Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

DS1961S
1kb Protected EEPROM iButton With SHA-1 Engine

www.maxim-ic.com

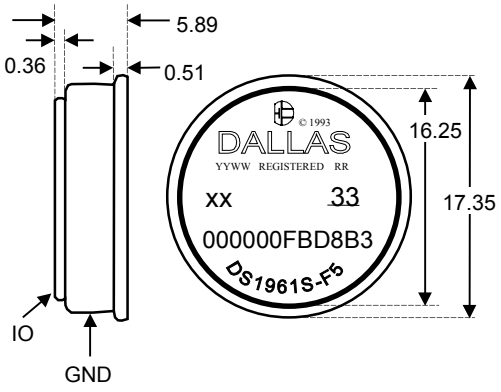
SPECIAL FEATURES

- 1128 Bits of 5V EEPROM Memory Partitioned into Four Pages of 256 Bits, a 64-Bit Write-Only Secret, and up to Five General-Purpose Read/Write Registers
- Write Access Requires Knowledge of the Secret and the Capability of Computing and Transmitting a 160-Bit MAC (Message Authentication Code) as Authorization
- Secret and Data Memory can be Write-Protected (All or Page 0 Only) or put in EPROM-Emulation Mode ("Write to 0", Page 1)
- On-Chip, 512-Bit SHA-1 Engine to Compute 160-Bit MACs and Generate Secrets
- Reads and Writes Over a Wide 2.8V to 5.25V Voltage Range from -40°C to +85°C
- Communicates to Host with a Single Digital Signal at 14.1kbps using 1-Wire® Protocol
- On-Chip, 16-Bit Cyclic Redundancy Check (CRC) Generator for Safeguarding Data Transfers
- Overdrive Mode Boosts Communication Speed to 125kbps
- Operating Temperature Range from -40°C to +85°C
- Minimum 10 Years of Data Retention at +85°C

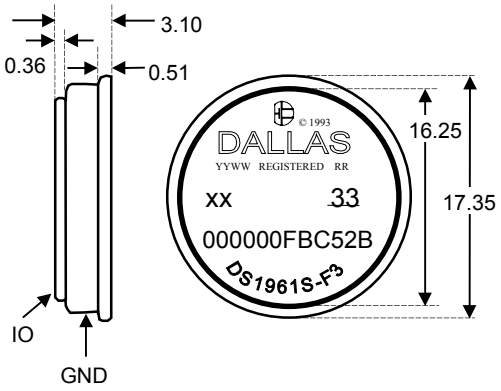
COMMON iButton FEATURES

- Unique, Factory-Lasered and Tested 64-Bit Registration Number (8-Bit Family Code + 48-Bit Serial Number + 8-Bit CRC Tester) Assures Absolute Traceability Because No Two Parts are Alike
- Multidrop Controller for 1-Wire Net
- Digital Identification and Information by Momentary Contact
- Chip-Based Data Carrier Compactly Stores Information
- Data can be Accessed While Affixed to Object
- Button Shape is Self-Aligning with Cup-Shaped Probes
- Durable Stainless-Steel Case Engraved with Registration Number Withstands Harsh Environments
- Easily Affixed with Self-Stick Adhesive Backing, Latched by its Flange, or Locked with a Ring Pressed onto its Rim
- Presence Detector Acknowledges when Reader First Applies Voltage
- Meets UL#913 (4th Edit.). Intrinsically Safe Apparatus: Approved Under Entity Concept for use in Class I, Division 1, Groups A, B, C, and D Locations (Application Pending)

F5 MicroCan



F3 MicroCan



All dimensions are shown in millimeters.

ORDERING INFORMATION

DS1961S-F5	F5 iButton
DS1961S-F3	F3 iButton

EXAMPLES OF ACCESSORIES

DS1963S	SHA Coprocessor and Button
DS9096P	Self-Stick Adhesive Pad
DS9101	Multipurpose Clip
DS9093RA	Mounting Lock Ring
DS9093A	Snap-In Fob
DS9092	iButton Probe

iButton DESCRIPTION

The DS1961S combines 1024 bits of EEPROM, a 64-bit secret, an 8-byte register/control page with up to five user-read/write bytes, a 512-bit SHA-1 engine, and a fully featured 1-Wire interface in a rugged iButton. Data is transferred serially through the 1-Wire protocol, which requires only a single data lead and a ground return. The DS1961S has an additional memory area called the scratchpad that acts as a buffer when writing to the main memory, the register page, or when installing a new secret. Data is first written to the scratchpad from where it can be read back. After the data has been verified, a copy scratchpad command transfers the data to its final memory location, provided that the DS1961S receives a matching 160-bit MAC. The computation of the MAC involves the secret and additional data stored in the DS1961S including the device's identity register. Only a new secret can be loaded without providing a MAC. The SHA-1 engine can also be activated to compute 160-bit MACs when reading a memory page or to compute a new secret, instead of loading it.

The DS1961S understands a unique command "Refresh Scratchpad." Proper use of a refresh sequence after a copy scratchpad operation reduces the number of weak bit failures in a touch environment (see the *Writing with Verification* section). The refresh sequence also provides a means to restore functionality in a device with bits in a weak state.

Each DS1961S has its own 64-bit ROM registration number that is factory lasered into the chip to provide a guaranteed unique identity for absolute traceability. The durable stainless-steel package is highly resistant to environmental hazards such as dirt, moisture, and shock. Its compact coin-shaped profile is self-aligning with mating receptacles, allowing the DS1961S to be easily used by human operators. Accessories permit the DS1961S to be mounted on almost any surface including plastic key fobs and photo-ID badges.

APPLICATIONS

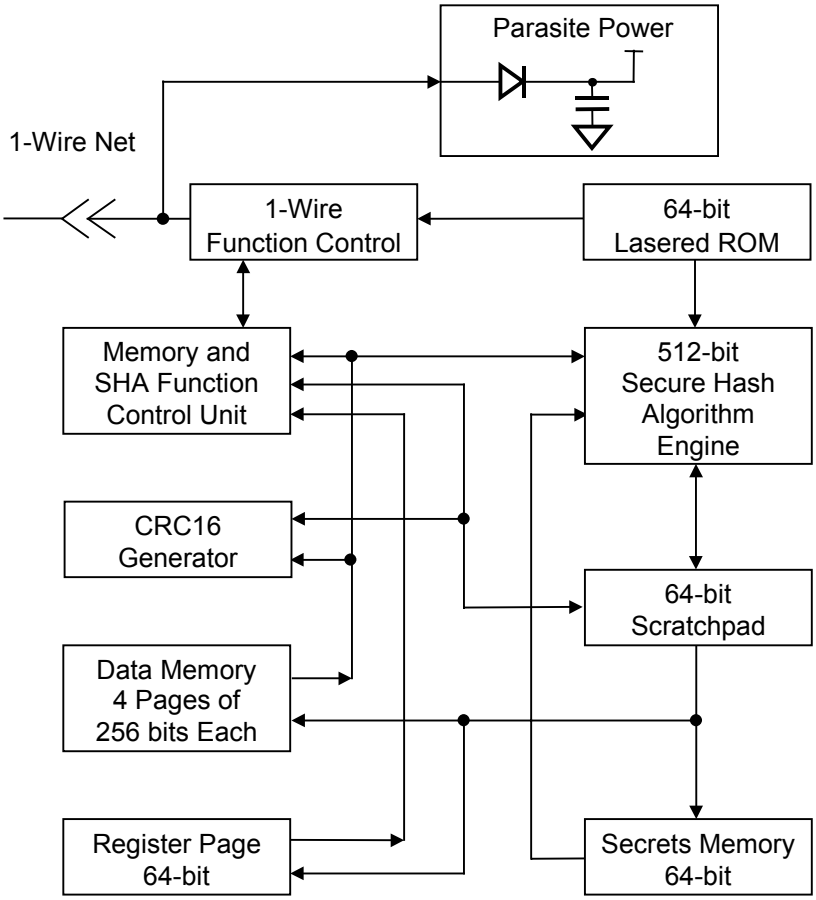
The DS1961S can be used for different purposes such as secure access control, user/product authentication, after-market management of consumables, and as monetary tokens in electronic payment systems. As carrier of electronic cash (eCash), the DS1961S can store up to three monetary files or "purses" of a single service provider, which make the device well suited for company-sized single-secret applications such as cafeteria, copy machines, and access control at entertainment parks or private clubs. For increased security or if the processing power of the host microcontroller is insufficient, a DS1963S can be used as secure coprocessor to verify MACs generated by the DS1961S or to compute MACs needed for writing to the DS1961S.

OVERVIEW

The block diagram in Figure 1 shows the relationships between the major control and memory sections of the DS1961S. The DS1961S has five main data components: 1) 64-bit lasered ROM, 2) 64-bit scratchpad, 3) four 32-byte pages of EEPROM, 4) 64-bit register page, 5) 64-bit secrets memory, and 6) a 512-bit

SHA-1 (Secure Hash Algorithm) engine. The hierarchical structure of the 1-Wire protocol is shown in Figure 2. The bus master must first provide one of the seven ROM function commands, 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, 5) Resume Communication, 6) Overdrive-Skip ROM, or 7) Overdrive-Match ROM. Upon completion of an overdrive ROM command byte executed at standard speed, the device enters overdrive mode where all subsequent communication occurs at a higher speed. The protocol required for these ROM function commands is described in Figure 9. After a ROM function command is successfully executed, the memory functions become accessible and the master can provide any one of the eight memory and SHA function commands. The protocol for these memory and SHA function commands is described in Figure 7. All data is read and written LSB first.

