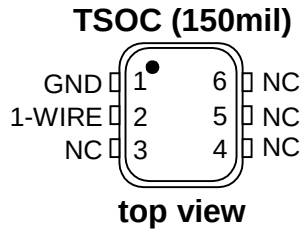
DALLAS SEMICONDUCTOR

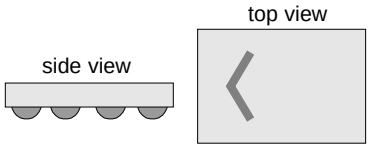
DS2432
1k-Bit Protected 1-Wire™ EEPROM with SHA-1 Engine

FEATURES

- 1128 bits of 5V EEPROM memory partitioned into four pages of 256 bits, a 64-bit write-only secret and up to 5 general purpose read/write registers
- On-chip 512-bit SHA-1 engine to compute 160-bit Message Authentication Codes (MAC) and to generate secrets
- Write access requires knowledge of the secret and the capability of computing and transmitting a 160-bit MAC 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)
- Unique, factory-lasered and tested 64-bit registration number assures absolute traceability because no two parts are alike
- Built-in multidrop controller ensures compatibility with other 1-Wire net products
- Reduces control, address, data and power to a single data pin
- Directly connects to a single port pin of a microprocessor and communicates at up to 16.3k bits per second
- Overdrive mode boosts communication speed to 142k bits per second
- Low cost 6-lead TSOC surface mount package, or solder-bumped Flipchip package
- Reads and writes over a wide voltage range of 2.8V to 5.25V from -40°C to +85°C

PIN ASSIGNMENT





See www.dalsemi.com for mechanical specifications of packages.

ORDERING INFORMATION

DS2432P	6-lead TSOC package
DS2432P/T&R	Tape & Reel DS2432P
DS2432X	Flipchip package, tape & reel

DESCRIPTION

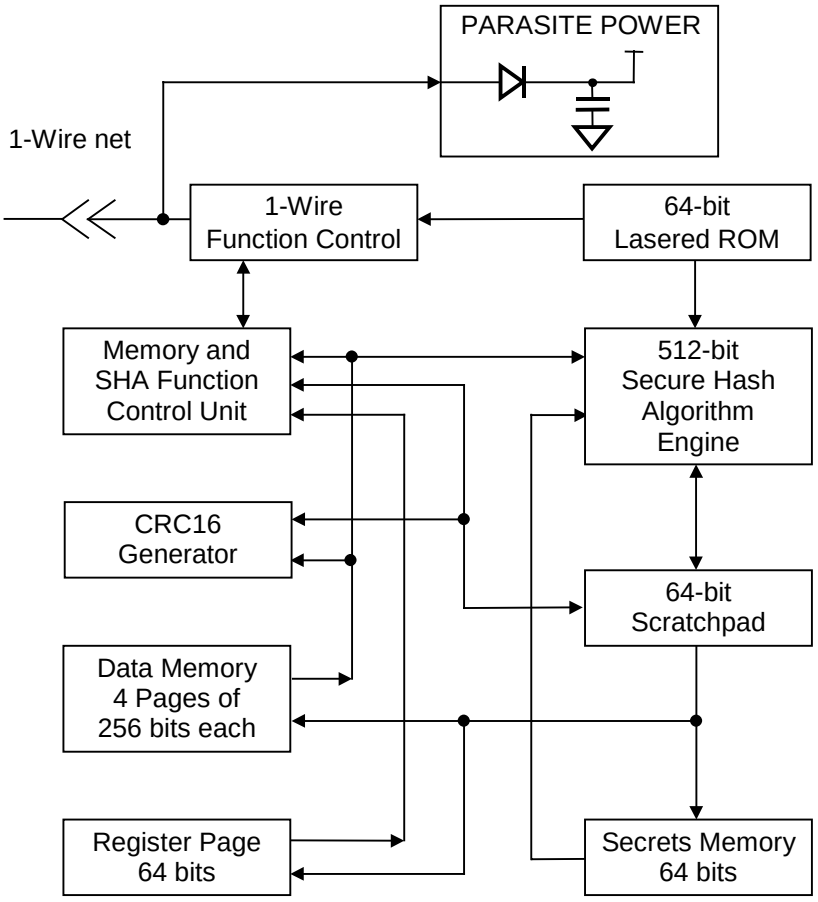
The DS2432 combines 1024 bits of EEPROM, a 64-bit secret, an 8-byte register/control page with up to 5 user read/write bytes, a 512-bit SHA-1 engine and a fully-featured 1-Wire interface in a single chip. Each DS2432 has its own 64-bit ROM registration number that is factory lasered into the chip to provide a guaranteed unique identity for absolute traceability. Data is transferred serially via the 1-Wire protocol, which requires only a single data lead and a ground return. The DS2432 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 will transfer the data to its final memory location, provided that the DS2432 receives a matching 160-Bit MAC. The computation of the MAC involves the secret and additional data stored in the DS2432 including the device’s registration number. Only a new secret can be loaded without providing a MAC. The SHA-1 engine can also be activated to compute

160-bit message authentication codes (MAC) when reading a memory page or to compute a new secret, instead of loading it. Applications of the DS2432 include intellectual property security, after-market management of consumables, and taper proof data carriers.

OVERVIEW

The block diagram in Figure 1 shows the relationships between the major control and memory sections of the DS2432. The DS2432 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 Engine (SHA = Secure Hash Algorithm). 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 will enter 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 may provide any one of the seven memory function commands. The protocol for these memory function commands is described in Figure 7. All data is read and written least significant bit first.

