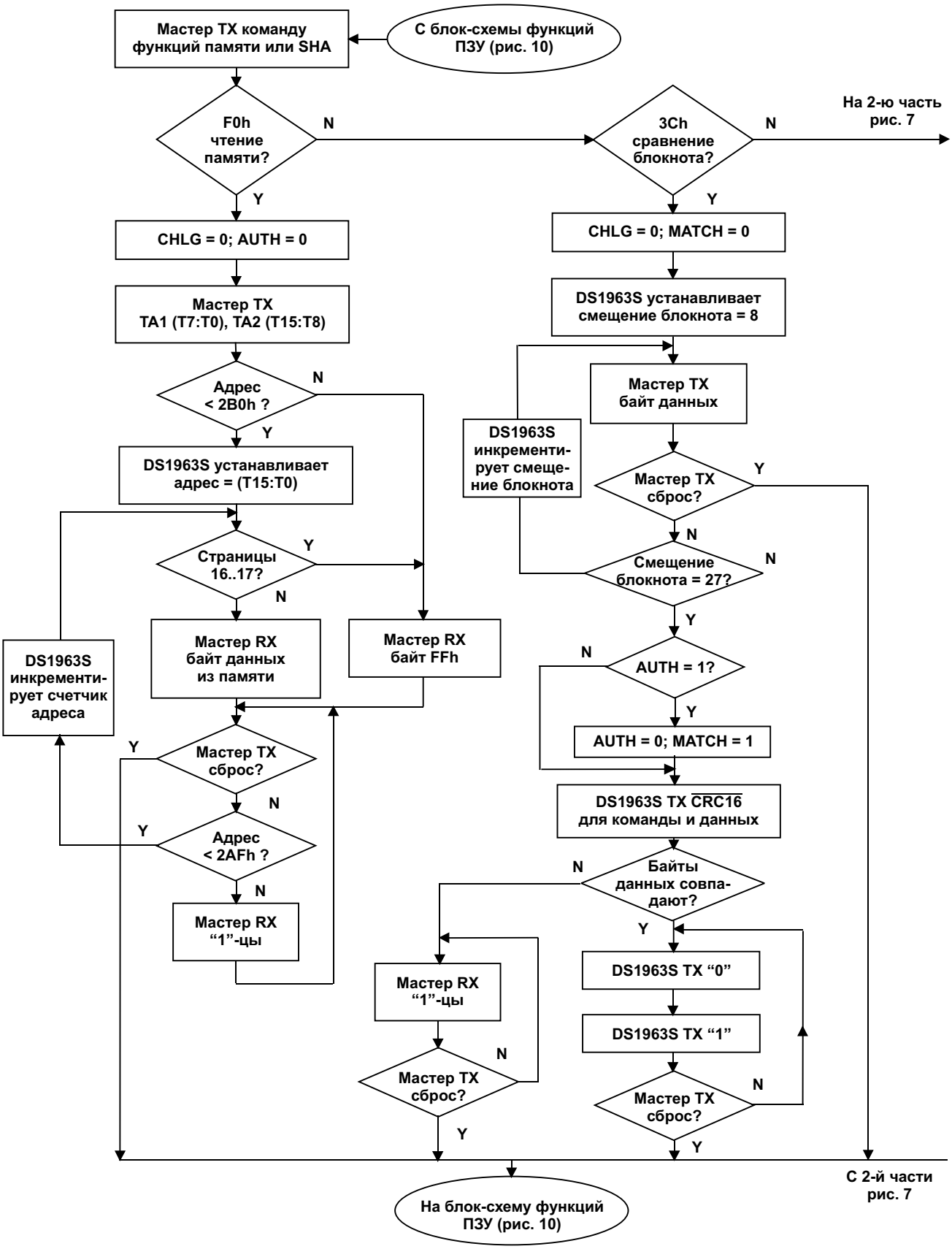


Рис. 7-2. БЛОК-СХЕМА ФУНКЦИЙ ПАМЯТИ И SHA (продолжение)

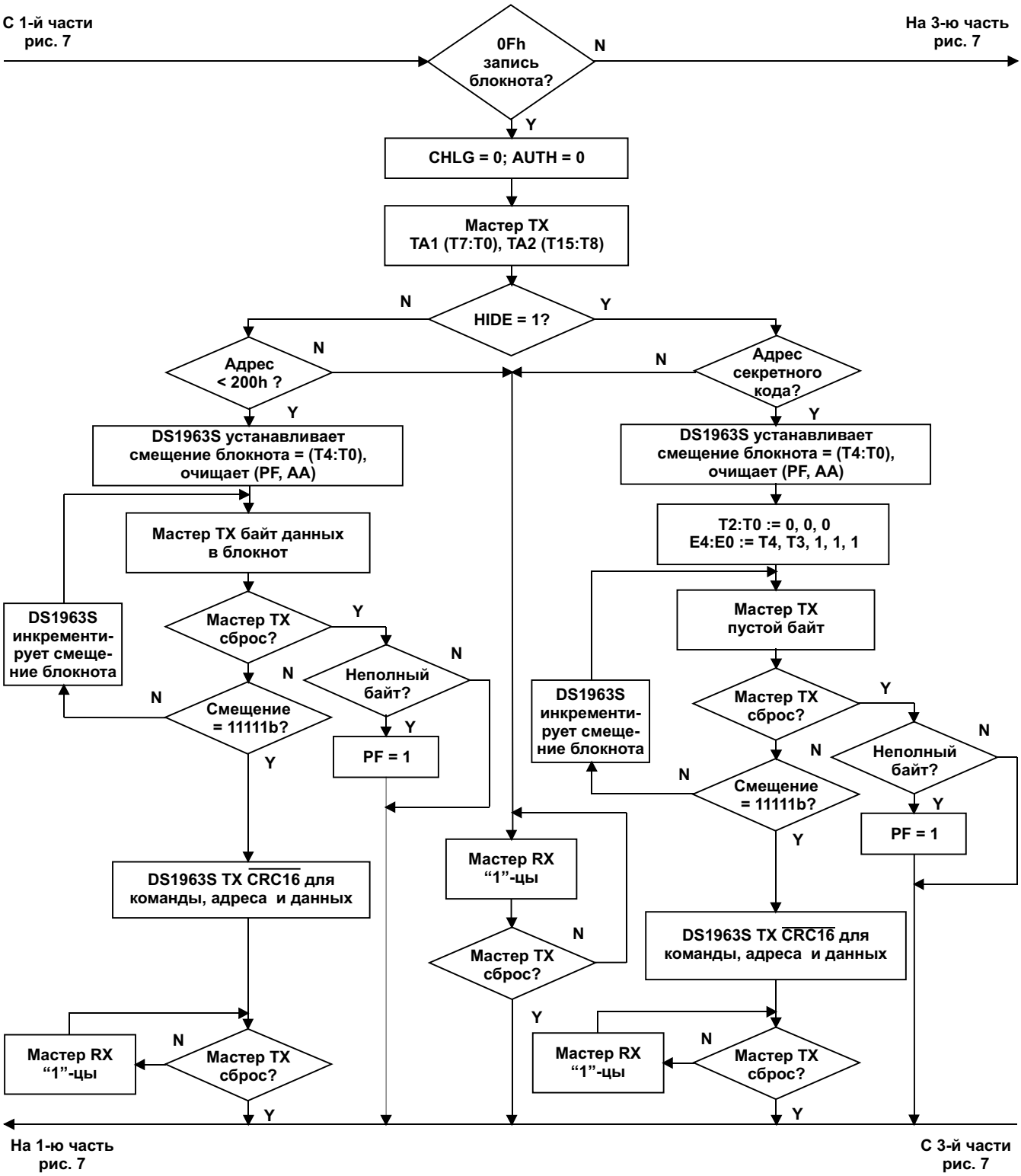


Рис. 7-3. БЛОК-СХЕМА ФУНКЦИЙ ПАМЯТИ И SHA (продолжение)