Figure 1. DS1961S BLOCK DIAGRAM



64-BIT LASERED ROM

Each DS1961S contains a unique ROM code that is 64 bits long. The first eight bits are a 1-Wire family code. The next 48 bits are a unique serial number. The last eight bits are a CRC of the first 56 bits (see Figure 3). The 1-Wire CRC is generated using a polynomial generator consisting of a shift register and XOR gates as shown in Figure 4. The polynomial is $X^8 + X^5 + X^4 + 1$. Additional information about the Dallas 1-Wire CRC is available in *The Book of DS19xx iButton Standards* from Dallas Semiconductor. The shift register bits are initialized to zero. Then starting with the LSB of the family code, one bit at a time is shifted in. After the 8th bit of the family code has been entered, then the serial number is entered. After the 48th bit of the serial number has been entered, the shift register contains the CRC value. Shifting in the eight bits of CRC should return the shift register to all zeros.

Figure 2. HIERARCHICAL STRUCTURE FOR 1-WIRE PROTOCOL

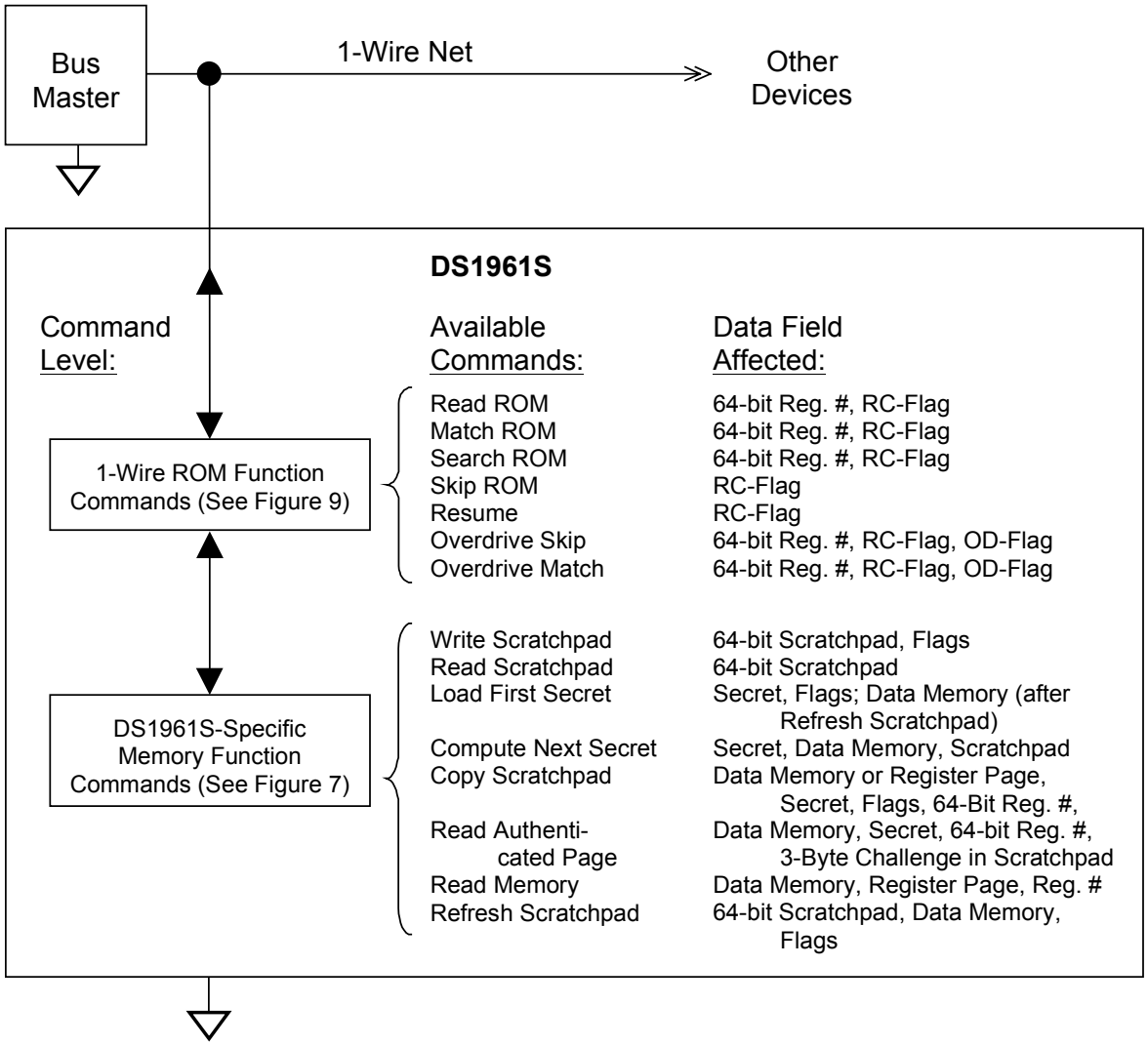


Figure 3. 64-BIT LASERED ROM

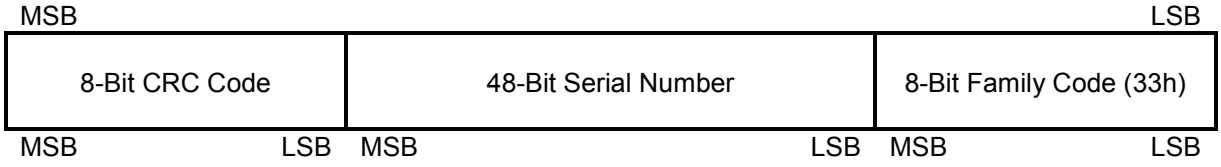
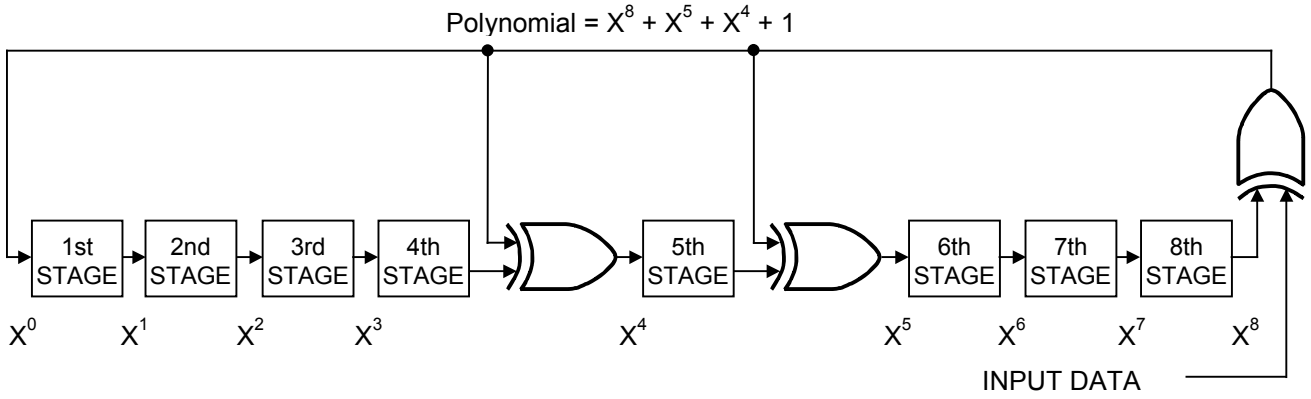


Figure 4. 1-WIRE CRC GENERATOR



MEMORY MAP

The DS1961S has four memory areas: data memory, secrets memory, register page with special function registers and user-bytes, and a scratchpad. The data memory is organized in pages of 32 bytes. Secret, register page, and scratchpad are 8 bytes each. The scratchpad acts as a buffer when writing to the data memory, loading the initial secret or when writing to the register page.

Data memory, secrets memory, and the register page are located in a linear address space, as shown in Figure 5. The data memory and the register page have unrestricted read access. Writing to the data memory and the register page requires knowledge of the secret.

Figure 5. DS1961S MEMORY MAP

ADDRESS RANGE	DESCRIPTION	NOTE
0000h to 001Fh	Data Memory Page 0	No Write-Access Without Secret
0020h to 003Fh	Data Memory Page 1	No Write-Access Without Secret
0040h to 005Fh	Data Memory Page 2	No Write-Access Without Secret
0060h to 007Fh	Data Memory Page 3	No Write-Access Without Secret
0080h to 0087h	Secrets Memory	No Read Access; No Secret Needed for Write Access
0088h [1)]	Write-Protect Secret, 008Ch to 008Fh	Protection Activated by Code AAh or 55h
0089h [1)]	Write-Protect Pages 0 to 3	Protection Activated by Code AAh or 55h
008Ah [1)]	User Byte, Self-Protecting	Protection Activated by Code AAh or 55h
008Bh	Factory Byte (Read Only)	Reads Either AAh or 55h; See Text
008Ch [1)]	User Byte/EPROM Mode Control for Page 1	Mode Activated by Code AAh or 55h
008Dh [1)]	User Byte/Write-Protect Page 0 Only	Protection Activated by Code AAh or 55h
008Eh to 008Fh	User Bytes/Manufacturer ID	Function Depends on Factory Byte
0090h to 0097h	64-Bit Identity Register	Read-Only Access

[1)] Once programmed to AAh or 55h this address becomes read-only. All other codes can be stored but will neither write-protect the address nor activate any function.

The secret can be installed either by copying data from the scratchpad to the secrets memory or by computation using the current secret and the scratchpad contents as partial secret. The secret cannot be read directly; only the SHA engine has access to it for computing message authentication codes.

The address range 0088h to 008Fh, also referred to as the Register Page, contains special function registers as well as general-purpose user-bytes and one factory byte. Once programmed to AAh or 55h, most of these bytes become write-protected and can no longer be altered. All other codes neither write-protect the address nor activate the special function associated to that particular byte. Special functions are: 1) write-protecting only the secret, 2) write-protecting all four data memory pages simultaneously, 3) activating EPROM mode for data memory page 1 only, and 4) write-protecting data memory page 0 only. Once EPROM mode is activated, bits in the address range 0020h through 003Fh can only be altered from a logic 1 to a logic 0, provided that the data memory is not write protected.