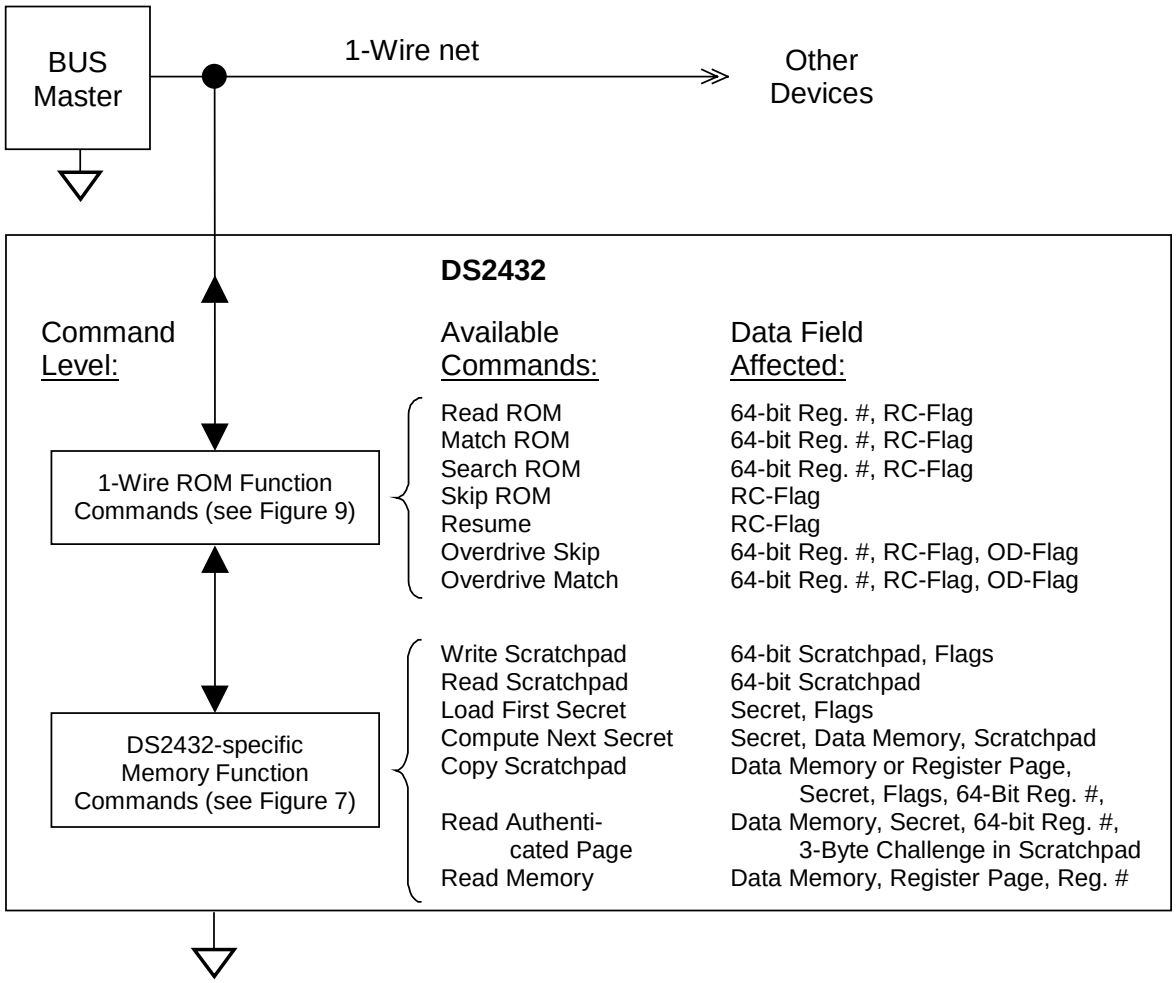
DS2432 BLOCK DIAGRAM Figure 1



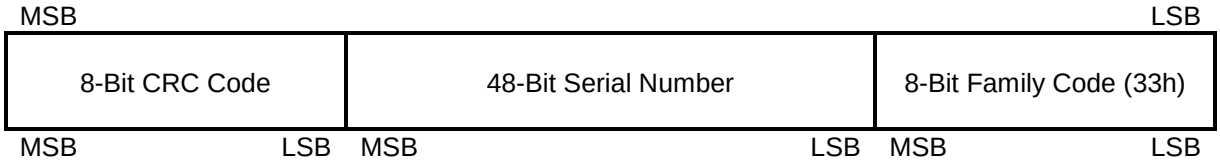
64-BIT LASERED ROM

Each DS2432 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 Cyclic Redundancy Check is available in the Book of DS19xx iButton Standards from Dallas Semiconductor. The shift register bits are initialized to zero. Then starting with the least significant bit 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.

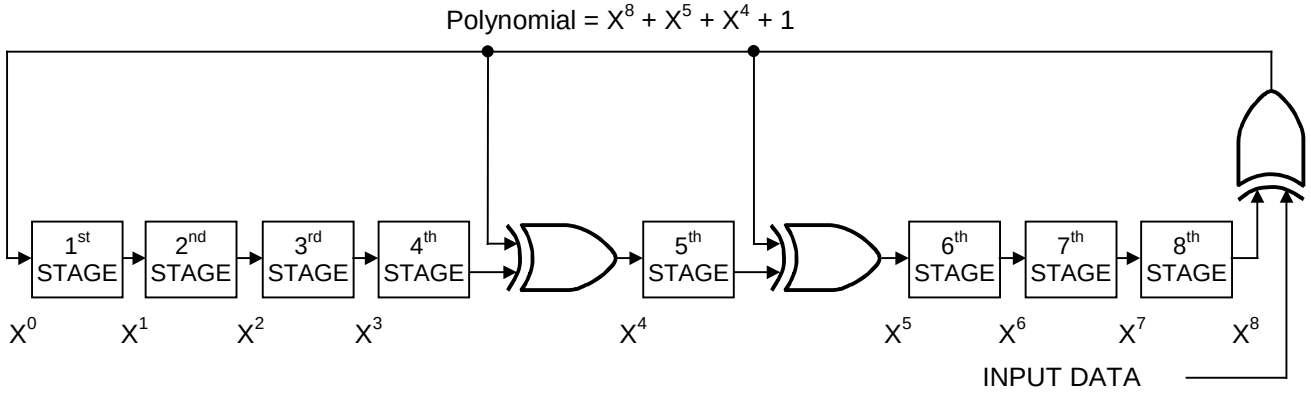
HIERARCHCAL STRUCTURE FOR 1-WIRE PROTOCOL Figure 2



64-BIT LASERED ROM Figure 3



1-WIRE CRC GENERATOR Figure 4



MEMORY MAP

The DS2432 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 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 the knowledge of the secret.

DS2432 MEMORY MAP Figure 5

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 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 Registration Number	(Alternate readout)

[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 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 will 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 the 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 will either read 55H or AAh. Typically, this address will read 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 DS2432 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 provides an alternate way to read 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 will receive the individual bits of the registration number in exactly the same sequence as with a ROM function command.

ADDRESS REGISTERS Figure 6

Bit #	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 DS2432 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 DS2432. Registers TA1 and TA2 must be loaded with the target address to which the data will be written or from which data will be read. Register E/S is a read-only transfer-status register, used to verify data integrity with write commands. Since the scratchpad of the DS2432 is designed to accept data in blocks of eight bytes only, the lower three bits of TA1 will be forced to 0 and the lower three bits of the E/S register (Ending Offset) will always read 1. This indicates that all the data in the scratchpad will be 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 8 or if the data in the scratchpad is not valid due to a loss of power. A valid write to the scratchpad will clear 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, called AA or Authorization Accepted, acts as a flag to indicate 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 DS2432, the scratchpad has to be used as intermediate storage. First the master issues the Write Scratchpad command to specify the desired target address, followed by the data to be written to the scratchpad. Note that writes to data memory must be performed on 8-byte boundaries with the 3 LSBs of the target address (T2..T0) equal to 000b. If T2..T0 are sent with non-zero values, the device will set these bits to zero and will write to the modified address upon completion of the command sequence. In addition, the entire 8-byte scratchpad will be copied to memory when commanded, therefore eight bytes of data should be written into the scratchpad to ensure that the data to be copied is known. Under certain conditions (see Write Scratchpad command) the master will receive an inverted CRC16 of the command, address (actual address sent) and data at the end of the write scratchpad command sequence. Note that the CRC is calculated based on the actual target address sent and not the modified address in the case of a non-zero T2..T0. Knowing this CRC value, the master can compare it to the value it has calculated itself to decide if the communication was successful and proceed to the Copy Scratchpad command. If the master could not receive the CRC16, it should send the Read Scratchpad command to verify data integrity. As preamble to the scratchpad data, the DS2432 repeats the target address TA1 and TA2 and sends the contents of the E/S register. If the PF flag is set, data did not arrive correctly in the scratchpad or there was a loss of power since data was last written to the scratchpad. The master does not need to continue reading; it can start a new trial to write data to the scratchpad. Similarly, a set AA flag together with a cleared PF flag indicates that the device did not recognize the Write command. If everything went correctly, both flags are cleared. Now the master can continue reading and verifying every data byte. After the master has verified the data, it can send the Copy Scratchpad command, for example. This command must be followed exactly by the data of the three address registers TA1, TA2 and E/S. The master should obtain the contents of these registers by reading the scratchpad.

MEMORY AND SHA FUNCTION COMMANDS

Due to its design as a secure device the DS2432 has to behave differently from other 1-Wire memory devices. Although most of the memory of the DS2432 can be read the same way as any other 1-Wire memory, attempts to read the secret will result 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 DS2432 takes place either at regular speed (default, OD = 0) or at Overdrive Speed (OD = 1). If not explicitly set into the Overdrive Mode the DS2432 assumes regular speed.

Write Scratchpad [0Fh]

The Write Scratchpad command applies to the data memory, the secret and the writable addresses in the register page. If the bus master sends a target address higher than 90h, the command will not be 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 will be written to the scratchpad starting at the beginning of the scratchpad. Note that the ending offset (E2..E0) will always be 111b regardless of the number of bytes that the master has transmitted. For this reason the master should always send

8 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 may be random data that is unknown to the master. Only full data bytes are accepted. If the last data byte is incomplete its content will be ignored and the partial byte flag PF will be set.

When executing the Write Scratchpad command the CRC generator inside the DS2432 (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 DS2432 will set TA1 bits T2..T0 to 000b for the actual Write Scratchpad command. The master may end the Write Scratchpad command at any time. However, if the scratchpad is filled to its capacity, the master may send 16 read time slots and will receive the CRC generated by the DS2432.

If a Write Scratchpad is attempted with a target address in data memory (00h-7Fh) or the register page (88h to 8Fh), then a subsequent Read Scratchpad command will read AAh or 55h for addresses that are write-protected rather than the value that was written in the Write Scratchpad command. Similarly, if the target address is within page 1 and the page is in EPROM mode, the read-back from the scratchpad will produce data that is the logical AND of the original scratchpad data and the current content of the target memory area.