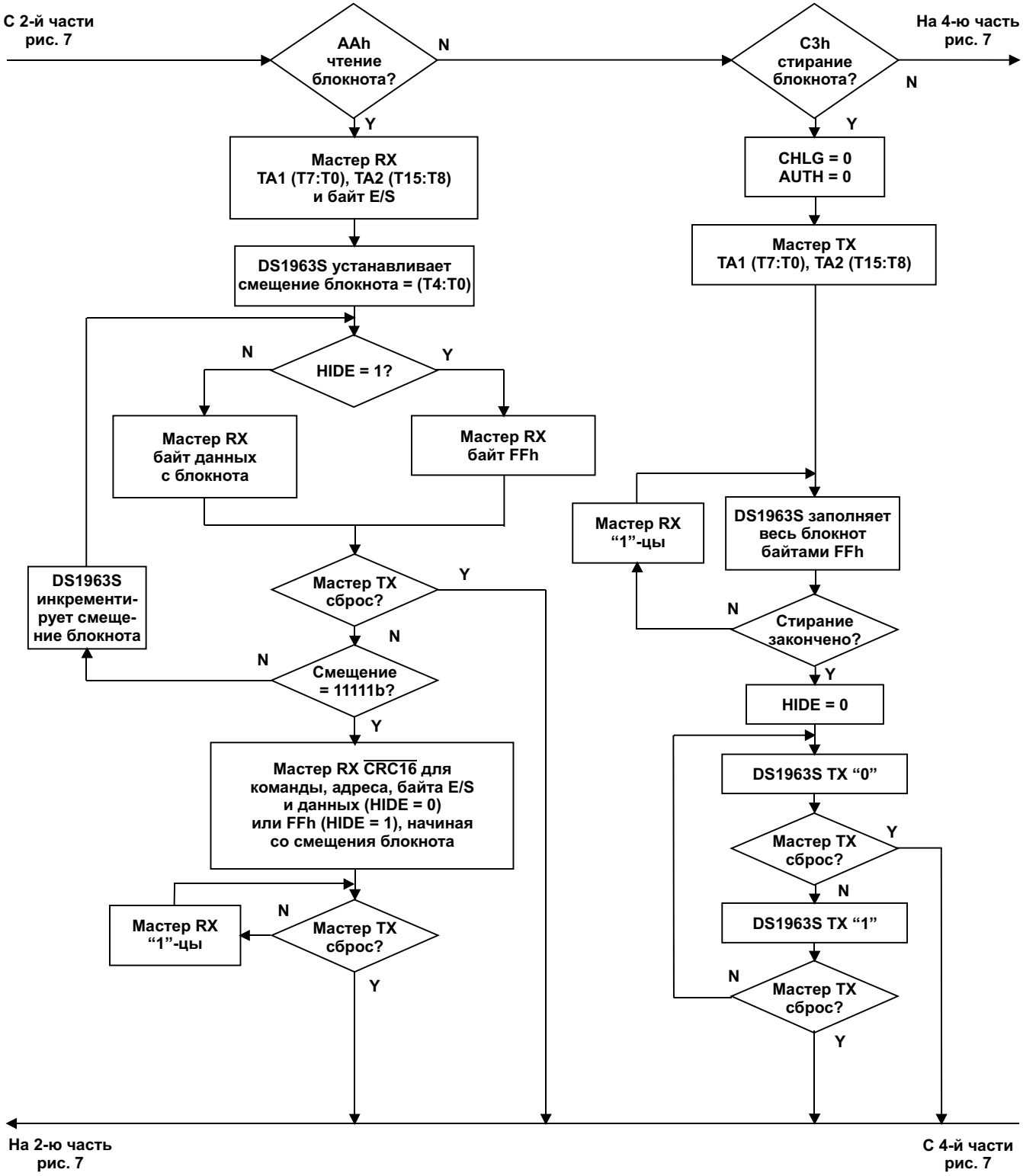


Рис. 7-4. БЛОК-СХЕМА ФУНКЦИЙ ПАМЯТИ И SHA (продолжение)

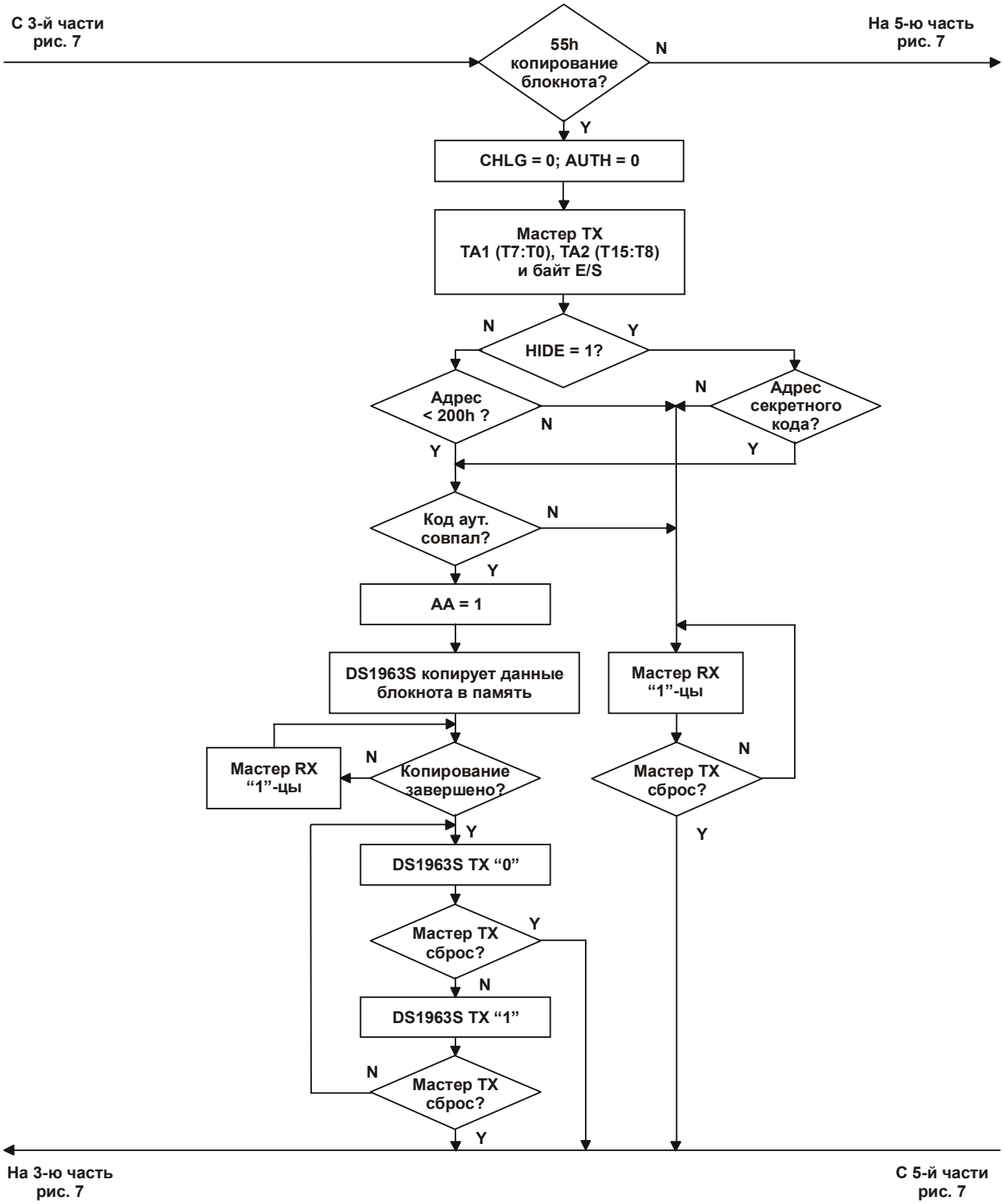


Рис. 7-5. БЛОК-СХЕМА ФУНКЦИЙ ПАМЯТИ И SHA (продолжение)

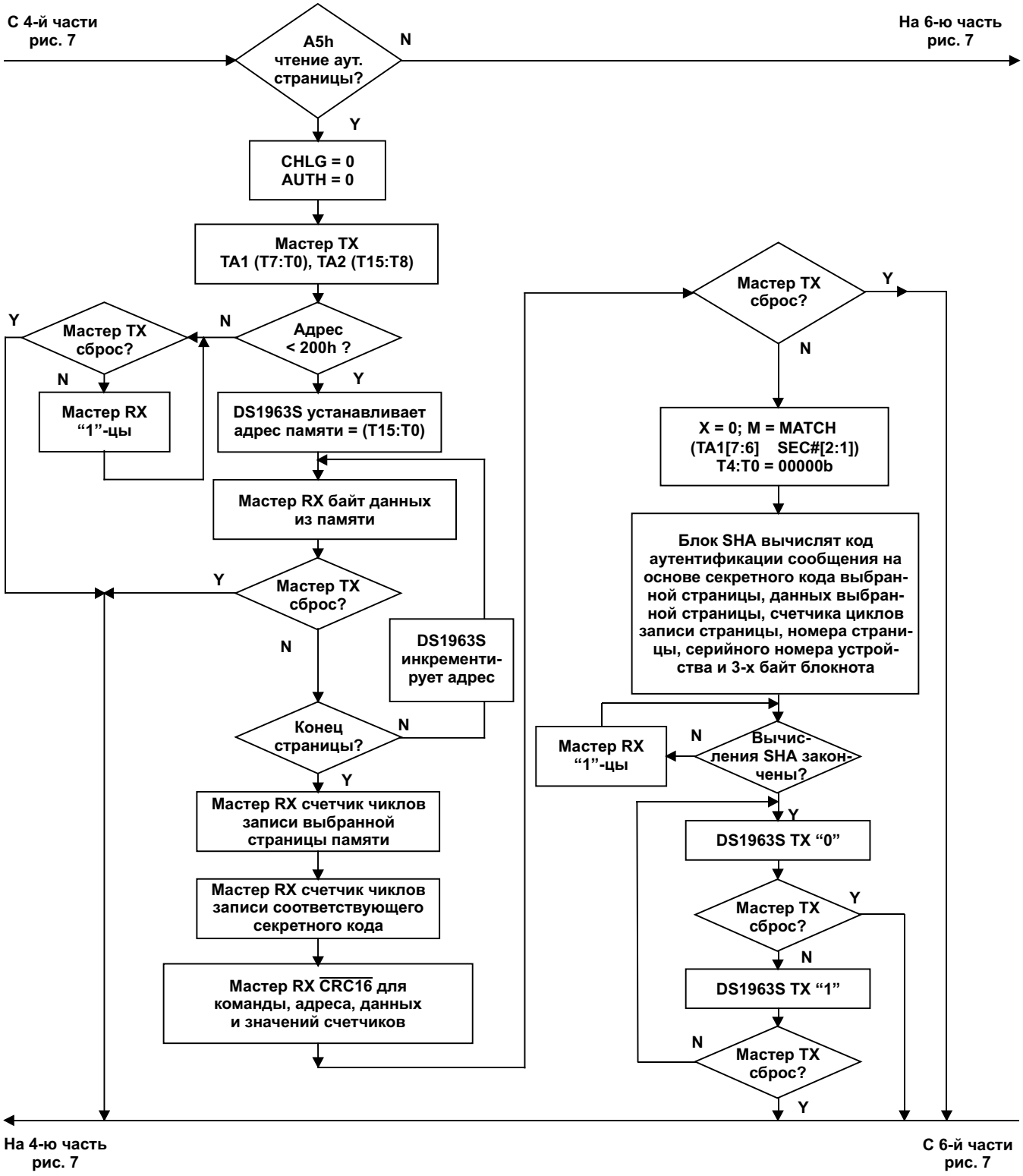
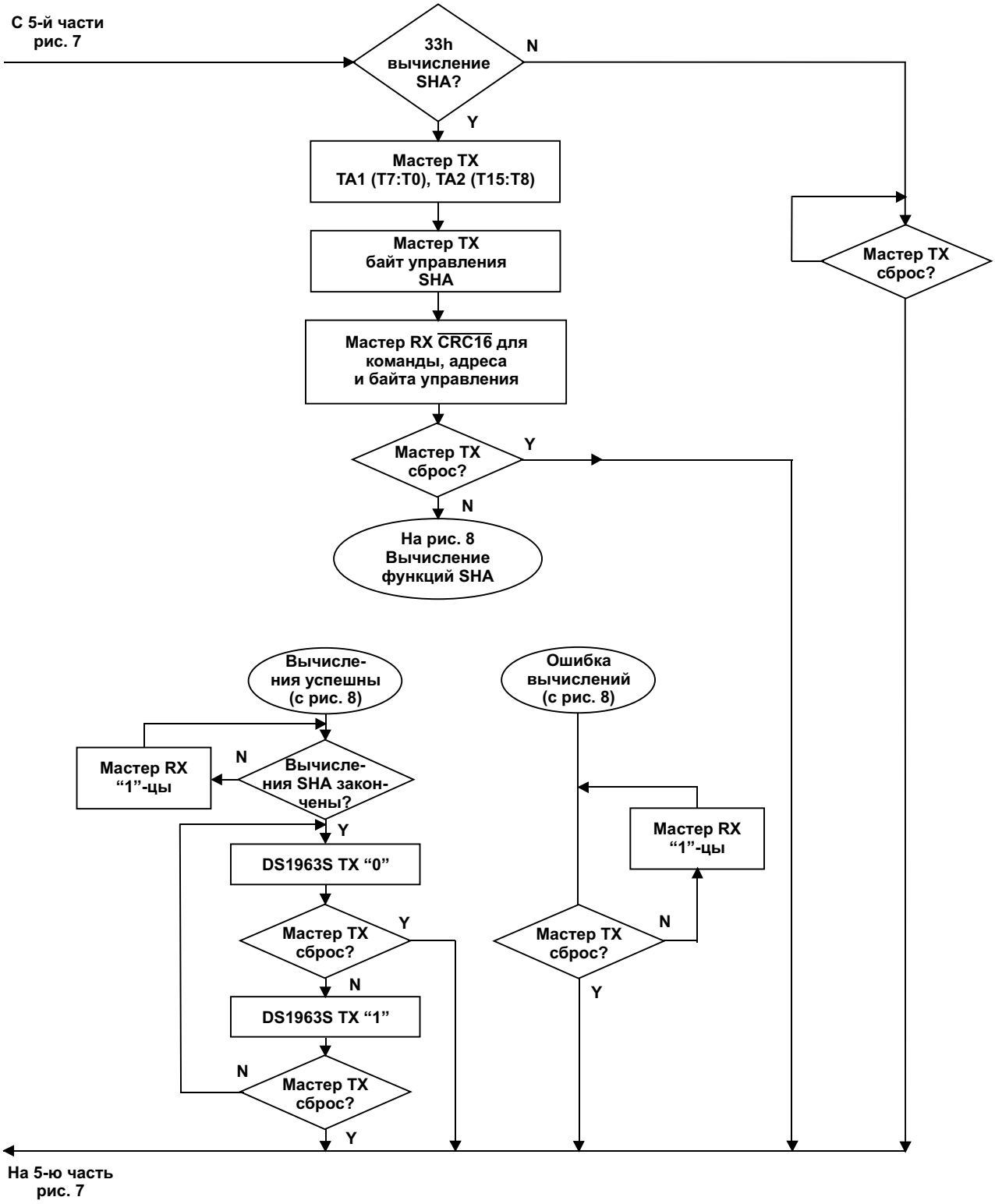