The factory byte either reads 55H or AAh. Typically, this address reads 55h, indicating that the addresses 008E and 008F are read/write user-bytes without any special function or locking mechanism. The code of AAh indicates that these two bytes are programmed with a 16-bit manufacturer ID and then write-protected at the factory. The manufacturer ID can be a customer-supplied identification code that assists the application software in identifying the product the DS1961S is associated with and in faster selection of the applicable secret. To setup and register a manufacturer ID contact the factory.

The address range 0090h to 0097h is called the identity register. Typically, the identity register contains a copy of the device's ROM registration number. The family code is stored at the lower address followed by the 48-bit serial number and the 8-bit CRC, which is stored at address 0097h. In reading through these addresses (0090h to 0097h) the bus master receives the individual bits of the registration number in exactly the same sequence as with a ROM function command. With customized versions, the content of the identity register can be any customer-specified constant pattern. For more information on customization contact the factory.

Figure 6. ADDRESS REGISTERS

Bit Number	7	6	5	4	3	2	1	0
Target Address (TA1)	T7	T6	T5	T4	T3	T2 (0)	T1 (0)	T0 (0)
Target Address (TA2)	T15	T14	T13	T12	T11	T10	T9	T8
Ending Address with Data Status (E/S) (Read Only)	AA	1	PF	1	1	E2 (1)	E1 (1)	E0 (1)

ADDRESS REGISTERS AND TRANSFER STATUS

The DS1961S employs three address registers: TA1, TA2, and E/S (Figure 6). These registers are common to many other 1-Wire devices but operate slightly differently with the DS1961S. Registers TA1 and TA2 must be loaded with the target address to which the data is written or from which data is read. Register E/S is a read-only transfer-status register, used to verify data integrity with write commands. Since the scratchpad of the DS1961S is designed to accept data in blocks of eight bytes only, the lower three bits of TA1 are forced to 0 and the lower three bits of the E/S register (ending offset) always read 1.

This indicates that all the data in the scratchpad is used for a subsequent copying into main memory or secret. Bit 5 of the E/S register, called PF or partial byte flag, is a logic-1 if the number of data bits sent by the master is not an integer multiple of eight or if the data in the scratchpad is not valid due to a loss of power. A valid write to the scratchpad clears the PF bit. Bits 3, 4, and 6 have no function; they always read 1. The partial flag supports the master checking the data integrity after a write command. The highest valued bit of the E/S register is called the AA or authorization accepted flag, which indicates that the data stored in the scratchpad has already been copied to the target memory address. Writing data to the scratchpad clears this flag.

WRITING WITH VERIFICATION

To write data to the DS1961S, the scratchpad has to be used as intermediate storage. First the master issues the write scratchpad command, which specifies the desired target address and the data to be written to the scratchpad. Note that writes to data memory must be performed on 8-byte boundaries with the three LSBs of the target address T2–T0 equal to 000b. Therefore, if T2–T0 are sent with non-zero values, the device sets these bits to zero and uses the modified address as the target address. The master should always send eight complete data bytes. After the eight bytes of data have been transmitted, the master can elect to receive an inverted CRC16 of the write scratchpad command, the address as sent by the master, and the data as sent by the master. The master can compare the CRC to the value it has calculated itself in order to determine if the communication was successful. After the scratchpad has been written, the master should always perform a read scratchpad to verify that the intended data was in fact written. During a read scratchpad, the DS1961S repeats the target address TA1 and TA2 and sends the contents of the E/S register. The partial flag (bit 5 of the E/S register) is set to 1 if the last data byte the DS1961S received during a write scratchpad or refresh scratchpad command was incomplete, or if there was a loss of power since data was last written to the scratchpad. The authorization-accepted (AA) flag (bit 7 of the E/S register) is normally cleared by a write scratchpad or refresh scratchpad; therefore, if it is set to 1, it indicates that the DS1961S did not understand the proceeding write (or refresh) scratchpad command. In either of these cases, the master should rewrite the scratchpad. After the master receives the E/S register, the scratchpad data is received. The descriptions of write scratchpad and refresh scratchpad provide clarification of what changes can occur to the scratchpad data under certain conditions. An inverted CRC of the read scratchpad command, target address, E/S register, and scratchpad data follows the scratchpad data. As with the write scratchpad command, this CRC can be compared to the value the master has calculated itself in order to determine if the communication was successful. After the master has verified the data, it can send the copy scratchpad to copy the scratchpad to memory. Alternatively, the load first secret or compute next secret command can be issued to change the secret. See the descriptions of these commands for more information.

In a touch environment the quality of the electrical contact cannot be guaranteed. With poor or intermittent contact it is possible for a copy scratchpad command to complete with insufficient energy, leaving the floating gate voltage of an EEPROM bit in the area of the threshold between 0 and 1. When this occurs, the logical value of the bit is not assured. Depending on voltage and/or temperature conditions, the same bit can be read by the host as one polarity and then by the internal SHA-1 engine as the opposite polarity. This becomes a fatal lockup mode because the host cannot formulate a proper SHA-1 MAC to enable the bit to be rewritten. To repair poorly written bits and thereby restore the device to functionality, the refresh scratchpad command was introduced. Combined with the load first secret command, refresh scratchpad provides a means to restore the EEPROM bits to normal values, removing lockup conditions and allowing the device to be written again.

To prevent the occurrence of poorly written bits, a refresh sequence should be performed after each copy scratchpad command. A refresh sequence is defined as a refresh scratchpad (to the same target address as the previous copy scratchpad), followed by a load first secret. The EN_LFS flag is set by the refresh

scratchpad command. The EN_LFS flag enables the use of load first secret to addresses 0000h–007Fh. Using load first secret allows the master to copy the scratchpad to memory without the MAC computation necessary during a copy scratchpad. If the master attempts any command after refresh scratchpad that could change the scratchpad data or the target address, EN_LFS is reset to 0. This prevents the use of load first secret to load any data other than the refreshed memory data to any location other than the one specified during refresh scratchpad. Refresh scratchpad behaves exactly as write scratchpad does for target addresses 0080h and above. In this case the EN_LFS flag is not set, so it is not possible to refresh the data in the secret (0080h) or in the register page (0088h). This prevents the secret from being revealed by a refresh scratchpad followed by a read scratchpad.

MEMORY AND SHA FUNCTION COMMANDS

Due to its design as a secure device, the DS1961S has to behave differently from other memory iButtons. Although most of the memory of the DS1961S can be read the same way as any other memory iButton, attempts to read the secret results in FFh-bytes rather than real data. The *Memory and SHA Function Flow Chart* (Figure 7) describes the protocols necessary for accessing the memory and operating the SHA engine. The communication between master and DS1961S takes place either at regular speed (default, OD = 0) or at overdrive speed (OD = 1). If not explicitly set into overdrive mode the DS1961S assumes regular speed.

Write Scratchpad [0Fh]

The write scratchpad command applies to the data memory, the secret and the writeable addresses in the register page. If the bus master sends a target address higher than 90h, the command is not executed.

After issuing the write scratchpad command, the master must first provide the 2-byte target address, followed by the data to be written to the scratchpad. The data is written to the scratchpad starting at the beginning of the scratchpad. Note that the ending offset (E2..E0, see Figure 6) is always 111b regardless of the number of bytes that the master has transmitted. For this reason the master should always send eight bytes, especially if the data is to be loaded as a secret. If the master sends less than eight data bytes and does not read back the scratchpad for verification, parts of the new secret can be random data that is unknown to the master. Only full data bytes are accepted. If the last data byte is incomplete its content is ignored and the partial byte flag (PF) is set.

When executing the write scratchpad command the CRC generator inside the DS1961S (see Figure 12) calculates a CRC of the entire data stream, starting at the command code and ending at the last data byte as sent by the master. This CRC is generated using the CRC16 polynomial by first clearing the CRC generator and then shifting in the command code (0Fh) of the write scratchpad command, the target addresses (TA1 and TA2), and all the data bytes. Note that the CRC16 calculation is performed with the actual TA1 sent by the master even though the DS1961S sets TA1 bits T2..T0 to 000b for the actual write scratchpad command. The master can end the write scratchpad command at any time. However, if the scratchpad is filled to its capacity, the master can send 16 read-time slots and receives the CRC generated by the DS1961S. If the master continues reading after the CRC all data is be FFh.

After receiving the target addresses (TA1 and TA2), the DS1961S clears the EN_LFS flag. If EPROM mode is active and a write scratchpad is attempted within page 1 (0020h–003Fh), the scratchpad is loaded with the logical AND of the scratchpad data sent by the master and the current content of the target memory location. If a write scratchpad is attempted to the register page (0088h–008Fh), any bytes that are write-protected overwrite the corresponding scratchpad data byte sent by the master with the existing value. In all other cases, the data sent by the master is written to the scratchpad unaltered.

Read Scratchpad [AAh]

The read scratchpad command allows verifying the target address and the integrity of the scratchpad data. After issuing the command code, the master begins reading. The first two bytes is the target address with T2 to T0 = 0. The next byte is the ending offset/data status byte (E/S) followed by the scratchpad data, which may be different from what the master has originally sent. This is of particular importance if the target address is the secret, the register page, page 1 (in EPROM mode), or if refresh was used to load the scratchpad. In these cases, the scratchpad can contain data other than that which was sent during either the write scratchpad or refresh scratchpad commands. The master should read through the end of the scratchpad after which it receives the inverted CRC that is computed with the data as sent by the DS1961S. If the master continues reading after the CRC all data is FFh.

The scratchpad can be loaded using the write scratchpad or refresh scratchpad command. The data found in the scratchpad depends on the command used, the target address, and whether or not EPROM mode is active. See the descriptions of write scratchpad and refresh scratchpad for clarification.