Read Scratchpad [AAh]

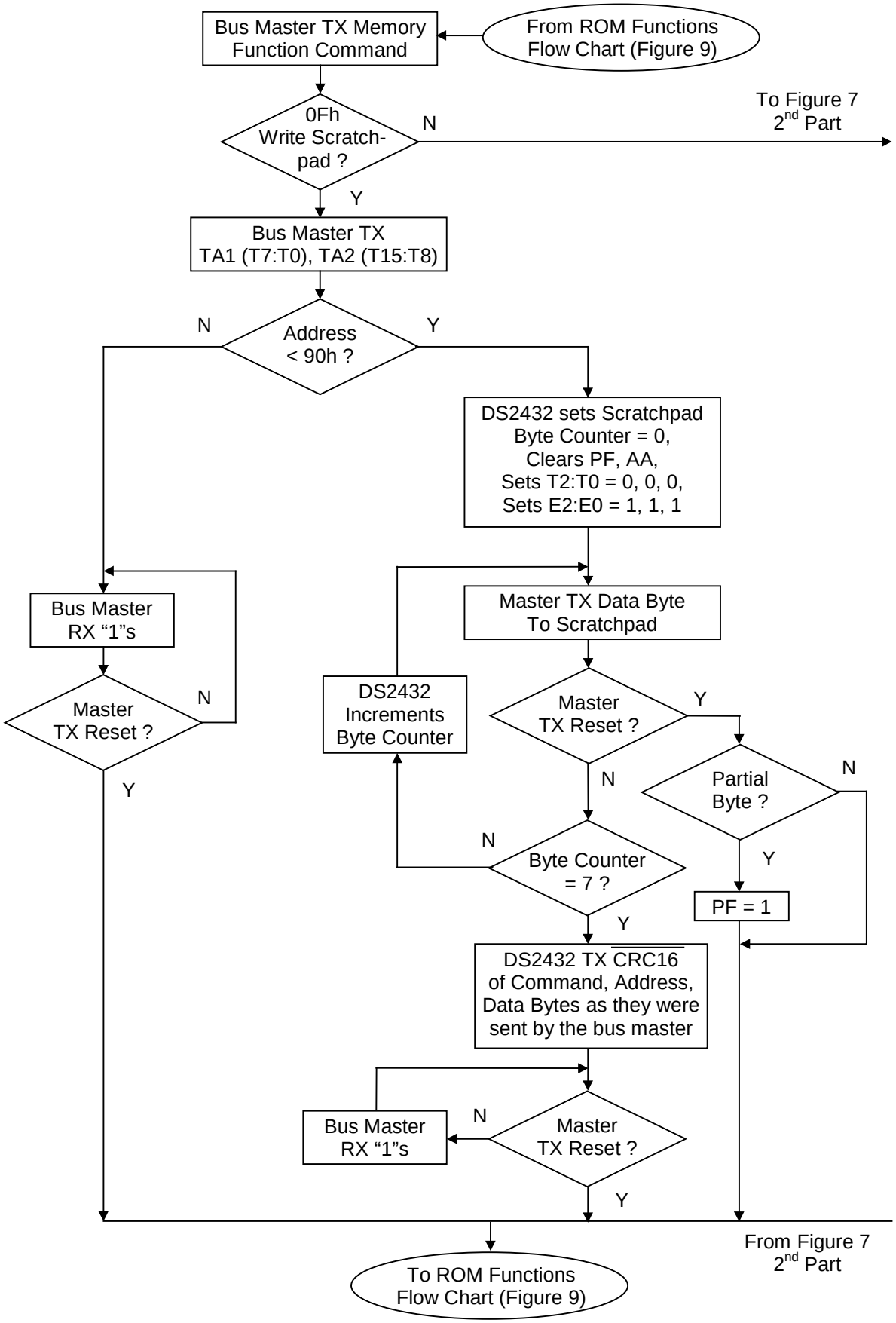
The Read Scratchpad command allows verifying the target address and the integrity of the scratchpad data. After issuing the command code, the master begins reading. The first two bytes will be the target address with T2 to T0 = 0. The next byte will be the ending offset/data status byte (E/S) followed 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 or page 1 in EPROM mode. The master should read through the end of the scratchpad after which it will receive the inverted CRC. This is based on data as it was sent by the DS2432. If the master continues reading after the CRC all data will be logic 1’s.

Load First Secret [5Ah]

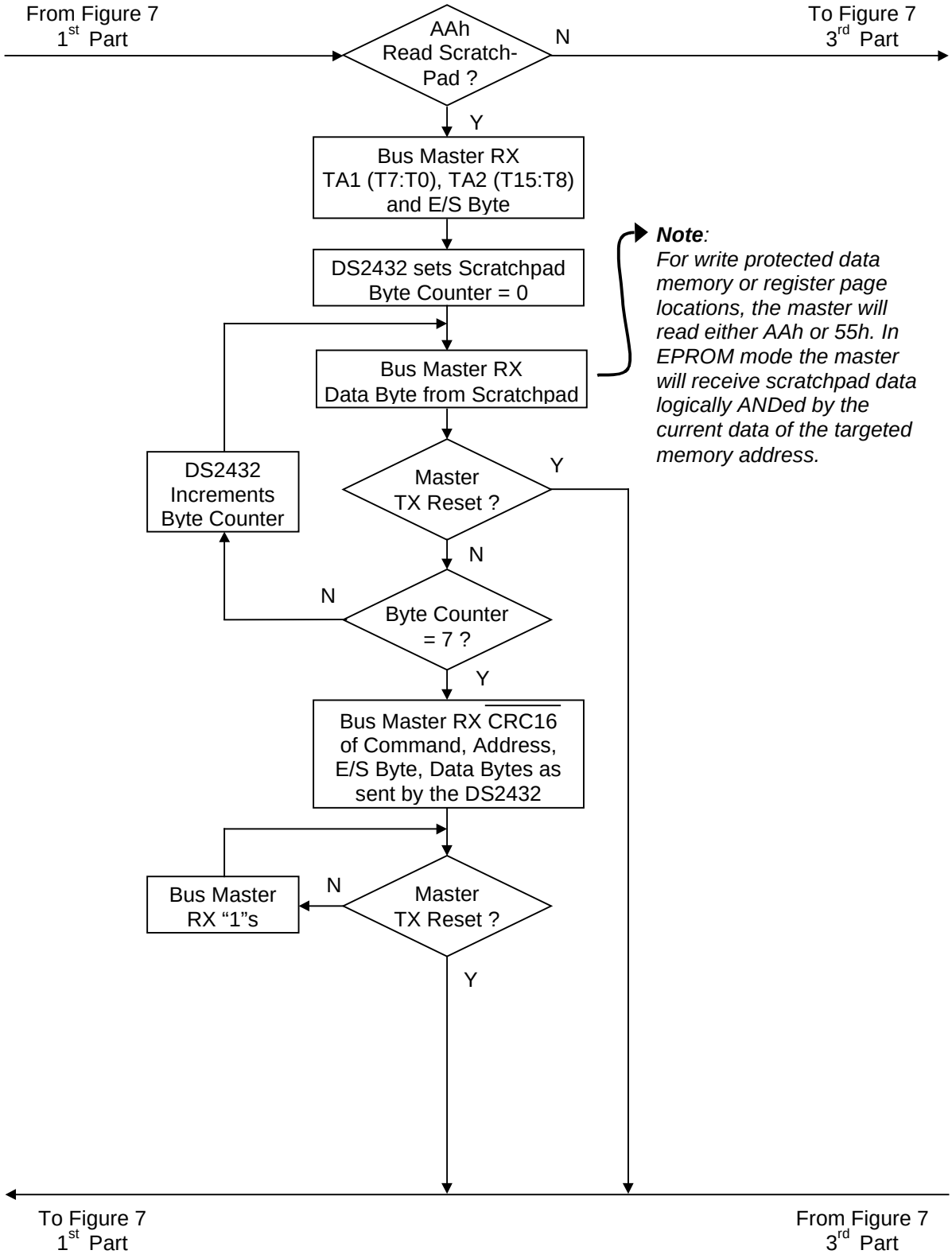
The Load First Secret command is used to replace the device’s current secret with the contents of the scratchpad, provided that the secret is not write-protected. This command does not require the knowledge of the device’s current secret. Before the Load First Secret command can be used the master must 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, 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 pattern matches and the secret is not write-protected, the AA (Authorization Accepted) flag will be set and the copy will begin. All eight bytes of scratchpad contents will be copied to the secret’s memory location. The device-internal data transfer takes 10 ms maximum during which the voltage on the 1-Wire bus must not fall below 2.8V. A pattern of alternating 1’s and 0’s will be transmitted after the data has been copied until the master issues a reset pulse.

Instead of using Load First Secret, a new secret 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.