Рис. 7-6. БЛОК-СХЕМА ФУНКЦИЙ ПАМЯТИ И SHA (продолжение)



Команда вычислений SHA [33h]

Команда вычислений SHA обеспечивает выполнение шести функций, которые задействуют блок SHA для вычисления кода аутентификации сообщения в различных случаях. Седьмым способом задействования блока SHA является выполнение команды чтения аутентифицированной страницы, которая была описана немного раньше. В этом разделе приведено подробное описание всех вычислений SHA. В таблице 1 приведен обзор всех функций.

Таблица 1. ОБЗОР ФУНКЦИЙ SHA

Имя команды или функции	iButton роуминга	iButton сопроцессора	Используемые страницы
Чтение аутентифицированной страницы	да	нет	Страницы 0..15
Функция проверки страницы данных	нет	да	Страницы 0..15
Функция подписи страницы данных	нет	да	Только страница 0 и 8
Функция вычисления запроса	да	нет	Кроме страниц 0 и 8
Функция авторизации хоста	да	нет	Кроме страниц 0 и 8
Функция вычисления первого секретного кода	да	да	Страницы 0..15
Функция вычисления следующего секретного кода	да	да	Страницы 0..15

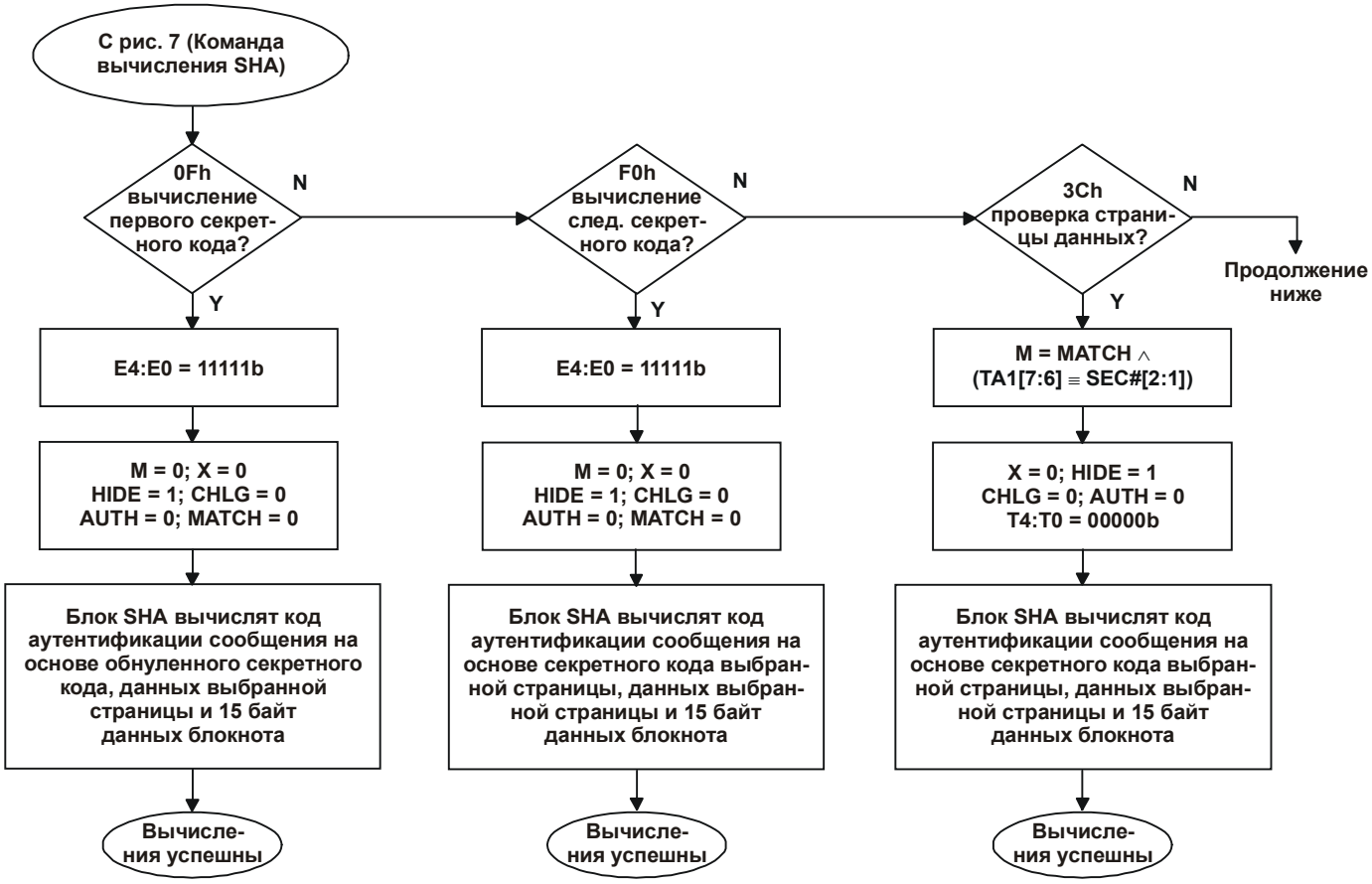
Устройство DS1963S может использоваться в системе двумя различными способами: а) как мобильный носитель данных, который закреплен за пользователем (устройство роуминга) и б) как сопроцессор и устройство безопасного хранения данных для хост-компьютера или «мастера шины». В любом случае требуется, чтобы в DS1963S был записан секретный код. Функции, которые требуются для установки секретных кодов за один или несколько шагов называются функциями вычисления первого секретного кода и вычисления следующего секретного кода. Когда DS1963S работает как сопроцессор, выполняются две функции: а) проверка принадлежности роумингового устройства системе (т.е. проверка правильности его секретного кода) и б) генерирование или проверка подписи, которая предохраняет данные от подделки. Эти действия выполняются функциями проверки страницы данных и подписи страницы данных.