Load First Secret [5Ah]

The load first secret command has two modes of operation, which are controlled by the EN_LFS flag. With EN_LFS = 0, the command replaces the device's current secret with the contents of the scratchpad, provided that the secret is not write-protected. With EN_LFS = 1, the command allows to rewrite memory data (addresses 0000h to 007Fh), bypassing the SHA-1 computation that is required when doing the same through the copy scratchpad command. The EN_LFS flag is 0 unless it has been set to 1 by executing the refresh scratchpad command prior to load first secret.

Case EN_LFS = 0

Before the load first secret command can be used in this mode, the master must have written the new secret to the scratchpad using the starting address of the secret (0080h). After issuing the load first secret command, the master must provide a 3-byte authorization pattern (TA1, TA2, E/S, in that order), that should have been obtained by an immediately preceding read scratchpad command. This 3-byte pattern must exactly match the data contained in the three address registers (see Figure 6). If the pattern matches and the secret is not write-protected, the AA flag is set and the copy begins. All eight bytes of scratchpad contents are copied to the secret's memory location.

Case EN_LFS = 1

To use the load first secret command in this mode, the refresh scratchpad command must have been executed to load eight bytes of memory data (address range 0000h to 007Fh) into the scratchpad, which sets the EN_LFS flag to 1. After issuing the load first secret command, the master must provide a 3-byte authorization pattern (TA1, TA2, E/S, in that order), that can be obtained by an immediately preceding read scratchpad command without affecting the EN_LFS flag. This 3-byte pattern must exactly match the data contained in the three address registers (see Figure 6). If the pattern matches and the memory is not write-protected, the AA flag is set and the copy begins. All eight bytes of scratchpad contents are copied to the memory location.

Regardless of the mode used, the duration of the copy operation is t_{PROG} during which the voltage on the 1-Wire bus must not fall below 2.8V. The master should read at least one byte at the conclusion of the copy delay. Reading AAh indicates that the copy was successful, while reading FFh indicates that the copy was not successful. Instead of using load first secret with EN_LFS = 0, a new secret can alternatively be loaded with the copy scratchpad command. However, this approach requires the knowledge of the current secret and the computation of a 160-bit MAC.

Figure 7-1. MEMORY AND SHA FUNCTIONS FLOW CHART

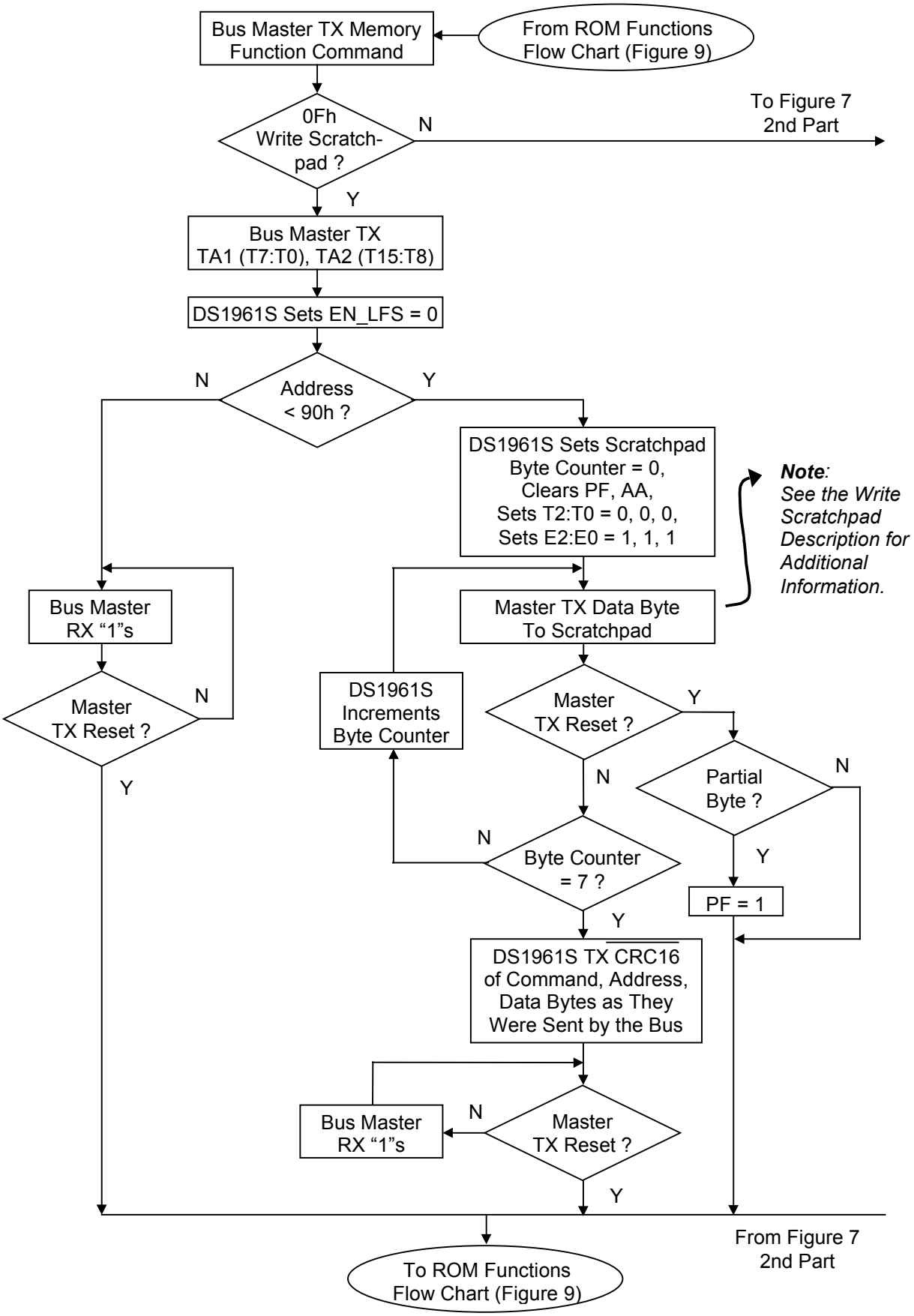


Figure 7-2. MEMORY AND SHA FUNCTIONS FLOW CHART (continued)

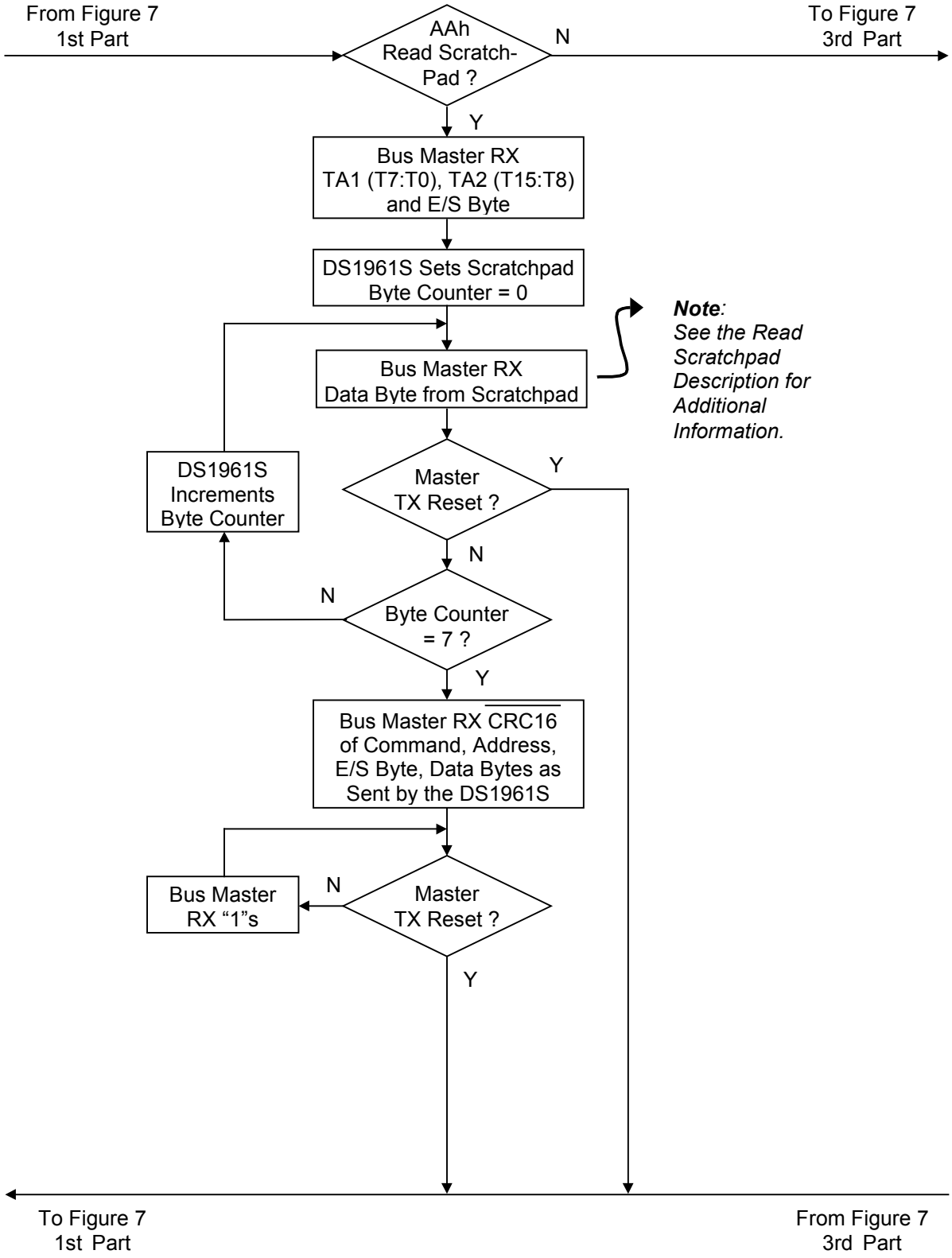


Figure 7-3. MEMORY AND SHA FUNCTIONS FLOW CHART (continued)