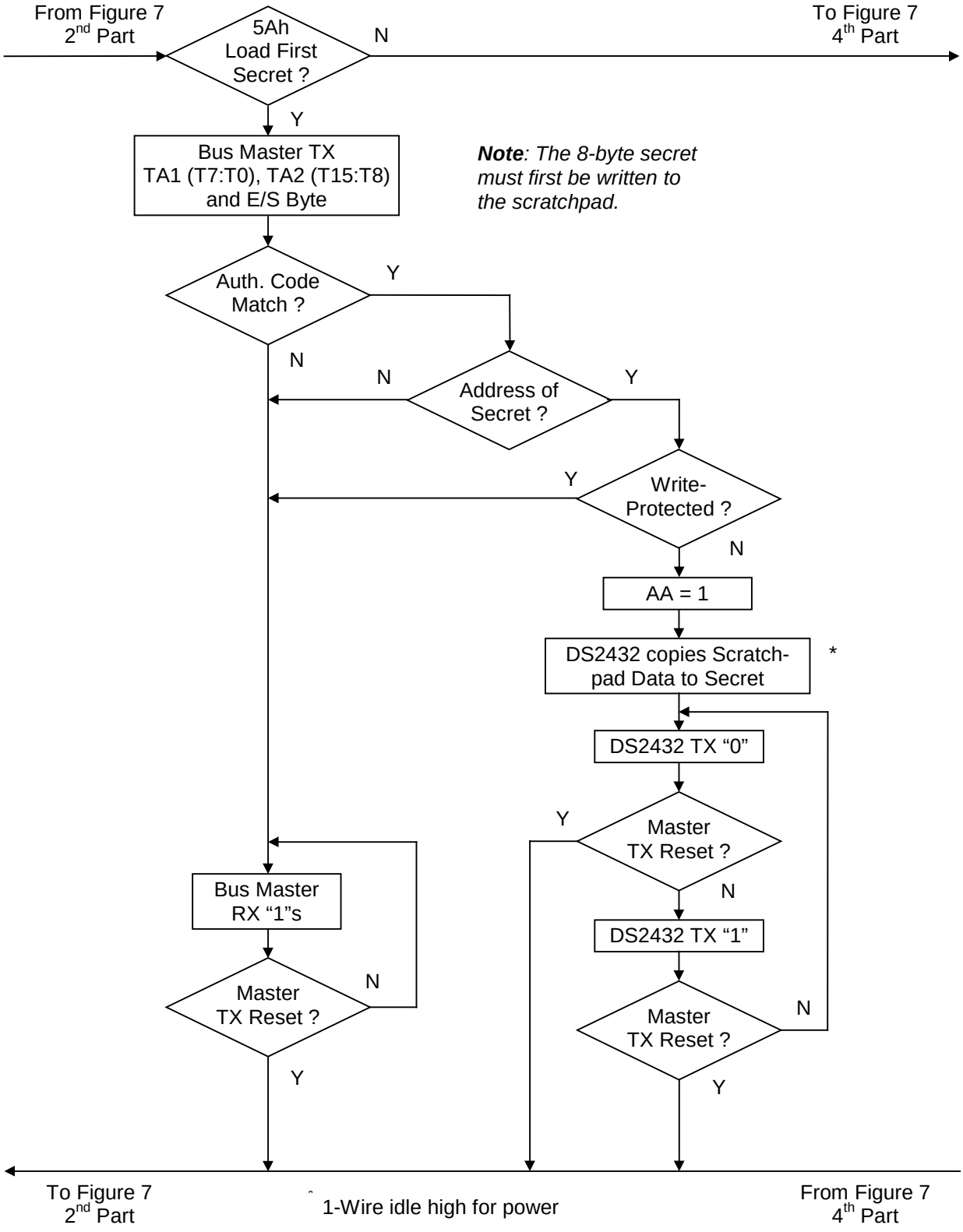
Memory and SHA Functions Flow Chart Figure 7



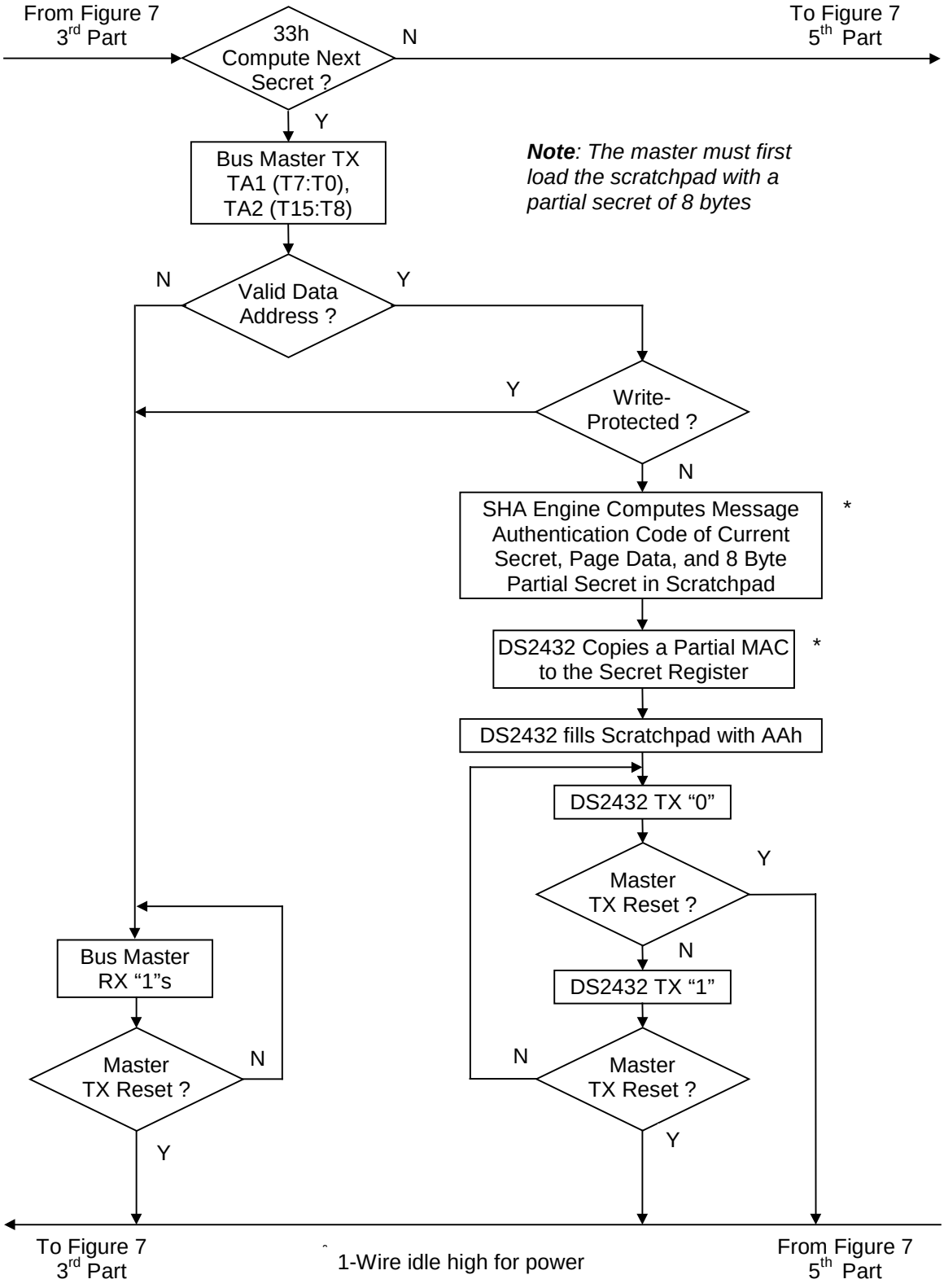
Memory and SHA Functions Flow Chart (continued) Figure 7