Основной функцией SHA для роумингового устройства является чтение аутентифицированной страницы, когда сопроцессору предоставляются данные и MAC-код, требуемые для выполнения функции проверки страницы данных. Две оставшиеся функции SHA, которые может выполнять роуминговое устройство, являются функциями вычисления запроса и авторизации хоста. Эти функции не используются в таких приложениях, как торговые автоматы. Однако они необходимы для аутентификации пользователя или хоста, который желает установить флаг MATCH в роуминговом устройстве. Так как флаг MATCH используется для вычислений SHA при чтении аутентифицированной страницы, проверке страницы данных и подписи страницы данных, от него зависит полученный MAC-код. Поэтому он показывает, когда авторизация хоста прошла успешно. Авторизация пользователя или хоста, если она задействована, мешает использовать DS1963S в роли сопроцессора, так как при этом для установки флага MATCH требуется несколько шагов.

После выдачи кода команды мастер шины выбирает страницу памяти и соответствующий секретный код путем передачи адреса назначения, указывающего в любое место внутри страницы. После этого мастер передает байт управления SHA, который представляет собой код одной из шести функций SHA, которая должна быть выполнена. Затем мастер принимает CRC, рассчитанную для кода команды, адреса и байта управления. Когда CRC принята, а байт управления и адрес являются правильными, сразу запускается блок SHA, который вычисляет код аутентификации сообщения, как показано на рис. 8. В то время, когда идут вычисления SHA, мастер считывает все единицы. Когда вычисления завершаются, последовательность сменяется

чередующимися нулями и единицами. В случае неправильного байта управления или адреса мастер будет считывать все единицы вплоть до выдачи им импульса сброса. Точное расположение различных сегментов данных, как они попадают во входной буфер блока SHA, показано в таблице 2.

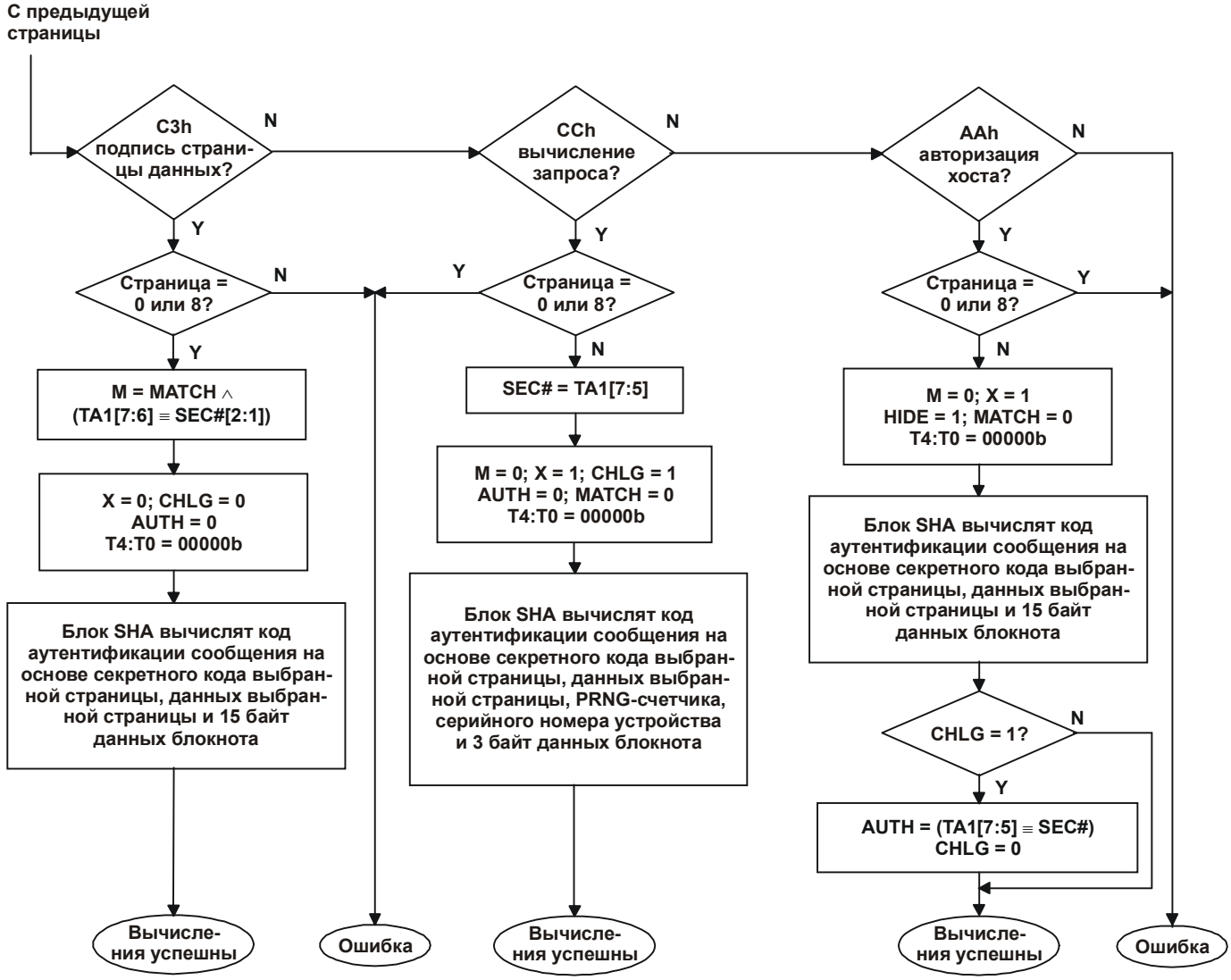
Рис. 8. ФУНКЦИИ ВЫЧИСЛЕНИЯ SHA



На рис. 7 (Команда вычисления SHA)

Чтение аутентифицированной страницы и вычисление запроса позволяют мастеру ввести в вычисления 3-байтный запрос, используя адреса блокнота 20..22. В качестве остальных данных используются данные из выбранной страницы памяти, соответствующий секретный код, значение счетчика циклов, регистрационный номер из ПЗУ и флаги. При вычислении первого секретного кода и вычислении следующего секретного кода перед началом вычислений SHA по адресам блокнота 8..22 должен быть помещен частичный секретный код. Сопроцессор, выполняя команды проверки страницы данных или подписи страницы данных, должен иметь по адресам блокнота 8..11 инкрементированное значение счетчика циклов для выбранной страницы памяти роумингового устройства, по адресам 13..19 - регистрационный номер из ПЗУ, а по адресу 12 - номер страницы. Роуминговое устройство перед выполнением команды авторизации хоста должно выполнить функцию вычисления запроса для заполнения блокнота псевдослучайными данными.

Рис. 8. ФУНКЦИИ ВЫЧИСЛЕНИЯ SHA (продолжение)



На рис. 7 (Команда вычисления SHA)

Функция вычисления запроса сохраняет 3 старших бита TA1 в регистре SEC#, который затем используется функцией авторизации хоста. Флаг AUTH будет установлен только в том случае, когда авторизация хоста и вычисление запроса вызываются для одной и той же страницы памяти (одного и того же секретного кода). Это предохраняет флаг AUTH от установки с секретным кодом другой страницы, которая может принадлежать другому приложению или провайдеру услуг.

Два старших бита регистра SEC# также используются при проверке страницы данных, подписи страницы данных и чтении аутентифицированной страницы, когда требуется определить бит управления M. Это имеет значение только для тех приложений, которые используют авторизацию хоста/пользователя. Бит управления M устанавливается только в том случае, если установлен флаг MATCH, а страница памяти назначения является смежной с той, которая была использована для авторизации. Это назначает одну пару секретных кодов (0 и 1, 2 и 3, 4 и 5, 6 и 7) и страниц, соответствующих этим кодам, одному провайдеру услуг.

Таблица 2. ФОРМАТЫ ВХОДНЫХ ДАННЫХ SHA-1

Команда чтения аутентифицированной страницы, функция вычисления запроса

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] = (CC+0)	M9[23:16] = (CC+1)	M9[15:8] = (CC+2)	M9[7:0] = (CC+3)
M10[31:24] = MP	M10[23:16] = FAMC	M10[15:8] = SN0	M10[7:0] = SN1
M11[31:24] = SN2	M11[23:16] = SN3	M11[15:8] = SN4	M11[7:0] = SN5
M12[31:24] = (SS+4)	M12[23:16] = (SS+5)	M12[15:8] = (SS+6)	M12[7:0] = (SS+7)
M13[31:24] = (SP+20)	M13[23:16] = (SP+21)	M13[15:8] = (SP+22)	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	**Начальный адрес секретного кода** См. рис. 5, карту памяти, страницы памяти 16 и 17
CC	Начальный адрес счетчика циклов *Команда чтения аутентифицированной страницы:* счетчик циклов записи выбранной страницы памяти, см. рис. 5, карту памяти, страницу памяти 19; младший байт счетчика хранится по меньшему адресу *Функция вычисления запроса:* PRNG-счетчик; см. рис. 5, карту памяти, страницу памяти 21; младший байт счетчика хранится по меньшему адресу
PP	**Начальный адрес страницы памяти** См. рис. 5, карту памяти, страницы памяти 0..15
FAMC	**Код семейства = 18h**
MP	MP[7] = бит управления M, см. рис. 7, чтение аутентифицированной страницы и рис. 8 MP[6] = бит управления X, см. рис. 7, чтение аутентифицированной страницы и рис. 8 MP[5:4] = 00b MP[3:0] = T8:T5 (эквивалент номера страницы в hex-формате)
SNx	**Серийный номер устройства из ПЗУ** SN0 = младший байт, SN5 = старший байт CRC не используется
(SP + n)	**Байт n блокнота** число n представлено в десятичном виде

Таблица 2. ФОРМАТЫ ВХОДНЫХ ДАННЫХ SHA-1 (продолжение)