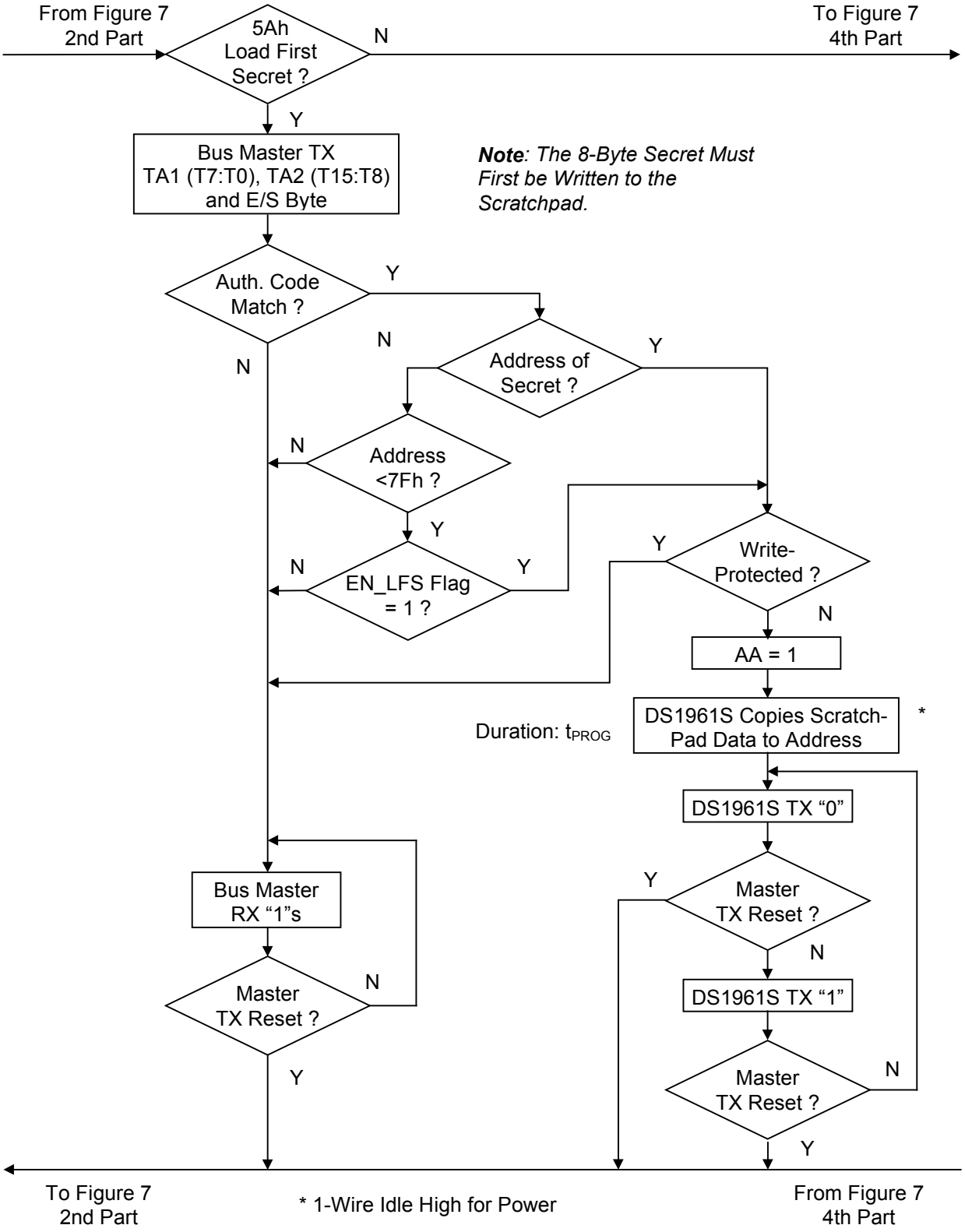


Figure 7-4. MEMORY AND SHA FUNCTIONS FLOW CHART (continued)

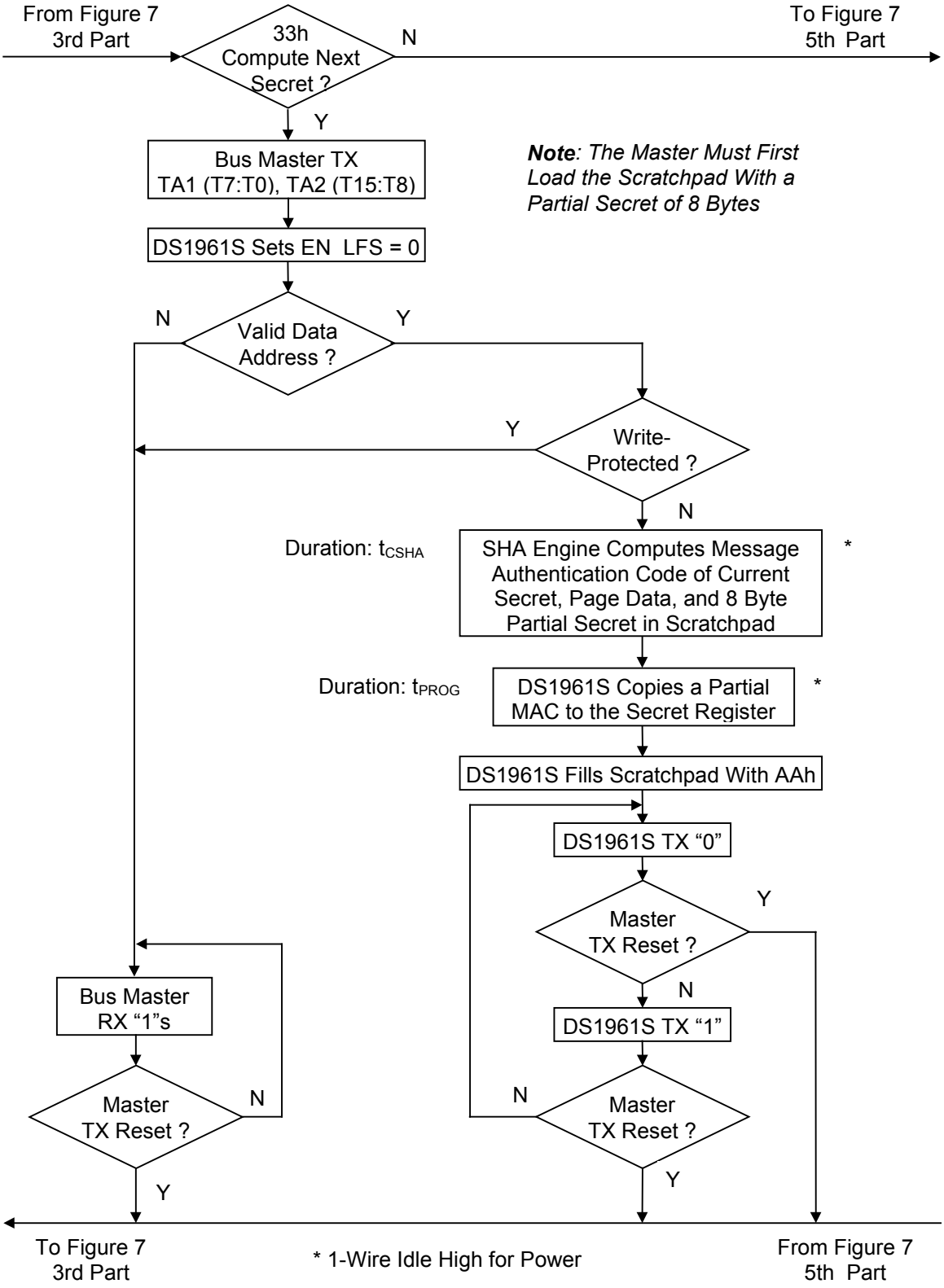


Figure 7-5. MEMORY AND SHA FUNCTIONS FLOW CHART (continued)