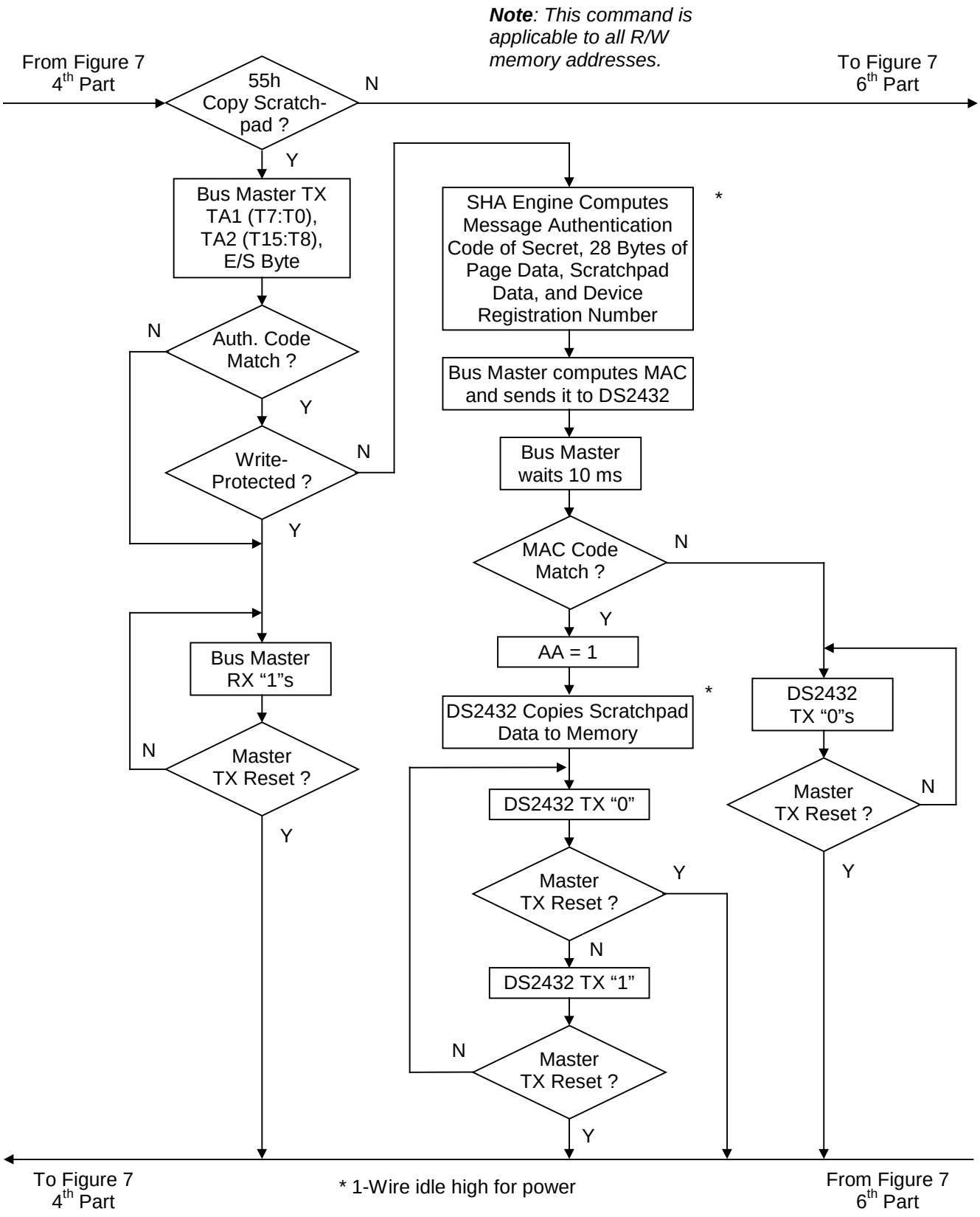
Memory and SHA Functions Flow Chart (continued) Figure 7



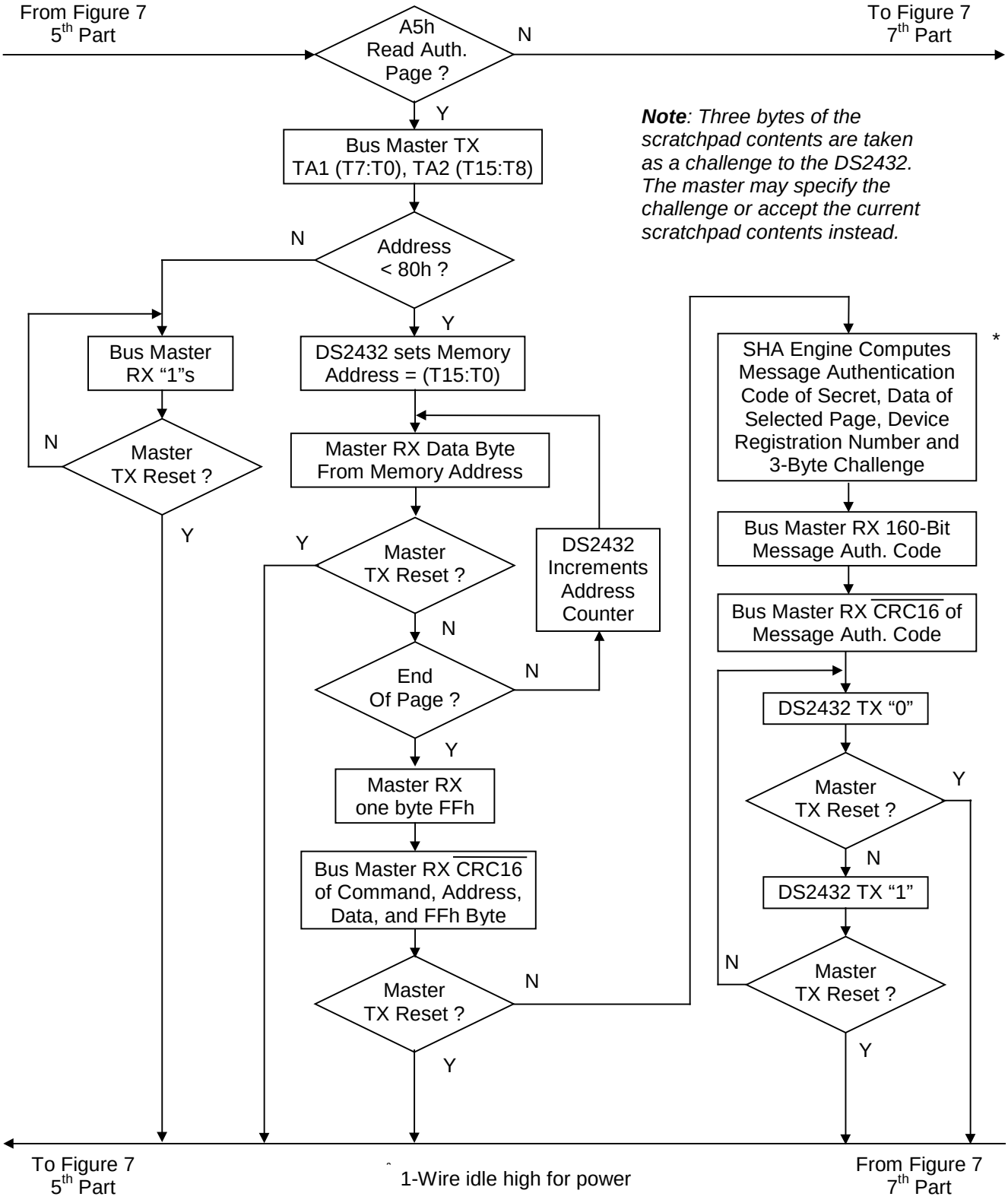
Memory and SHA Functions Flow Chart (continued) Figure 7



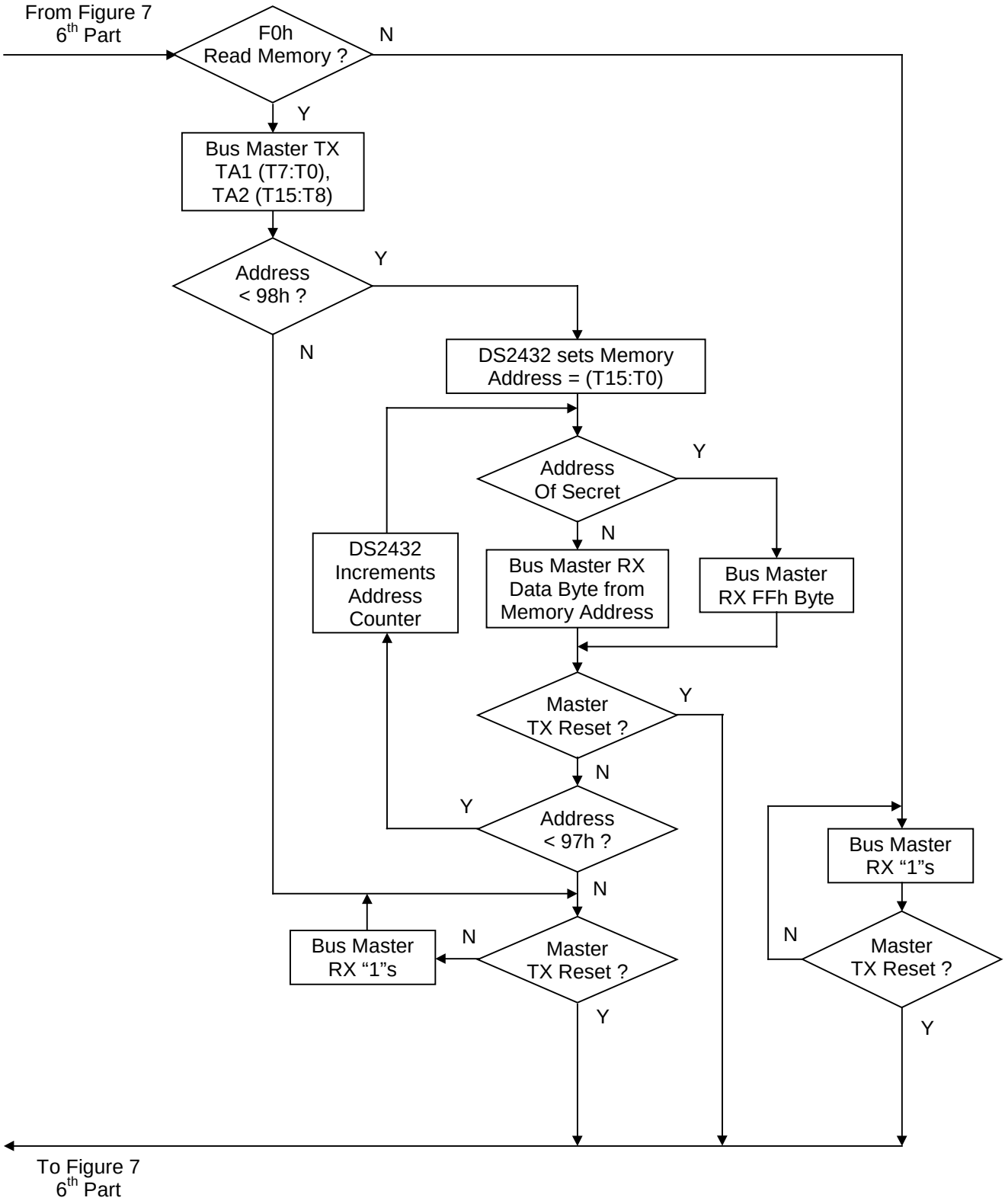
Memory and SHA Functions Flow Chart (continued) Figure 7