Проверка страницы данных, подпись страницы данных, авторизация хоста, вычисление первого секретного кода, вычисление 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] = (SP+8)	M9[23:16] = (SP+9)	M9[15:8] = (SP+10)	M9[7:0] = (SP+11)
M10[31:24] = MPX	M10[23:16] = (SP+13)	M10[15:8] = (SP+14)	M10[7:0] = (SP+15)
M11[31:24] = (SP+16)	M11[23:16] = (SP+17)	M11[15:8] = (SP+18)	M11[7:0] = (SP+19)
M12[31:24] = (SS+4)	M12[23:16] = (SS+5)	M12[15:8] = (SS+6)	M12[7:0] = (SS+7)
M13[31:24] = (SP+20)	M13[23:16] = (SP+21)	M13[15:8] = (SP+22)	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	**Начальный адрес секретного кода** См. рис. 5, карту памяти, страницы памяти 16 и 17 При вычислении первого секретного кода секретные данные заменяются всеми нулями
PP	**Начальный адрес страницы памяти** См. рис. 5, карту памяти, страницы памяти 0..15
MPX	MPX[7] = бит управления M, см. рис. 8 MPX[6] = бит управления X, см. рис. 8 MPX[5:0] = (SP+12)[5:0]
(SP + n)	**Байт n блокнота** число n представлено в десятичном виде

Функции SHA, также как и функции памяти, используют несколько флагов, которые влияют как на выполнение самой функции, так и на результаты функций, выполняемых на следующих шагах. Этими флагами являются HIDE, CHLG, AUTH и MATCH. В таблице 3 представлены операции с этими флагами. Единственной командой, которая не изменяет флаги, является команда чтения блокнота. Заметьте, что начальный сброс, производимый схемой с паразитным питанием, также влияет на флаги. Условие «возобновление контакта со считывателем» выполняется при контакте DS1963S со считывателем хост-компьютера или мастера шины, или если контакт прерывается. Наиболее значимым является флаг HIDE. Если он установлен, это защищает данные блокнота от считывания; текущее значение адреса назначения и байт E/S, тем не менее, остаются доступными для чтения. Флаг HIDE также влияет на команды записи блокнота и копирования блокнота. Три других флага используются только в специальных случаях и остаются сброшенными большую часть времени. Флаги CHLG и AUTH работают в паре при авторизации хоста/пользователя для подтверждения того, что команды выполнялись в определенной последовательности. Если последовательность правильная, и следующая команда сравнения блокнота не обнаруживает различий данных, устанавливается флаг MATCH. Этот флаг может влиять на проверку страницы данных, подпись страницы данных или чтение аутентифицированной страницы.

Таблица 3. ОПЕРАЦИИ С ФЛАГАМИ

Команда, функция или условие	HIDE	CHLG	AUTH	MATCH
Условие возобновления контакта со считывателем	Установка	-----	-----	-----
Команда чтения памяти	-----	Сброс	Сброс	-----
Команда сравнения блокнота	-----	Сброс	Сброс	Прим. 1)
Команда записи блокнота	-----	Сброс	Сброс	-----
Команда чтения блокнота	-----	-----	-----	-----
Команда стирания блокнота	Сброс	Сброс	Сброс	-----
Команда копирования блокнота	-----	Сброс	Сброс	-----
Команда чтения аутентифицированной страницы	-----	Сброс	Сброс	-----
Функция проверки страницы данных	Установка	Сброс	Сброс	-----
Функция подписи страницы данных	-----	Сброс	Сброс	-----
Функция вычисления запроса	-----	Установка	Сброс	Сброс
Функция авторизации хоста	Установка	Сброс	Прим. 2)	Сброс
Функция вычисления первого секретного кода	Установка	Сброс	Сброс	Сброс
Функция вычисления следующего секретного кода	Установка	Сброс	Сброс	Сброс

1) Флаг **устанавливается**, если данные совпадают, и флаг AUTH был установлен предыдущей командой; иначе флаг **сбрасывается**. Установка флага MATCH требует успешного выполнения вычисления запроса, авторизации хоста и сравнения блокнота в виде непрерывной последовательности.
2) **Устанавливается** только в том случае, если флаг CHLG был установлен предыдущей командой; иначе флаг **сбрасывается**.

АЛГОРИТМ ВЫЧИСЛЕНИЙ SHA-1

Данное описание алгоритма вычисления SHA является адаптированным вариантом документа под названием «Secure Hash Standard SHA-1», ссылка на который была дана на стр. 3. Алгоритм использует в качестве входных данных шестнадцать 32-битных слов M_t ($0 \le t \le 15$), как показано в таблице 2. 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

MAC-код загружается в блокнот DS1963S двумя различными способами, в зависимости от выбранной функции SHA. При вычислении первого секретного кода и вычислении следующего секретного кода используются 64 бита MAC-кода в виде повторяющихся последовательностей, каждая из которых может быть скопирована в любой из восьми секретных кодов. Все другие функции SHA загружают в блокнот полный 160-битный результат. В таблице 4 показано размещение байтов в блокноте.

Таблица 4. ФОРМАТЫ ВЫХОДНЫХ ДАННЫХ SHA-1

Частичный код (только вычисление первого секретного кода и следующего секретного кода)

(SP+0) := E[7:0]	(SP+1) := E[15:8]	(SP+2) := E[23:16]	(SP+3) := E[31:24]
(SP+4) := D[7:0]	(SP+5) := D[15:8]	(SP+6) := D[23:16]	(SP+7) := D[31:24]
(SP+8) := E[7:0]	(SP+9) := E[15:8]	(SP+10) := E[23:16]	(SP+11) := E[31:24]
(SP+12) := D[7:0]	(SP+13) := D[15:8]	(SP+14) := D[23:16]	(SP+15) := D[31:24]
(SP+16) := E[7:0]	(SP+17) := E[15:8]	(SP+18) := E[23:16]	(SP+19) := E[31:24]
(SP+20) := D[7:0]	(SP+21) := D[15:8]	(SP+22) := D[23:16]	(SP+23) := D[31:24]
(SP+24) := E[7:0]	(SP+25) := E[15:8]	(SP+26) := E[23:16]	(SP+27) := E[31:24]
(SP+28) := D[7:0]	(SP+29) := D[15:8]	(SP+30) := D[23:16]	(SP+31) := D[31:24]

Полный 160-битный код (все остальные функции SHA)

(SP+8) := E[7:0]	(SP+9) := E[15:8]	(SP+10) := E[23:16]	(SP+11) := E[31:24]
(SP+12) := D[7:0]	(SP+13) := D[15:8]	(SP+14) := D[23:16]	(SP+15) := D[31:24]
(SP+16) := C[7:0]	(SP+17) := C[15:8]	(SP+18) := C[23:16]	(SP+19) := C[31:24]
(SP+20) := B[7:0]	(SP+21) := B[15:8]	(SP+22) := B[23:16]	(SP+23) := B[31:24]
(SP+24) := A[7:0]	(SP+25) := A[15:8]	(SP+26) := A[23:16]	(SP+27) := A[31:24]

1-ПРОВОДНАЯ ШИНА

1-проводная шина представляет собой систему, в которой имеется один мастер шины и одно или несколько подчиненных устройств. Во всех случаях DS1963S является подчиненным устройством. Мастером шины обычно является микроконтроллер или PC. Для небольших систем сигналы 1-проводного протокола могут генерироваться программно, используя один вывод порта. Для более крупных систем рекомендуется использовать микросхему драйвера однопроводной линии DS2480B или адаптеры для последовательного порта, построенные на основе этой микросхемы (серия DS9097U). Это упрощает аппаратную часть и освобождает микропроцессор от необходимости реагирования на события в реальном 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 состояний шины во время специальных временных интервалов, которые начинаются спадом импульса синхронизации, выдаваемого мастером. Более детальное описание протокола приведено в главе 4 книги *«Book of DS19xx iButton Standards»*.

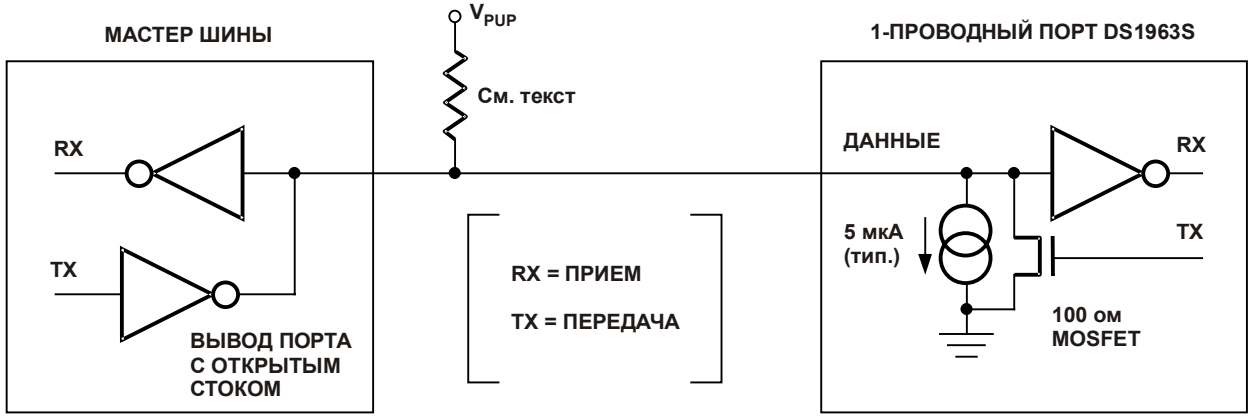
АППАРАТНАЯ КОНФИГУРАЦИЯ

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. DS1963S имеет выход с открытым стоком, его внутренняя схема эквивалентна показанной на рис. 9.

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. DS1963S не гарантирует полной совместимости со стандартом iButton, так как для DS1963S максимальная скорость обмена составляет 15,4 Кбит в секунду в стандартном режиме и 125 Кбит в секунду в ускоренном режиме. Значение номинала подтягивающего резистора зависит от протяженности сети и от величины нагрузки. Для большинства приложений подходит номинал 2,2 Ком.

В состоянии покоя на линии 1-проводной шины присутствует высокий уровень. Если по каким-либо причинам пересылка должна быть приостановлена, линию **следует** 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 DS1963S при повышенной скорости линия не должна находиться в состоянии низкого уровня дольше, чем 15,4 мкс, чтобы быть уверенным в том, что ни одно устройство не будет сброшено. Несмотря на такую неполную совместимость, DS1963S корректно работает в паре с драйвером 1-проводной шины DS2480B и с адаптерами последовательного порта, построенными на основе этой микросхемы.