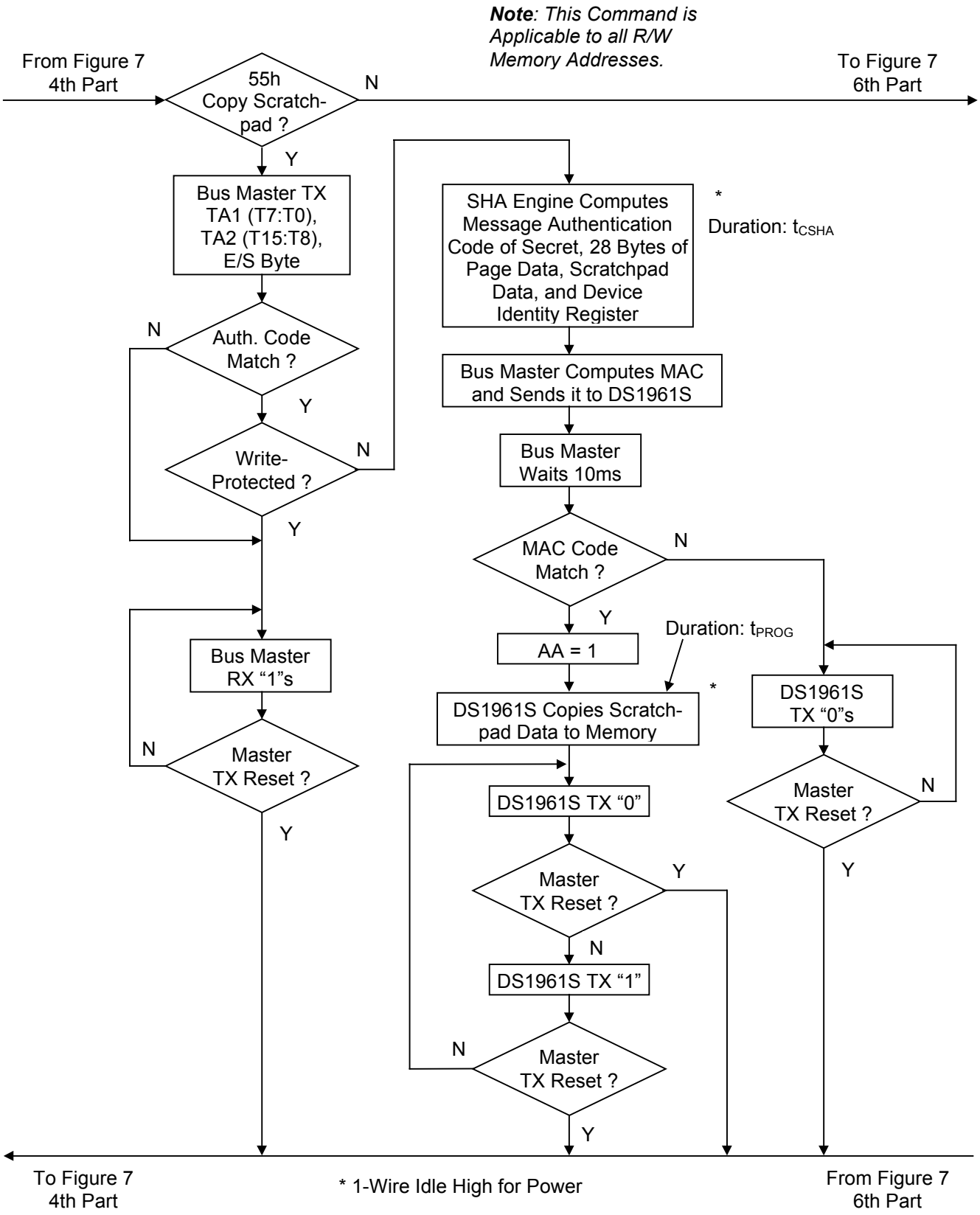


Figure 7-6. MEMORY AND SHA FUNCTIONS FLOW CHART (continued)

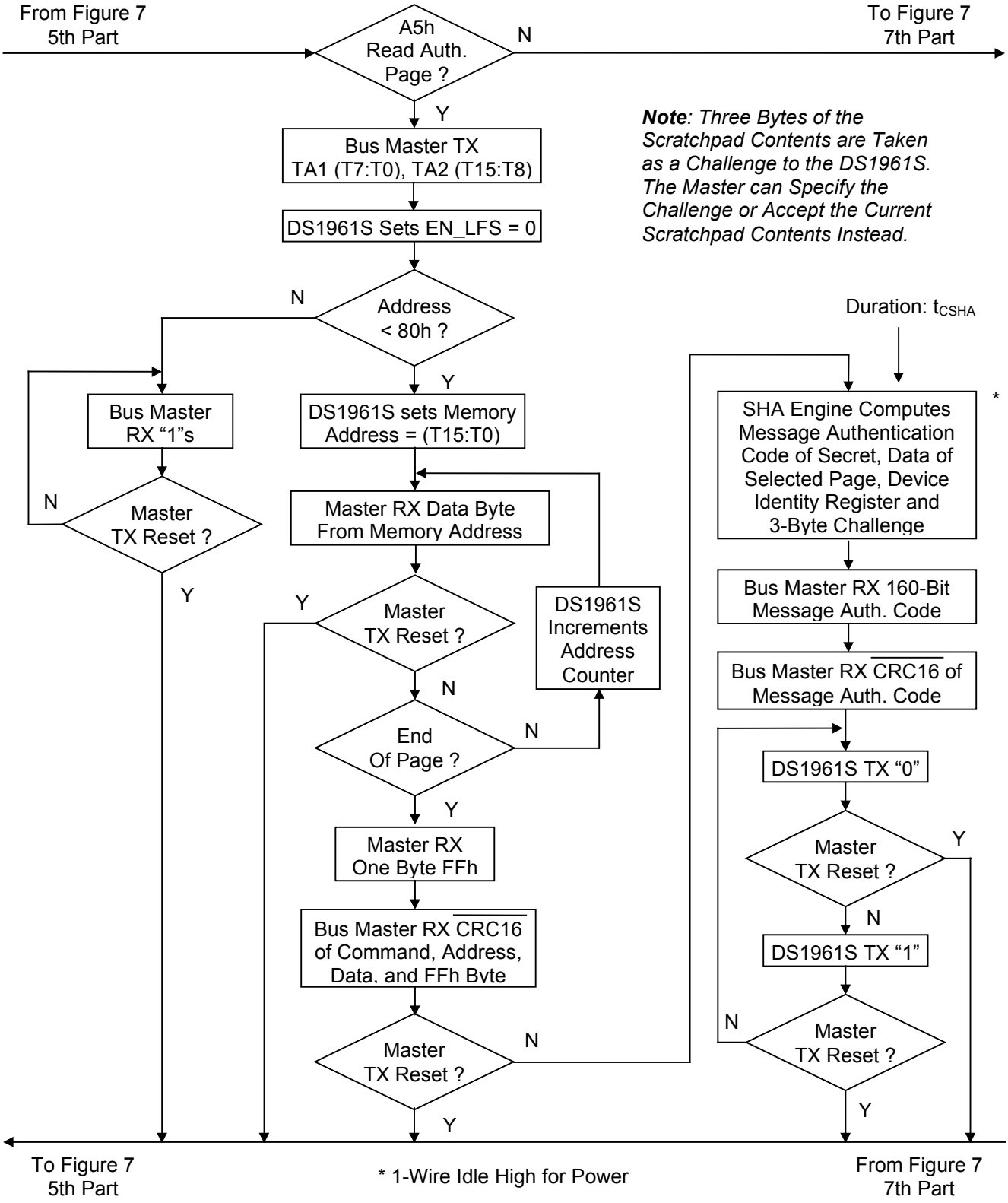


Figure 7-7. MEMORY AND SHA FUNCTIONS FLOW CHART (continued)

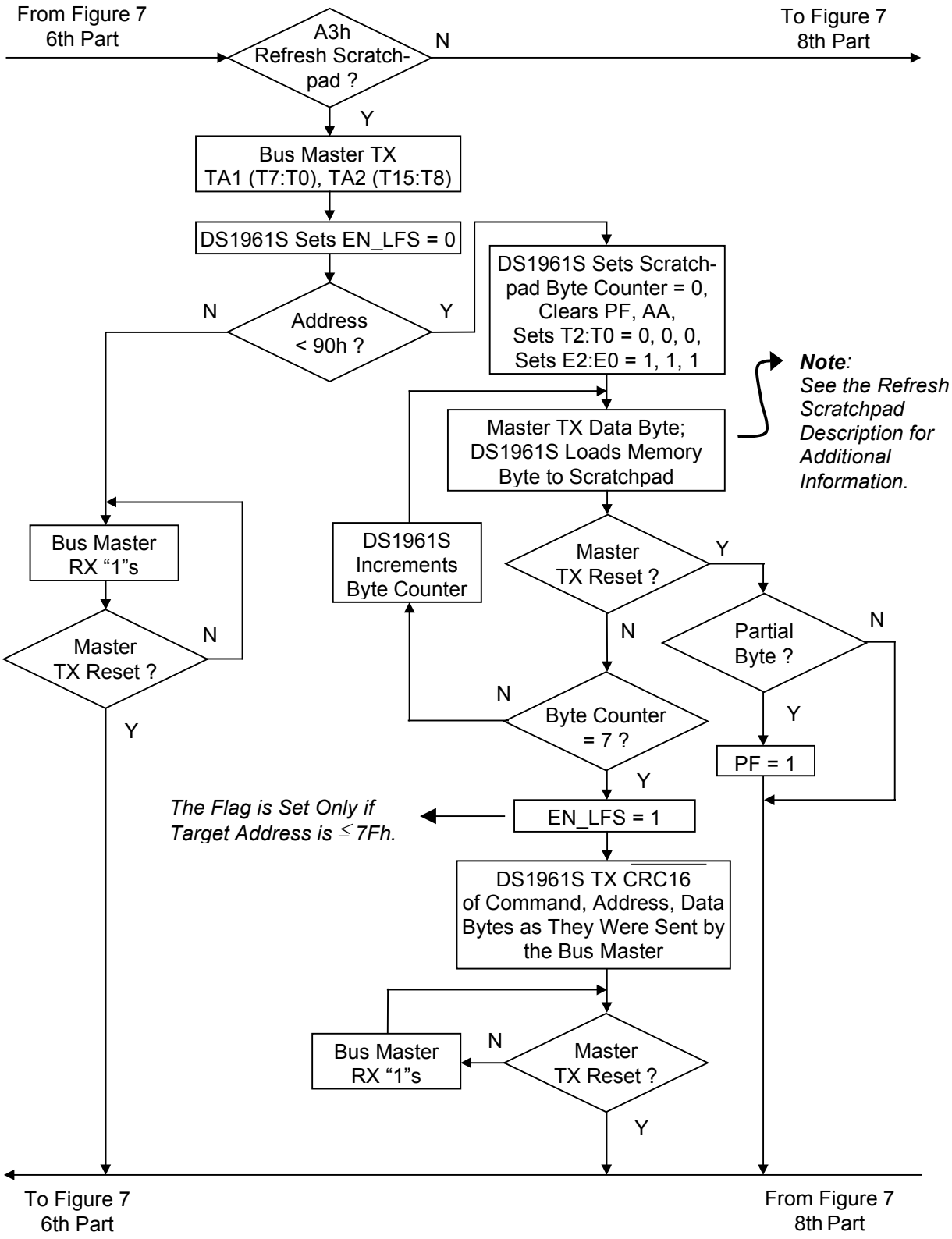
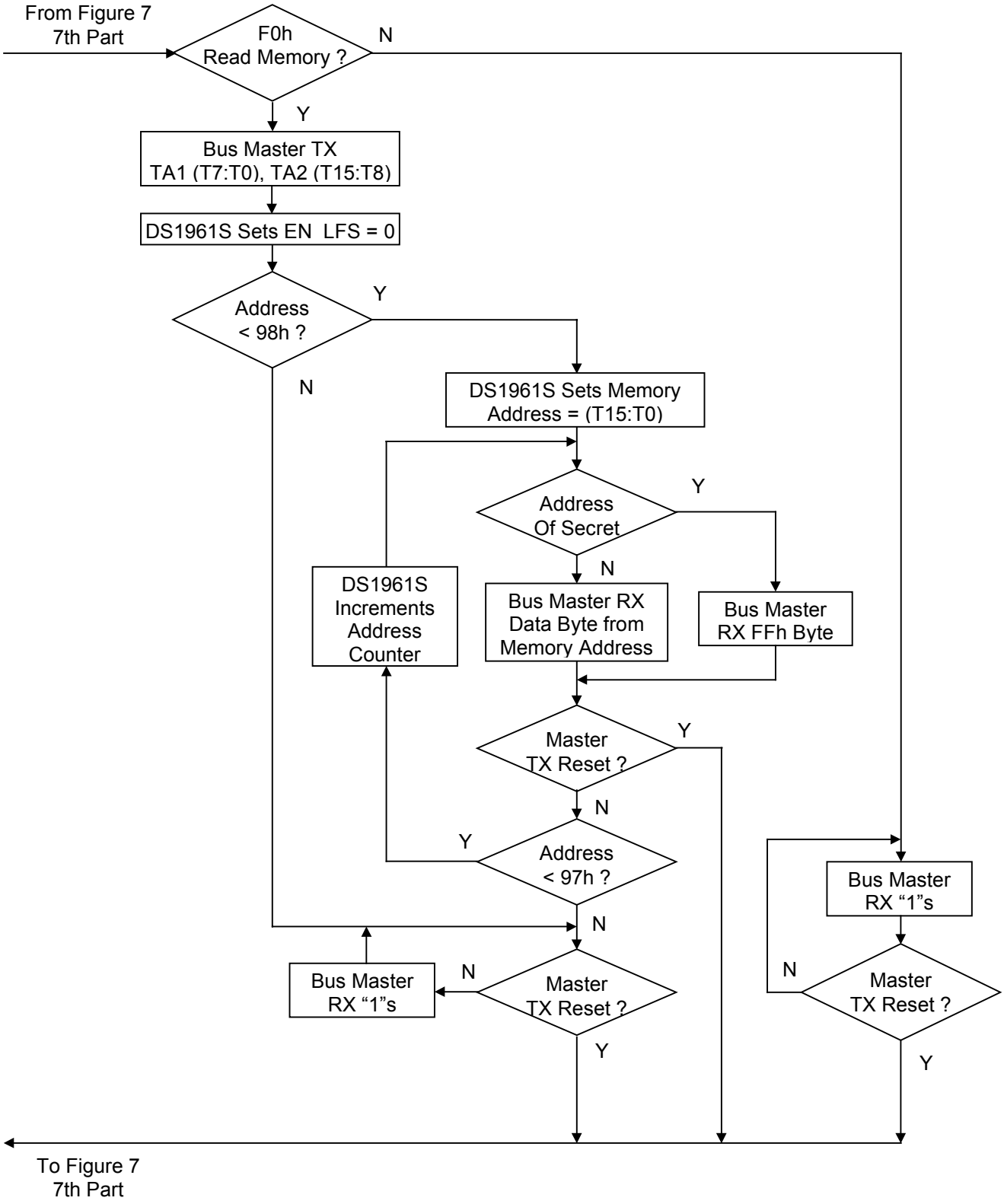