Memory and SHA Functions Flow Chart (continued) Figure 7



Memory and SHA Functions Flow Chart (continued) Figure 7



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 DS2432 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.

SHA-1 Input Data for Compute Next Secret Command Table 1

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** $0 \le t \le 15$; 32-bit words
SS	**Starting address of secret (80h)**
PP	**Starting address of memory page** See Memory Map, memory pages 0 through 3
(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. The lower five bits of the target address TA1 are not relevant. If the target address is valid, i. e. is in the range of 0000h to 007Fh, and the secret is

not write-protected the SHA engine will start and within 2.0 ms compute a new secret that is then automatically copied to the secrets register. Replacing the secret takes maximum 10 ms. During this time and the computation of the secret the voltage on the 1-Wire bus must not fall below 2.8V. After copying is finished the DS2432 fills the scratchpad with AAh bytes. Now a pattern of alternating 1's and 0's will be transmitted until the master issues a reset pulse.

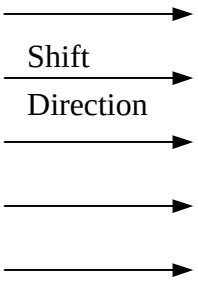
Since the content of the scratchpad is used as a partial secret, the master must fill the scratchpad with a known 8-byte data pattern using the Write Scratchpad command **before** it issues the Compute Next Secret command. Otherwise the new secret will depend on data that was unintentionally left in the scratchpad from previous commands.

Copy Scratchpad [55h]

The data memory of the DS2432 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 a SHA-1 computation to generate the 160-bit Message Authentication Code (MAC) to start the data transfer from the scratchpad to the memory. The master may 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 DS2432 is shown in Table 2. Table 3 shows how the various data components are entered into the SHA engine. The SHA computation algorithm is explained later in this document.

Message Authentication Code Transmission Sequence Table 2

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 DS2432 will start 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 DS2432's registration number (without the CRC). Simultaneously the master computes a MAC from the same data and sends it to the DS2432 as evidence that it is authorized to write to the EEPROM. Now the master waits for 10 ms during which the voltage on the 1-Wire bus must not fall below 2.8V. If the MAC generated by the DS2432 matches the MAC that the master computed, the DS2432 will set its AA (Authorization Accepted) flag, and copy the entire scratchpad contents to the data EEPROM. As indication for a successful copy the master will be able to read a pattern of alternating 1's and 0's until it issues a Reset Pulse. A pattern of all zeros tells the master that the copy did not take place.

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 all with the intended modification to the scratchpad. When writing to the register page (or the secret using Copy Scratchpad), the input data for M1 to M7 of the SHA engine will be the current secret (M1, M2), the current content of the register page (M3, M4), the full 64-bit registration number (M5, M6), and 4 bytes FFh (M7).

SHA-1 Input Data for Copy Scratchpad Command Table 3

M0[31:24] = (SS+0)	M0[23:16] = (SS+1)	M0[15:8] = (SS+2)	M0[7:0] = (SS+3)
M1[31:24] = (PP+0)	M1[23:16] = (PP+1)	M1[15:8] = (PP+2)	M1[7:0] = (PP+3)
M2[31:24] = (PP+4)	M2[23:16] = (PP+5)	M2[15:8] = (PP+6)	M2[7:0] = (PP+7)
M3[31:24] = (PP+8)	M3[23:16] = (PP+9)	M3[15:8] = (PP+10)	M3[7:0] = (PP+11)
M4[31:24] = (PP+12)	M4[23:16] = (PP+13)	M4[15:8] = (PP+14)	M4[7:0] = (PP+15)
M5[31:24] = (PP+16)	M5[23:16] = (PP+17)	M5[15:8] = (PP+18)	M5[7:0] = (PP+19)
M6[31:24] = (PP+20)	M6[23:16] = (PP+21)	M6[15:8] = (PP+22)	M6[7:0] = (PP+23)
M7[31:24] = (PP+24)	M7[23:16] = (PP+25)	M7[15:8] = (PP+26)	M7[7:0] = (PP+27)
M8[31:24] = (SP+0)	M8[23:16] = (SP+1)	M8[15:8] = (SP+2)	M8[7:0] = (SP+3)
M9[31:24] = (SP+4)	M9[23:16] = (SP+5)	M9[15:8] = (SP+6)	M9[7:0] = (SP+7)
M10[31:24] = MP	M10[23:16] = 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] = 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	**Starting address of secret (80h)**
PP	**Starting address of memory page** See Memory Map, memory pages 0 through 3
(SP+n)	**Byte n of scratchpad**
MP	MP[7:4] = 0000 for Copy Scratchpad MP[3:0] = T8:T5 (equivalent to page number in hex)
FAMC	**Family Code = 33h**
SNx	**Serial number of device** SN0 = least significant byte, SN5 = most significant byte. The CRC is not used

Read Authenticated Page [A5h]

The Read Authenticated Page command provides the master with the data of a full or partial memory page plus a message authentication code (MAC). The MAC allows the master to determine whether the secret stored in the DS2432 is valid within the application. The DS2432 computes the MAC from its secret, all the data of the selected memory page, its registration number 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 5^{th}, 6^{th} and 7^{th} byte. 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 DS2432 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 a valid target address it will receive 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. Immediately after the CRC is received the master waits for 2.0 ms during which the voltage on the 1-Wire bus must not fall below 2.8V. During this time the SHA engine of the DS2432 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 will receive a pattern of alternating 0's and 1's until it issues a Reset Pulse.

SHA-1 Input Data for Read Authenticated Page Command Table 4

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] = 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+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** $0 \le t \le 15$; 32-bit words
SS	**Starting address of secret (80h)**
PP	**Starting address of memory page** See Memory Map, memory pages 0 through 3
FAMC	**Family Code = 33h**
MP	MP[7:4] = 0100 MP[3:0] = T8:T5 (equivalent to page number in hex)
SNx	**ROM Serial number of device** SN0 = least significant byte, SN5 = most significant byte The CRC is not used
(SP+n)	**Byte n of Scratchpad**

Read Memory [F0h]

The read memory command may be used to read all memory except for the secret. Attempting to read the secret will not reveal any data. After issuing the command, the master must provide the 2-byte target address. After these two bytes, the master reads data beginning from the target address and may continue until address 0097h. If the master continues reading the result will be logic 1's. It is important to realize that the target address registers will point to the last byte read. The ending offset/data status byte and the scratchpad are unaffected.

The hardware of the DS2432 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 would typically store 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. (See 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 as it can be downloaded from the NIST web site (http://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, 2 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:

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

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

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

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

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

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 Message Authentication Code (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 DS2432. With the Compute Next Secret command the MAC is not exposed. Instead, the content of the SHA computation registers E and D is directly copied to the secret register, as shown in Table 1.