Рис. 9. АППАРАТНАЯ КОНФИГУРАЦИЯ



ПОСЛЕДОВАТЕЛЬНОСТЬ ПЕРЕСЫЛКИ

Последовательность действий для доступа к DS1963S 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 DS1963S. Все команды функций ПЗУ имеют длину 8 бит. Список этих команд приведен ниже (см. блок-схему на рис. 10).

Чтение ПЗУ [33h]

Эта команда позволяет мастеру шины считать из DS1963S 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 будут неправильными.

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

DS1963S, 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.

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 методом исключения для 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 желаемого 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»).

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 DS1963S 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»).

Рис. 10-1. БЛОК-СХЕМА ФУНКЦИЙ ПЗУ

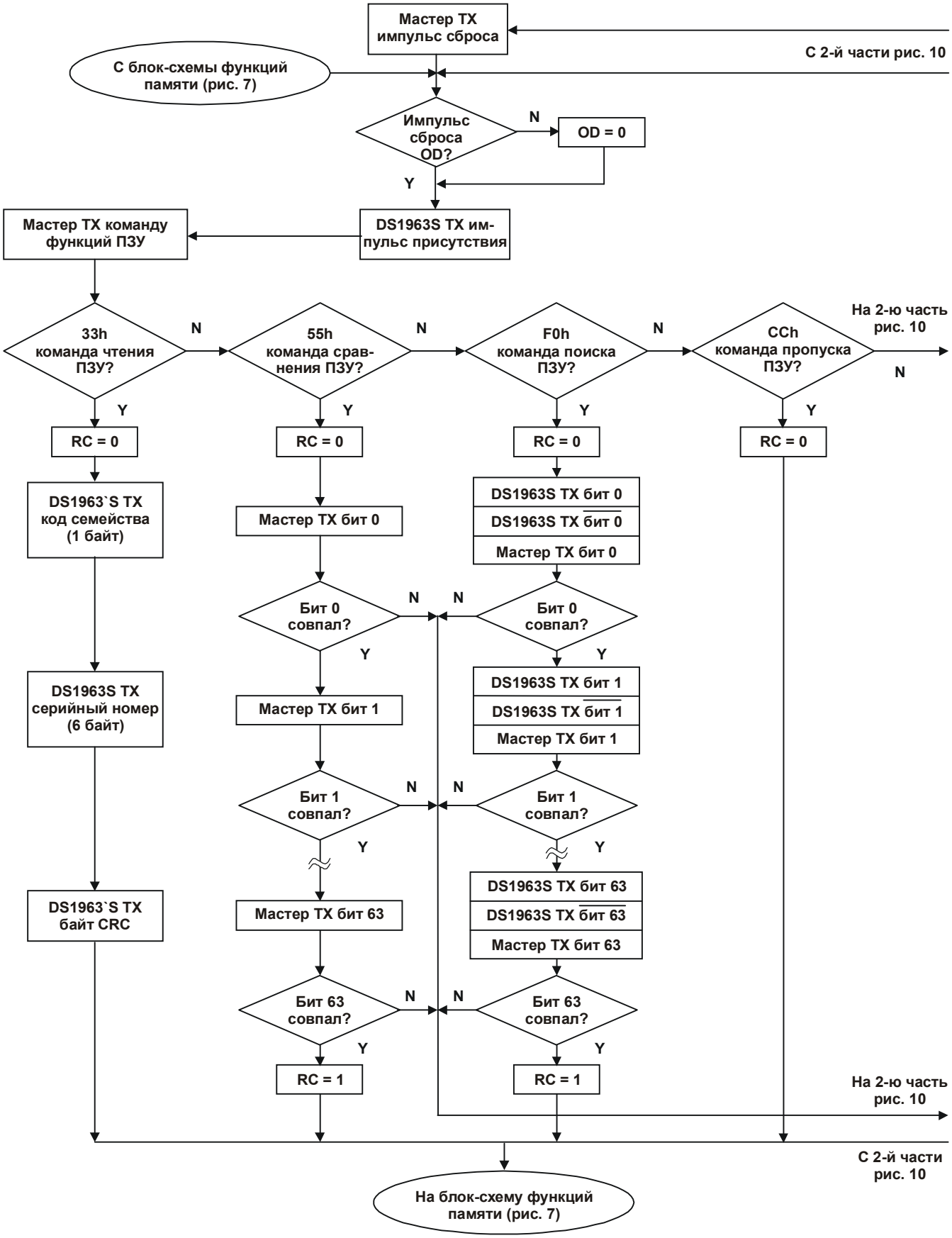
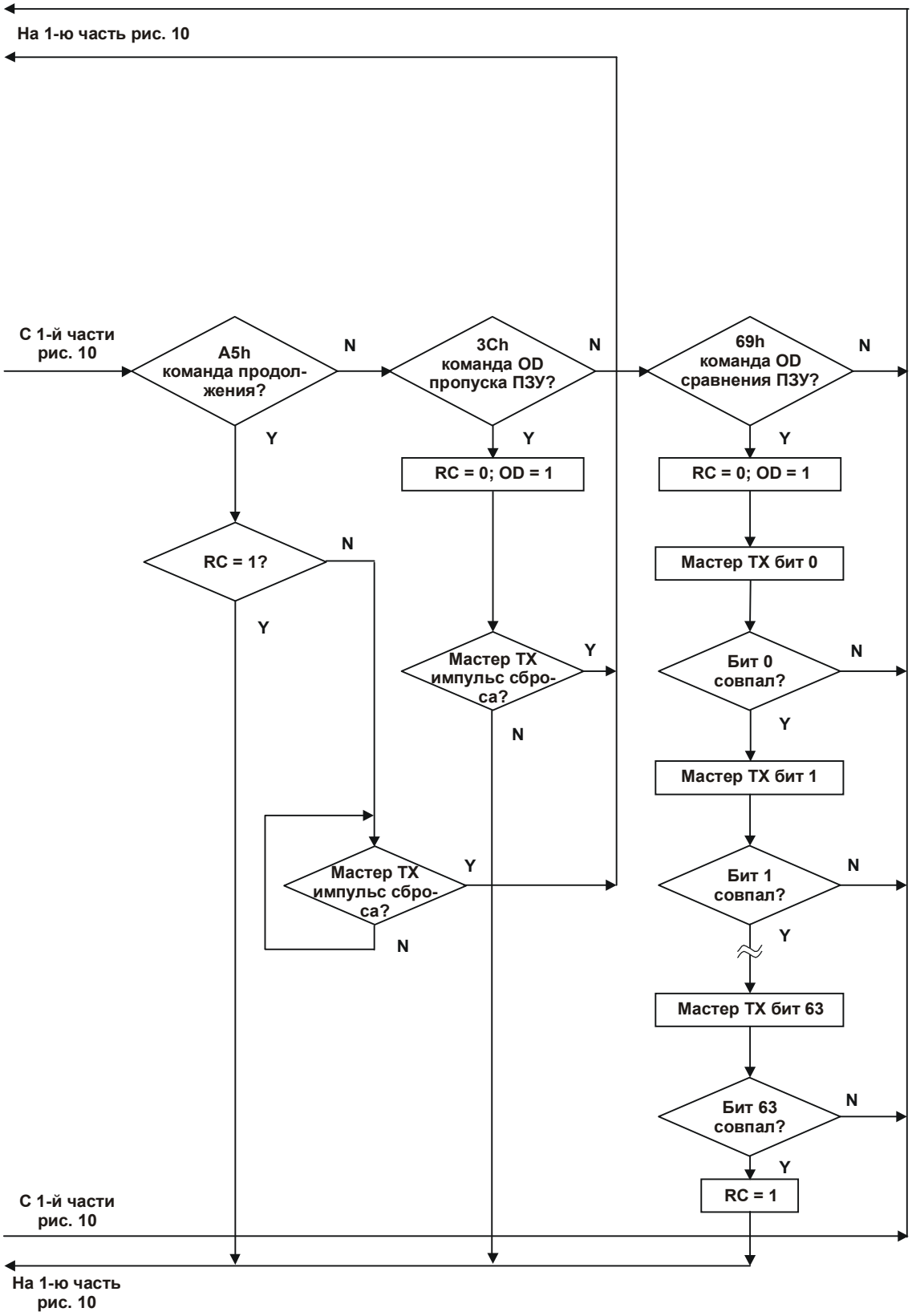


Рис. 10-2. БЛОК-СХЕМА ФУНКЦИЙ ПЗУ (продолжение)



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 DS1963S, 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.

Команда продолжения [A5h]

Обычно для полного проведения электронного платежа требуется получить доступ к DS1963S несколько раз. Количество необходимых операций доступа возрастает еще больше, когда осуществляется авторизация хоста/пользователя. 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.