Figure 7-8. MEMORY AND SHA FUNCTIONS FLOW CHART (continued)



Compute Next Secret [33h]

Some applications may require a higher level of security than can be achieved by a single, directly written secret. For additional security the DS1961S can compute a new secret based on the current secret, the contents of a selected memory page, and a partial secret that consists of all data in the scratchpad. To install a computed secret the master issues the compute next secret command, which activates the 512-bit SHA-1 engine, provided that the secret is not write-protected. Table 1 shows how the various data components involved enter the SHA engine and how a portion of the SHA result is loaded into the secret's memory location. The SHA computation algorithm itself is explained later in this document. The compute next secret command can be applied as often as desired to increase the level of security. The bus master does not need to know the device's current secret in order to successfully compute a new one and then overwrite the existing secret.

Table 1. SHA-1 INPUT DATA FOR COMPUTE NEXT SECRET COMMAND

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

RESULT OF COMPUTE NEXT SECRET

(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]

Legend

Mt	**Input Buffer of SHA Engine** 0 ≤ t ≤ 15; 32-Bit Words
(SS + N)	**Byte N of Secret; Secret Begins at Address 0080h** (See *Memory Map*)
(PP + N)	**Byte N of Memory Page; Memory Pages Begin at 0000h, 0020h, 0040h and 0060h** (See *Memory Map*)
(SP + N)	**Byte N of Scratchpad**
MPX	MPX[7] = 0; MPX[6] = 0; MPX[5:0] = (SP + 0)[5:0]
D, E	32-Bit Words, Portions of the 160-Bit SHA Result

After issuing the compute next secret command the master must provide a 2-byte target address to select the memory page that contributes 256 bits of the SHA input data. After receiving the target addresses

(TA1 and TA2), the DS1961S clears the EN_LFS flag. The lower five bits of the target address TA1 are ignored because only the page number is relevant. If the target address as sent by the master is valid (i.e., in the range of 0000h and 007Fh), and the secret is not write-protected, the SHA engine starts. The master must wait for t_{CSHA} during which the new secret is computed. Immediately following the SHA delay, the master must wait for t_{PROG} during which the new secret is copied to the secret register. During the t_{CSHA} and t_{PROG} the voltage on the 1-Wire bus must not fall below 2.8V. The DS1961S fills the scratchpad with AAh if the copy was successful, but does not modify the scratchpad if the SHA engine did not start because of an incorrect address or because of write protection. The master should read at least one byte at the conclusion of the copy delay. Reading AAh indicates that the copy was successful. Reading FFh indicates that the copy was not successful because of an incorrect address or because of write protection.

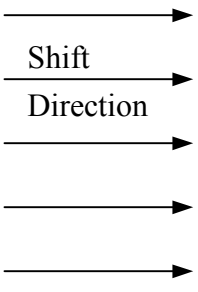
Since the content of the scratchpad is used as a partial secret, the master must fill the scratchpad with a known 8-byte data pattern using the write scratchpad command before it issues the compute next secret command. Otherwise the new secret depends on data that was unintentionally left in the scratchpad from previous commands.

Copy Scratchpad [55h]

The data memory of the DS1961S can be read without any restrictions. Executing the copy scratchpad command to write new data to the memory or register page, however, requires the knowledge of the device's secret and the ability to perform an SHA-1 computation to generate the 160-bit MAC to start the data transfer from the scratchpad to the memory. The master can perform the MAC computation in software or use a DS1963S as a coprocessor. The coprocessor approach has the benefit that the secret remains hidden in the coprocessor iButton. The sequence in which the resulting MAC needs to be sent to the DS1961S is shown in Table 2. Tables 3A and 3B show how the various data components are entered into the SHA engine. The SHA computation algorithm is explained later in this document.

Table 2. MESSAGE AUTHENTICATION CODE TRANSMISSION SEQUENCE

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]



The transmission is least significant bit first starting with register E.

After issuing the copy scratchpad command, the master must provide a 3-byte authorization pattern, which should have been obtained by an immediately preceding read scratchpad command. This 3-byte pattern must exactly match the data contained in the three address registers (TA1, TA2, E/S, in that order). If the authorization code matches and the target memory is not write-protected, the DS1961S starts its SHA engine to compute a 160-bit MAC that is based on the current secret, all of the scratchpad data, the first 28 bytes of the addressed memory page, and the first seven bytes of the identity register (the byte at address 0097h is not used; see Table 3A). The duration of this computation is t_{CSHA}, during which the voltage on the 1-Wire line must not drop below 2.8V. Simultaneously the master computes a MAC from the same data and, after t_{CSHA} is expired, sends it to the DS1961S as evidence that it is authorized to write to the EEPROM. Now the master waits for t_{PROG} during which the voltage on the 1-Wire bus must

not fall below 2.8V. If the MAC generated by the DS1961S matches the MAC that the master computed, the DS1961S sets its AA flag, and copy the entire scratchpad contents to the data EEPROM. The master should read at least one byte at the conclusion of the copy delay. Reading AAh indicates that the copy was successful. Reading 00h indicates that the copy was not successful because the computed MAC did not match the MAC sent by the master. Reading FFh indicates that the copy was not successful because of write protection or because of an incorrect authorization pattern.

Table 3a. SHA-1 INPUT DATA FOR COPY SCRATCHPAD COMMAND WHEN COPYING TO A DATA MEMORY PAGE

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] = (S + 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

Legend

Mt	**Input Buffer of SHA Engine** 0 ≤ t ≤ 15; 32-Bit Words
(SS + N)	**Byte N of Secret; Secret Begins at Address 0080h** (see *Memory Map*)
(PP + N)	**Byte N of Memory Page; Memory Pages Begin at 0000h, 0020h, 0040h and 0060h** (see *Memory Map*)
(SP + N)	**Byte N of Scratchpad**
MP	MP[7:3] = 00000b, MP[2:0] = T7:T5
(ID + N)	**Byte N of Identity Register** The Last Byte of the Identity Register is Not Used.

Special attention is required when copying data to the register page. In order to prevent unintentional locking of a special function register or user byte it is recommended to first read the register page and then write it with all intended modifications to the scratchpad. When copying data to the register page (or the secret using copy scratchpad), the input data for M1 to M7 of the SHA engine is the current secret (M1, M2), the current content of the register page (M3, M4), the full content of the identity register (M5, M6), and 4 bytes FFh (M7), as shown in Table 3B. As a consequence, when using a DS1963S as coprocessor to compute the MAC to transfer data from the scratchpad to the register page, the secret must be used as page data. This precludes the use of partial (computed) secrets if writing to the register page is

required. For practical use of the DS1961S as a monetary token, partial secrets are more critical than being able to write-protect the secret or other areas of the device.