1-WIRE BUS SYSTEM

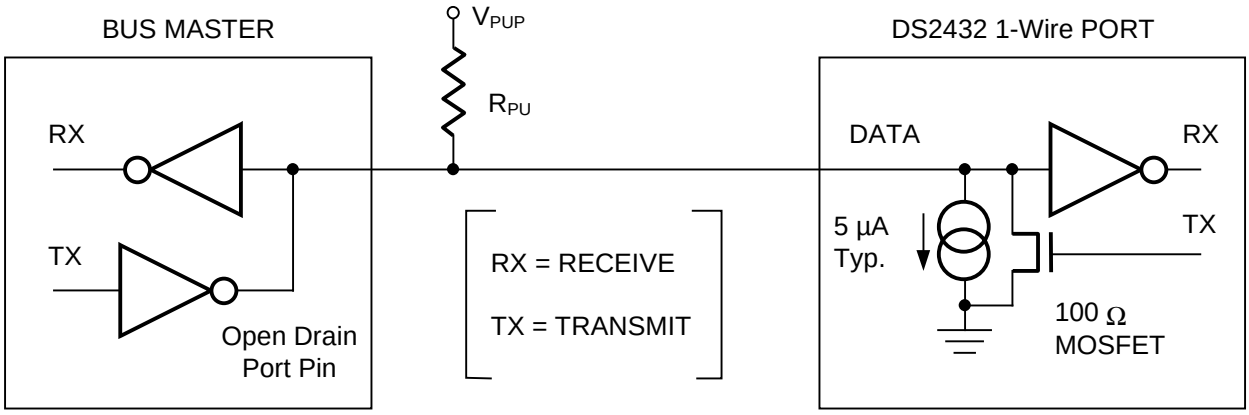
The 1-Wire bus is a system, which has a single bus master and one or more slaves. In all instances the DS2432 is a slave device. The bus master is typically a microcontroller. The discussion of this bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing). A 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 3-state outputs. The 1-Wire port of the DS2432 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 regular speed the 1-Wire bus has a maximum data rate of 16.3k bits per second. The speed can be boosted to 142k bits per second by activating the Overdrive Mode. The DS2432 requires a 1-Wire pull-up resistor of maximum 2.2 kΩ for executing any of its memory and SHA function commands at any speed. When communicating with several DS2432 simultaneously, e. g., to install the same secret in several devices, the resistor should be bypassed by a low-impedance pull-up to V_{PUP} while the device transfers data from the scratchpad to the EEPROM and updates the tamper-detect register.

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 16 µs (Overdrive Speed) or more than 120 µs (regular speed), one or more devices on the bus may be reset.

HARDWARE CONFIGURATION Figure 8



TRANSACTION SEQUENCE

The protocol for accessing the DS2432 via the 1-Wire port is as follows:

- Initialization
- ROM Function Command
- Memory or SHA Function Command
- Transaction/Data

INITIALIZATION

All transactions on the 1-Wire bus begin with an initialization sequence. The initialization sequence consists of a reset pulse transmitted by the bus master followed by presence pulse(s) transmitted by the slave(s). The presence pulse lets the bus master know that the DS2432 is on the bus and is ready to operate. For more details, see the "1-Wire Signaling" section.

ROM FUNCTION COMMANDS

Once the bus master has detected a presence, it can issue one of the seven ROM function commands that the DS2432 supports. All ROM function commands are eight bits long. A list of these commands follows (refer to flowchart in Figure 9):

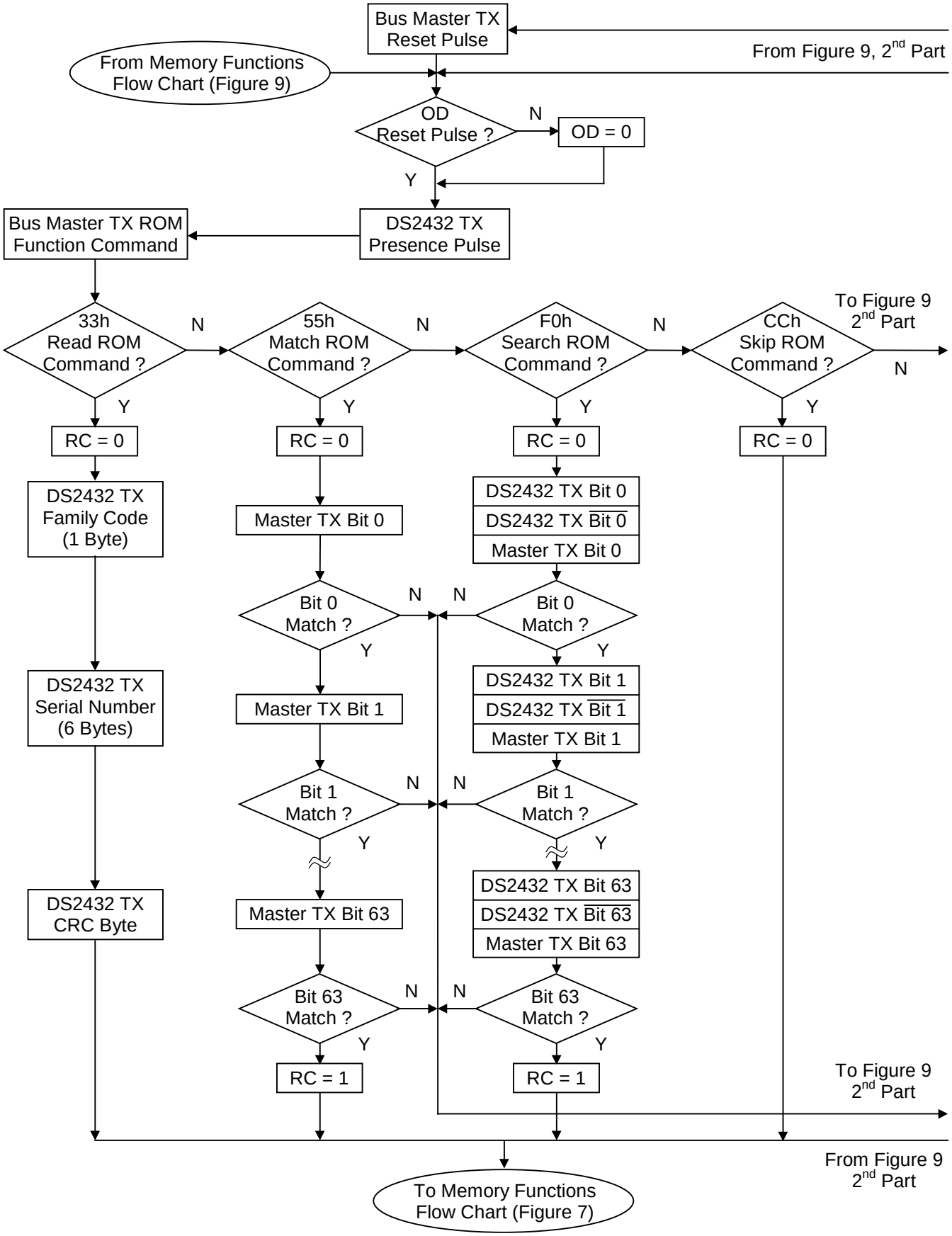
Read ROM [33h]

This command allows the bus master to read the DS2432's 8-bit family code, unique 48-bit serial number, and 8-bit CRC. This command should only be used if there is a single slave on the bus. If more than one slave is present on the bus, a data collision will occur when all slaves try to transmit at the same time (open drain will produce a wired-AND result). The resultant family code and 48-bit serial number read by the master will be invalid.