СИГНАЛЫ 1-ПРОВОДНОЙ ШИНЫ

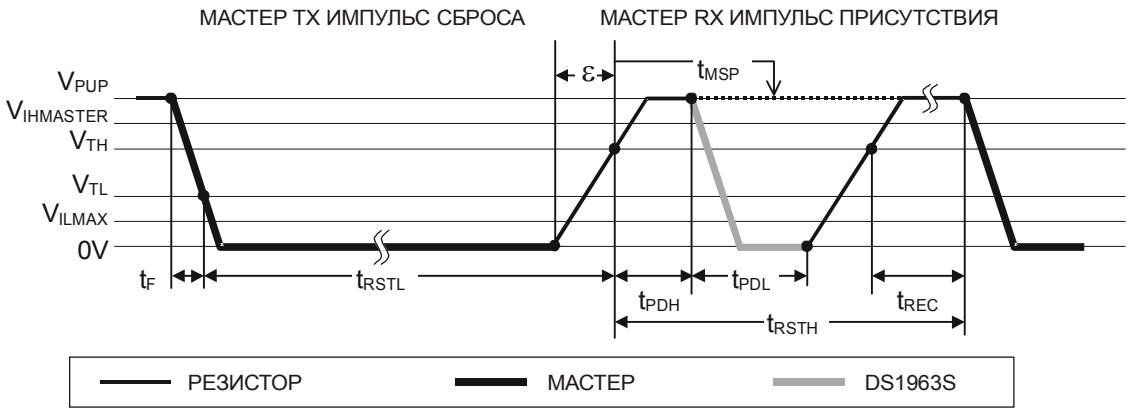
DS1963S требует строгого соблюдения протокола для гарантии целостности данных. Протокол содержит четыре типа сигналов: последовательность сброса с импульсом сброса и импульсом присутствия, запись нуля, запись единицы и чтение данных. Все эти сигналы, за исключением импульса присутствия, инициируются мастером шины. DS1963S имеет возможность вести обмен на двух разных скоростях: стандартной скорости и повышенной скорости в ускоренном режиме. Если устройство специально не переведено в ускоренный режим, DS1963S 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 DS1963S, показана на рис. 11. За импульсом сброса следует импульс присутствия, который говорит о готовности DS1963S 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. Эта длительность составляет 480 мкс на стандартной скорости и 48 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 DS1963S находится в ускоренном режиме, и длительность t_{RSTL} не превышает 80 мкс, устройство остается в ускоренном режиме.

Рис. 11. 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}, DS1963S 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 DS1963S 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. 12.

Любой интервал начинается с того, что мастер переводит линию в состояние низкого уровня. Как только напряжение на линии упадет ниже порога V_{TL}, в DS1963S 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 DS1963S 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}.

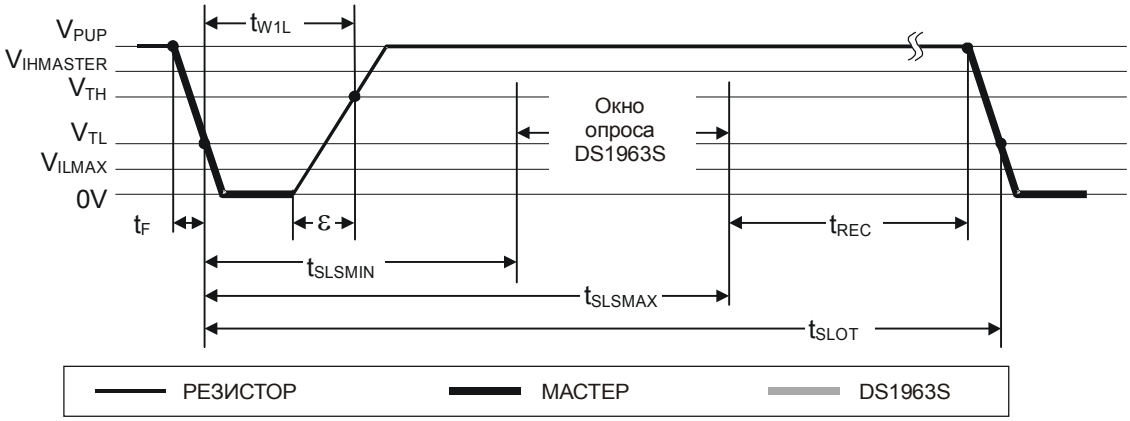
Передача данных от мастера к подчиненному устройству

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 DS1963S. После самой дальней точки опроса (t_{SLSMAX}) перед началом следующего временного интервала требуется время восстановления (t_{REC}).

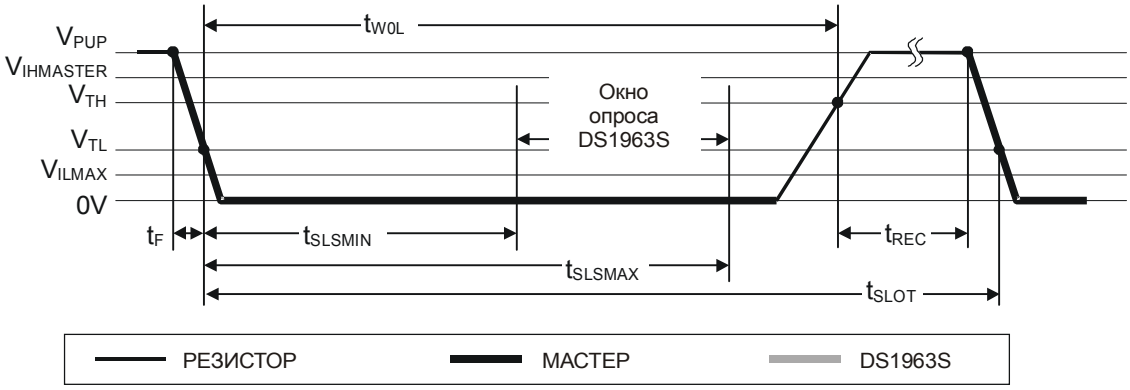
Для временного интервала **записи нуля** время удержания мастером низкого уровня ($t_{MPD0} = t_{W0L} + t_F$) должно быть достаточно большим, чтобы сохранить напряжение на линии ниже значения V_{ILMAX} до самой дальней точки опроса DS1963S в момент t_{SLSMAX}. Перед началом следующего временного интервала напряжение на линии данных должно подняться выше V_{TH} и оставаться таким в течение времени восстановления t_{REC}.

Рис. 12. ВРЕМЕННАЯ ДИАГРАММА ЗАПИСИ/ЧТЕНИЯ

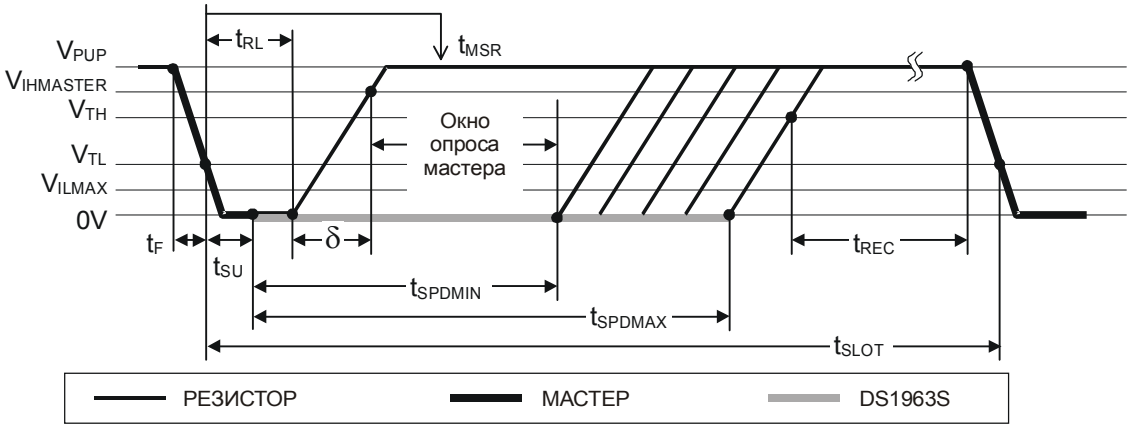
Временной интервал записи единицы



Временной интервал записи нуля



Временной интервал чтения данных



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 DS1963S 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 DS1963S выдает бит данных на 1-проводный порт. Если передается 0, DS1963S удерживает линию данных в состоянии низкого уровня в течение времени t_{SPD}. Если бит данных равен 1, DS1963S 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 $t_{SU} + t_{SPDMIN}$ с другой. Оптимальное положение точки опроса в случае **чтения нуля** находится не позднее момента $t_{SU} + t_{SPDMIN}$. 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}.

ВЫЧИСЛЕНИЕ CRC

DS1963S 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 DS1963S. 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, вычислении SHA, чтении блокнота и для быстрой проверки правильности пересылки данных при записи блокнота. Этот же тип CRC используется в iButton с энергонезависимой памятью в рамках расширенной файловой структуры. В отличие от 8-битной CRC, 16-битная CRC всегда считывается и передается в инвертированном виде. Внутренний генератор CRC в DS1963S (рис. 13) вычисляет новое значение 16-битной CRC в соответствии с блок-схемой команд, 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 генерация CRC начинается очисткой сдвигового регистра генератора CRC. Затем по одному биту в сдвиговый регистр вводится код команды, адрес назначения TA1 и TA2, а также все байты данных, переданные мастером. DS1963S передает эту CRC только в том случае, если записанные в блокнот данные достигли конечного смещения, равного 11111b. Данные могут начинаться с любой позиции блокнота. Этот алгоритм используется независимо от состояния флага HIDE. Однако если флаг HIDE установлен, байты данных, которые следуют за адресом назначения, используются только для подсчета CRC и не записываются в блокнот.