Table 3b. SHA-1 INPUT DATA FOR COPY SCRATCHPAD COMMAND WHEN COPYING TO THE REGISTER PAGE OR SECRET

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

Legend

Mt	**Input Buffer of SHA Engine** $0 \le t \le 15$; 32-Bit Words
(SS + N)	**Byte N of Secret; Secret Begins at Address 0080h** (see *Memory Map*)
(RP + N)	**Byte N of Register Page; Page Begins at 0088h** (see *Memory Map*)
(SP + N)	**Byte N of Scratchpad**
MP	MP[7:0] = 04h
(ID + N)	**Byte N of Identity Register**

Read Authenticated Page [A5h]

The read authenticated page command provides the master with the data of a full or partial memory page plus a MAC. The MAC allows the master to determine whether the secret stored in the DS1961S is valid within the application. The DS1961S computes the MAC from its secret, all the data of the selected memory page, the first seven bytes of the identity register and a 3-byte challenge, which the master should write to the scratchpad prior to issuing the read authenticated page command. To do this, the master can use the write scratchpad command with any target address within the data memory. The relevant portions of the challenge are the 5th, 6th, and 7th bytes. Alternatively, the master can accept the data that happens to reside in the scratchpad from a previous command as a challenge. The 160-bit MAC is transmitted in the same way as with the copy scratchpad command, Table 2, but the data flows from the DS1961S to the master. The data input to the SHA engine as it applies to the read authenticated page command is shown in Table 4.

After the master has issued the command code and specified the target addresses (TA1 and TA2), the DS1961S first clears the EN_LFS flag. If the target address is valid (< 0080h), the master receives the page data beginning at the target address through the end of the data page, one byte FFh and the inverted CRC of the command code, target address, transmitted page data and FFh byte. If the target address is invalid (≥ 0080h), the master receives FFh bytes rather than page data. Immediately after the CRC is received, the master waits for t_{CSHA} during which the voltage on the 1-Wire bus must not fall below 2.8V. During this time the SHA engine of the DS1961S computes the message authentication code over the secret, all 32 data bytes of the selected page, the device's registration number (without the CRC) and the 3-byte challenge. Now the master reads the 160-bit MAC, which is followed by an inverted CRC as a means to safeguard the data transfer. If the master continues reading after the CRC it receives AAh.

Table 4. SHA-1 INPUT DATA FOR READ AUTHENTICATED PAGE COMMAND

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

Legend

Mt	**Input Buffer of SHA Engine** 0 ≤ t ≤ 15; 32-Bit Words
(SS + N)	**Byte N of Secret; Secret Begins at Address 0080h** (See *Memory Map*)
(PP + N)	**Byte N of Memory Page; Memory Pages Begin at 0000h, 0020h, 0040h and 0060h** (see *Memory Map*)
(SP + N)	**Byte N of Scratchpad**
MP	MP[7:3] = 01000b, MP[2:0] = T7:T5
(ID + N)	**Byte N of Identity Register** The Last Byte of the Identity Register is Not Used.

Refresh Scratchpad [A3h]

Refresh scratchpad loads memory data to the scratchpad and sets the EN_LFS flag, which enables the use of the load first secret command to re-write the data that was just read from the memory, bypassing the MAC computation of copy scratchpad.

The command flow chart of refresh scratchpad is very similar to write scratchpad. If the target address is between 0000h–007Fh, there are two primary differences. 1) The data bytes that the master transmits following the target address are discarded; instead, the scratchpad is loaded with the unaltered memory data located at the target address, even if the memory page is in EPROM mode. 2) After the master has transmitted the eight dummy bytes, the EN_LFS flag is set to 1. The EN_LFS flag is cleared to 0 after receiving TA1 and TA2 during a write scratchpad, compute next secret, read authenticated page, refresh scratch, read memory, or by a power-on reset, because these commands can change the target address and/or the data in the scratchpad.

When applied to addresses 0080h–008Fh, the refresh scratchpad command behaves the same way as write scratchpad. This protects the secret from being exposed by a subsequent read scratchpad command.

Read Memory [F0h]

The read memory command can be used to read all memory except for the secret. Attempting to read the secret results in FFh bytes instead of the actual secret. After the master has issued the command code and specified the target addresses (TA1 and TA2), the DS1961S first clears the EN_LFS flag. If the target address is valid, the master reads data beginning from the target address and can continue until address 0097h. If the master continues reading, the result is logic 1s. It is important to realize that the target address registers point to the last byte read. The ending offset/data status byte and the scratchpad are unaffected.

The hardware of the DS1961S provides a means to accomplish error-free writing to the memory section. To safeguard reading data in the 1-Wire environment and to simultaneously speed up data transfers, it is recommended to packetize data into data packets of the size of one memory page each. Such a packet typically stores a master-calculated 16-bit CRC with each page of data to ensure rapid, error-free data transfers that eliminate having to read a page multiple times to determine if the received data is correct or not. (Refer to *Application Note 114* for the recommended file structure, which is also referred to as TMEX Format.)

SHA-1 COMPUTATION ALGORITHM

This description of the SHA computation is adapted from the Secure Hash Standard SHA-1 document that can be downloaded from the NIST website (www.itl.nist.gov/fipspubs/fip180-1.htm). The algorithm takes as its input data sixteen 32-bit words M_t ($0 \le t \le 15$), as shown in Tables 1, 3A, 3B, and 4 for the compute next secret, copy scratchpad, and read authenticated page command, respectively. The SHA computation involves a sequence of eighty 32-bit words called W_t ($0 \le t \le 79$), a sequence of eighty 32-bit words called K_t ($0 \le t \le 79$), a Boolean function f_t (B, C, D) ($0 \le t \le 79$) with B, C, and D being 32-bit words, and three more 32-bit words called A, E, and TMP. The operations required for the SHA computation are arithmetic addition without carry ("+"), logical inversion or 1's complement ("\"), EXCLUSIVE OR ("⊕"), logical AND ("∧"), logical OR ("∨"), assignment (":="), and circular shifting within a 32-bit word. The expression "$S^n(X)$" represents a circular shift of X by n positions to the left, with X being a 32-bit word.

The function f_t is defined as follows:

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

The sequence W_t ($0 \le t \le 79$) is defined as follows:

$W_t :=$	M_t	$(0 \le t \le 15)$
	$S^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16})$	$(16 \le t \le 79)$

The sequence K_t ($0 \le t \le 79$) is defined as follows:

$K_t :=$	5A827999h	$(0 \le t \le 19)$
	6ED9EBA1h	$(20 \le t \le 39)$
	8F1BBCDCh	$(40 \le t \le 59)$
	CA62C1D6h	$(60 \le t \le 79)$

The variables A, B, C, D, E are initialized as follows:

A	:=	67452301h
B	:=	EFCDAB89h
C	:=	98BADCFEh
D	:=	10325476h
E	:=	C3D2E1F0h

The 160-bit MAC is the concatenation of A, B, C, D, and E after looping through the following set of computations for t = 0 to 79 (discarding any carry-out):

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

The master can read the MAC with the read authenticated page command in a register and bit sequence as shown in Table 3. With the copy scratchpad command the bit transmission sequence is the same, however, the master has to compute the MAC and send it to the DS1961S. With the compute next secret command the MAC is not exposed. Instead, the contents of the D and E SHA computation registers are directly copied to the secret, as shown in Table 1.

1-WIRE BUS SYSTEM

The 1-Wire bus is a system, which has a single bus master and one or more slaves. In all instances the DS1961S is a slave device. The bus master is typically a microcontroller. For small configurations the 1-Wire communication signals can be generated under software control using a single port pin. For larger configurations, the DS2480B 1-Wire line driver chip or serial port adapters based on this chip (DS9097U series) are recommended. This simplifies the hardware design and frees the microprocessor from responding in real-time.

The discussion of this bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing). The 1-Wire protocol defines bus transactions in terms of the bus state during specific time slots that are initiated on the falling edge of sync pulses from the bus master. For a more detailed protocol description, refer to Chapter 4 of *The Book of DS19xx iButton Standards*.

HARDWARE CONFIGURATION

The 1-Wire bus has only a single line by definition; it is important that each device on the bus be able to drive it at the appropriate time. To facilitate this, each device attached to the 1-Wire bus must have open-drain or tri-state outputs. The 1-Wire port of the DS1961S is open drain with an internal circuit equivalent to that shown in Figure 8.

A multidrop bus consists of a 1-Wire bus with multiple slaves attached. At standard speed the 1-Wire bus has a maximum data rate of 16.3kbps. The speed can be boosted to 142kbps by activating the overdrive mode. The DS1961S is not guaranteed to be fully compliant to the iButton standard. Its maximum data rate in standard speed mode is 14.1kbps and 125kbps in overdrive. The DS1961S requires a 1-Wire pullup resistor of maximum 2.2kΩ for executing any of its memory and SHA function commands at any speed. When communicating with several DS1961S simultaneously, e.g., to install the same secret in several devices, the resistor should be bypassed by a low-impedance pullup to V_{PUP} while the device transfers data from the scratchpad to the EEPROM.

The idle state for the 1-Wire bus is high. If for any reason a transaction needs to be suspended, the bus must be left in the idle state if the transaction is to resume. If this does not occur and the bus is left low for more than 16µs (overdrive speed) or more than 120µs (regular speed), one or more devices on the bus can be reset. With the DS1961S the bus must be left low for no longer than 15.2µs at overdrive speed to ensure that none of the slave devices on the 1-Wire bus performs a reset. Despite of its limited compliance, the DS1961S communicates properly when used in conjunction with a DS2480B 1-Wire driver and serial port adapters that are based on this driver chip.