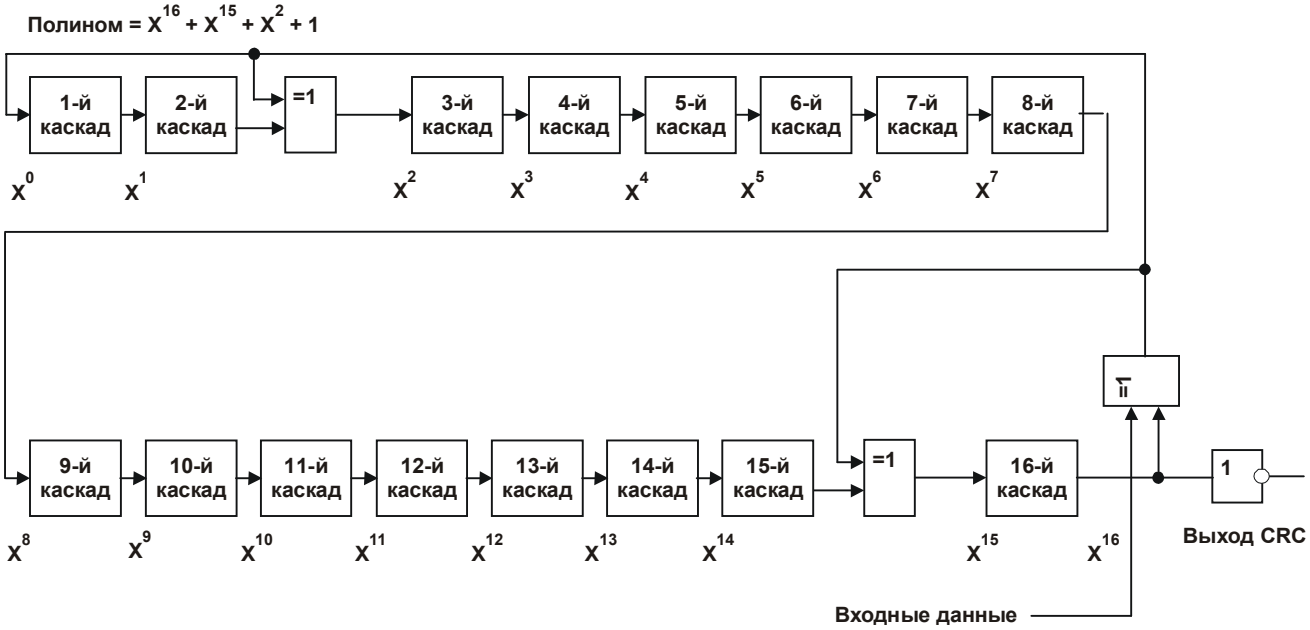
При чтении блокнота генерация CRC также начинается очисткой сдвигового регистра генератора CRC. Затем по одному биту в сдвиговый регистр вводится код команды, адрес назначения TA1 и TA2, байт E/S, а также данные блокнота, начиная со смещения блокнота. DS1963S будет передавать эту CRC только в том случае, если мастер прочитает блокнот до конца, независимо от

значения конечного смещения. Если установлен флаг HIDE, при вычислении CRC используются байты FFh вместо данных блокнота, которые остаются скрытыми.

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 значения счетчиков количества циклов записи для адресованной страницы памяти и соответствующего секретного кода. Для счетчиков количества циклов записи первым сдвигается младший байт. При выполнении команды вычисления SHA CRC получается путем сдвига в предварительно очищенный генератор CRC байта команды, за которым следует адрес назначения TA1 и TA2, а также байт управления SHA.

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

Рис. 13. АППАРАТНАЯ РЕАЛИЗАЦИЯ И ПОЛИНОМ ВЫЧИСЛЕНИЯ CRC-16



ФИЗИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Размер	см. чертеж корпуса
Вес	3,3 грамма
Допустимая относительная влажность	90% при +50°C
Допустимая высота над уровнем моря	3000 м
Ожидаемый срок службы	см. график
Условия безопасности	Соответствует UL#913 (4-я редакция); взрывобезопасное исполнение, утверждено для использования в классе I, раздел 1, группы A, B, C и D

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	–25°C .. +50°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. Устройство не должно подвергаться воздействию температур выше +70°C в течение длительного периода времени.*

ЭЛЕКТРИЧЕСКИЕ ХАРАКТЕРИСТИКИ (V_{PUP} = 2,8В .. 5,25В, T_A = -40°C .. +85°C)

ПАРАМЕТР	СИМВ.	УСЛОВИЯ	МИН.	ТИП.	МАКС.	ЕД.	ПРИМ.
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}			9	мкА	4
Пороговое напряжение при переходе из 1 в 0	V_{TL}		0,7		2,9	В	5, 6, 7
Входное напряжение низкого уровня	V_{IL}				0,30	В	1, 5, 8
Пороговое напряжение при переходе из 0 в 1	V_{TH}		0,6		2,9	В	5, 6, 9
Выходное напряжение низкого уровня при токе 4 мА	V_{OL}				0,4	В	5, 10
Длительность спада	t_F				5	мкс	1

ПАРАМЕТР	СИМВ.	УСЛОВИЯ	МИН.	ТИП.	МАКС.	ЕД.	ПРИМ.
Время восстановления	t_{REC}	Стандартная скорость, R_{PUP} = 2,2 Ком	5			мкс	1, 15
		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	69			мкс	1, 15
		Стандартная скорость, –20°C .. +85°C	65				
		Повышенная скорость, V_{PUP} > 4,5 В	8				
ВХОД/ВЫХОД, ЦИКЛ СБРОСА И ПОЛУЧЕНИЯ ИМПУЛЬСА ПРИСУТСТВИЯ							
Длительность низкого уровня сброса	t_{RSTL}	Стандартная скорость	540		960	мкс	1, 14
		Стандартная скорость, –20°C .. +85°C	480		960		
		Повышенная скорость, V_{PUP} > 4,5 В	48		80		
Длительность высокого уровня импульса присутствия	t_{PDH}	Стандартная скорость	17		60	мкс	14
		Повышенная скорость, V_{PUP} > 4,5 В	1,8		6		
Длительность низкого уровня импульса присутствия	t_{PDL}	Стандартная скорость	78		260	мкс	14
		Стандартная скорость, –20°C .. +85°C	78		240		
		Повышенная скорость, V_{PUP} > 4,5 В	7,7		24		
Время опроса импульса присутствия	t_{MSP}	Стандартная скорость	60		95	мкс	1
		Повышенная скорость, V_{PUP} > 4,5 В	6		9,5		
ВХОД/ВЫХОД, ЦИКЛ ЗАПИСИ							
Длительность низкого уровня при записи 0	t_{W0L}	Стандартная скорость	64		120	мкс	1, 14
		Стандартная скорость, –20°C .. +85°C	60		120		
		Повышенная скорость, V_{PUP} > 4,5 В	6		15,4		
Длительность низкого уровня при записи 1	t_{W1L}	Стандартная скорость	5		15 - ε	мкс	1, 11
		Повышенная скорость	1		2 - ε		
Окно опроса при записи (для подчиненного устройства)	t_{SLS}	Стандартная скорость	19		64	мкс	14
		Стандартная скорость, –20°C .. +85°C	19		60		
		Повышенная скорость, V_{PUP} > 4,5 В	2		4,8		
ВХОД/ВЫХОД, ЦИКЛ ЧТЕНИЯ							
Время установления при чтении 0	t_{SU}	Стандартная скорость			5	мкс	
		Повышенная скорость			1		
Длительность низкого уровня при чтении	t_{RL}	Стандартная скорость	5		15 - δ	мкс	1, 12
		Повышенная скорость	1		2 - δ		

ПАРАМЕТР	СИМВ.	УСЛОВИЯ	МИН.	ТИП.	МАКС.	ЕД.	ПРИМ.
Длительность низкого уровня при чтении 0 (для подчиненного устройства)	t_{SPD}	Стандартная скорость	19		64	мкс	14
		Стандартная скорость, –20°C .. +85°C	19		60		
		Повышенная скорость, V_{PUP} > 4,5 В	2		4,8		
Окно опроса при чтении	t_{MSR}	Стандартная скорость	$t_{RL} + \delta$		15	мкс	1, 12
		Повышенная скорость	$t_{RL} + \delta$		2		
БЛОК SHA-1							
Длительность вычислений	t_{SHA}			0,4	1,15	мс	
Количество вычислений SHA-1	N_{SHA}		(см. графики)			-	13

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 2,2 Ком с линии данных на V_{PUP}, то достаточно времени 2,5 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, при переходе из единицы в ноль 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, при переходе из ноля в единицу 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) Количество вычислений SHA-1, которое может быть проделано с внутренним источником питания, зависит от рабочей температуры и температуры хранения устройства.
14) Выделенные цветом значения **не** соответствуют опубликованному стандарту на iButton. Смотрите сравнительную таблицу, которая приведена ниже.
15) Время восстановления было специально увеличено по сравнению со стандартным значением 1 мкс для улучшения паразитного питания устройства. Это изменение улучшает характеристики микросхемы и не рассматривается как несоответствие опубликованному стандарту.

Таблица несовместимости для T_A = -40°C .. +85°C

Название параметра	Стандартные значения				Значения для DS1963S			
	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	-	69 мкс	-	8 мкс	-
t_{RSTL}	480 мкс	-	48 мкс	80 мкс	540 мкс	960 мкс	48 мкс	80 мкс
t_{PDH}	15 мкс	60 мкс	2 мкс	6 мкс	17 мкс	60 мкс	1,8 мкс	6 мкс
t_{PDL}	60 мкс	240 мкс	8 мкс	24 мкс	78 мкс	260 мкс	7,7 мкс	24 мкс
t_{W0L}	60 мкс	120 мкс	6 мкс	16 мкс	64 мкс	120 мкс	6 мкс	15,4 мкс
t_{SLS}, t_{SPD}	15 мкс	60 мкс	2 мкс	6 мкс	19 мкс	64 мкс	2 мкс	4,8 мкс

Таблица несовместимости для T_A = -20°C .. +85°C

Название параметра	Стандартные значения				Значения для DS1963S			
	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	-	8 мкс	-
t_{RSTL}	480 мкс	-	48 мкс	80 мкс	480 мкс	960 мкс	48 мкс	80 мкс
t_{PDH}	15 мкс	60 мкс	2 мкс	6 мкс	17 мкс	60 мкс	1,8 мкс	6 мкс
t_{PDL}	60 мкс	240 мкс	8 мкс	24 мкс	78 мкс	240 мкс	7,7 мкс	24 мкс
t_{W0L}	60 мкс	120 мкс	6 мкс	16 мкс	60 мкс	120 мкс	6 мкс	15,4 мкс
t_{SLS}, t_{SPD}	15 мкс	60 мкс	2 мкс	6 мкс	19 мкс	60 мкс	2 мкс	4,8 мкс

Зависимость ожидаемого срока службы от температуры