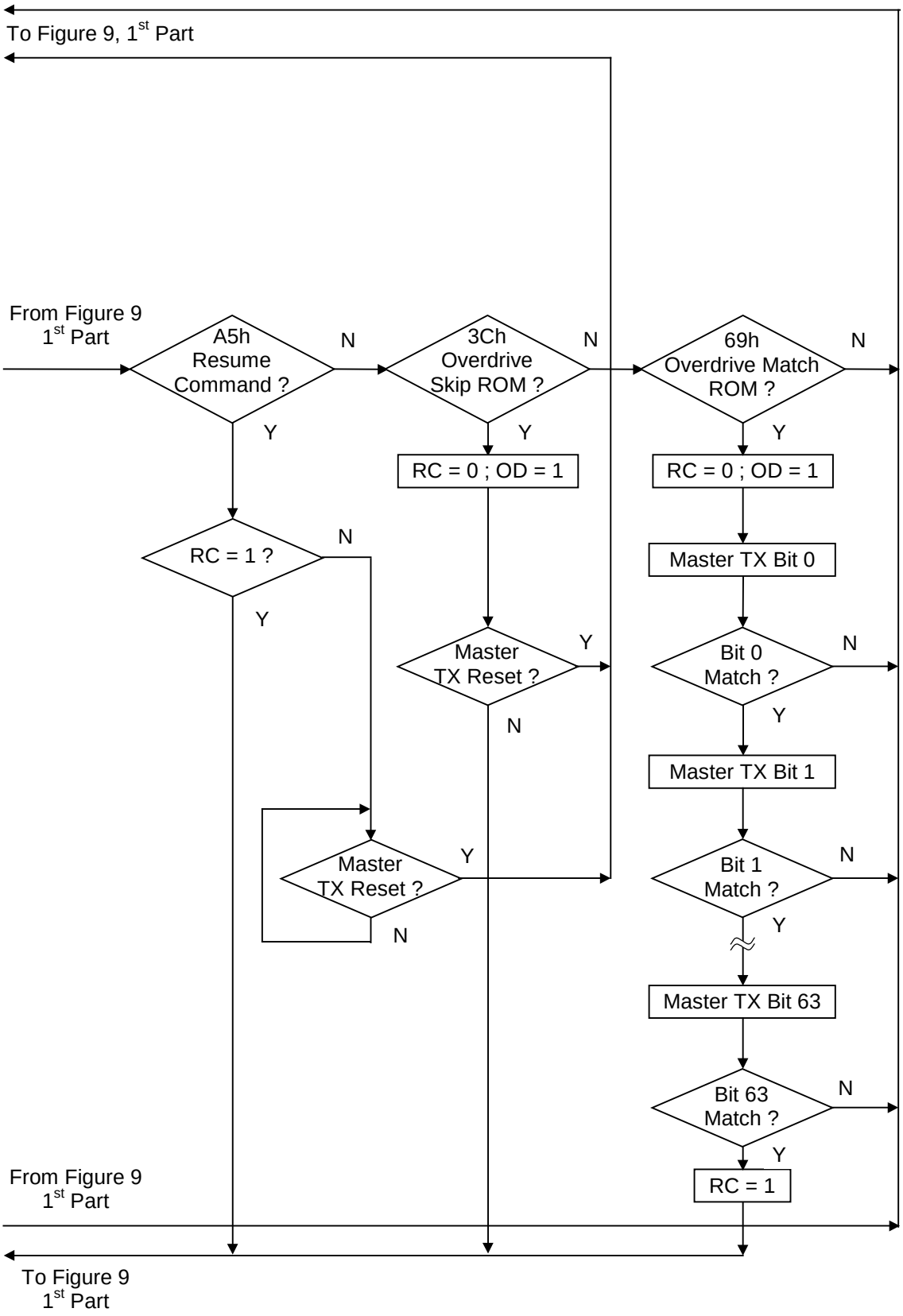
Match ROM [55h]

The match ROM command, followed by a 64-bit registration number, allows the bus master to address a specific DS2432 on a multidrop bus. Only the DS2432 that exactly matches the 64-bit registration number will respond to the following memory function command. All other slaves will wait for a reset pulse. This command can be used with a single or multiple devices on the bus.

ROM FUNCTIONS FLOW CHART Figure 9



ROM FUNCTIONS FLOW CHART (continued) Figure 9



Search ROM [F0h]

When a system is initially brought up, the bus master might not know the number of devices on the 1-Wire bus or their 64-bit registration numbers. The search ROM command allows the bus master to use a process of elimination to identify the 64-bit numbers of all slave devices on the bus. The search ROM process is the repetition of a simple 3-step routine: read a bit, read the complement of the bit, then write the desired value of that bit. The bus master performs this 3-step routine on each bit of the registration number. After one complete pass, the bus master knows the 64-bit number of one device. Additional passes will identify the registration numbers of the remaining devices. See Chapter 5 of the Book of DS19xx iButton Standards for a detailed discussion of a search ROM, including an actual example.

Skip ROM [CCh]

This command can save time in a single drop bus system by allowing the bus master to access the memory and SHA functions without providing the 64-bit registration number. If more than one slave is present on the bus and, for example, a read command is issued following the Skip ROM command, data collision will occur on the bus as multiple slaves transmit simultaneously (open drain pull-downs will produce a wired-AND result).

Overdrive Skip ROM [3Ch]

On a single-drop bus this command can save time by allowing the bus master to access the memory and SHA functions without providing the 64-bit registration number. Unlike the normal Skip ROM command the Overdrive Skip ROM sets the DS2432 in the Overdrive Mode (OD = 1). All communication following this command code has to occur at Overdrive Speed until a reset pulse of minimum 480 µs duration resets all devices on the bus to regular speed (OD = 0).

When issued on a multidrop bus this command will set all Overdrive-supporting devices into Overdrive mode. To subsequently address a specific Overdrive-supporting device, a reset pulse at Overdrive speed has to be issued followed by a Match ROM or Search ROM command sequence. This will speed up the search process. If more than one Overdrive-supporting slave is present on the bus and the Overdrive Skip ROM command is followed by a read command, data collision will occur on the bus as multiple slaves transmit simultaneously (open drain pull-downs will produce a wired-AND result).

Overdrive Match ROM [69h]

The Overdrive Match ROM command, followed by a 64-bit registration number transmitted at Overdrive Speed, allows the bus master to address a specific DS2432 on a multidrop bus and to simultaneously set it in Overdrive Mode. Only the DS2432 that exactly matches the 64-bit number will respond to the subsequent memory or SHA function command. Slaves already in Overdrive mode from a previous Overdrive Skip or a successful Overdrive Match command will remain in Overdrive mode. All Overdrive-capable slaves will return to regular speed at the next Reset Pulse of minimum 480 µs duration. The Overdrive Match ROM command can be used with a single or multiple devices on the bus.

Resume Command [A5h]

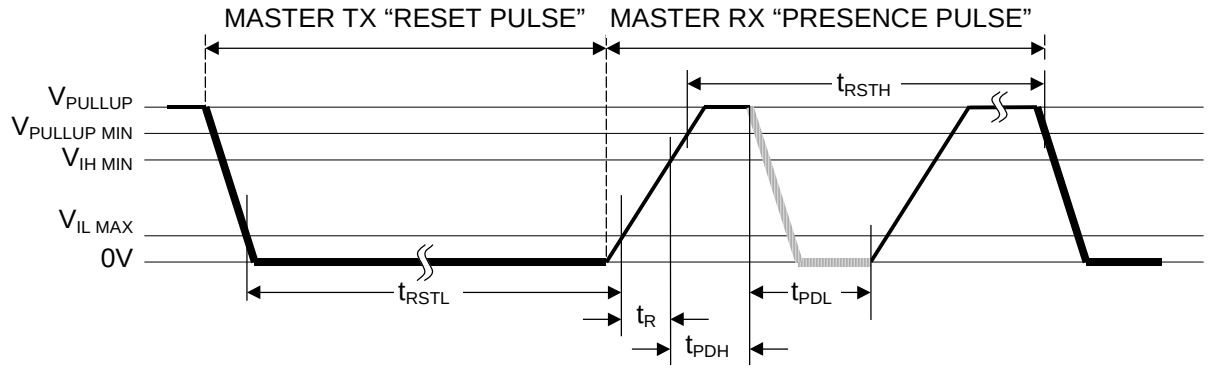
In a typical application the DS2432 needs to be accessed several times to write a full 32-byte page. In a multidrop environment this means that the 64-bit registration number of a Match ROM command has to be repeated for every access. To maximize the data throughput in a multidrop environment the Resume Command function was implemented. This function checks the status of the RC bit and, if it is set, directly transfers control to the Memory and SHA functions, similar to a Skip ROM command. The only way to set the RC bit is through successfully executing the Match ROM, Search ROM or Overdrive Match ROM command. Once the RC bit is set, the device can repeatedly be accessed through the Resume Command function. Accessing another device on the bus will clear the RC bit, preventing two or more devices from simultaneously responding to the Resume Command function.

1-WIRE SIGNALING

The DS2432 requires strict protocols to ensure data integrity. The protocol consists of four types of signaling on one line: Reset Sequence with Reset Pulse and Presence Pulse, Write 0, Write 1 and Read Data. Except for the presence pulse the bus master initiates all these signals. The DS2432 can communicate at two different speeds, regular speed and Overdrive Speed. If not explicitly set into the Overdrive mode, the DS2432 will communicate at regular speed. While in Overdrive Mode the fast timing applies to all waveforms.

The initialization sequence required to begin any communication with the DS2432 is shown in Figure 10. A Reset Pulse followed by a Presence Pulse indicates the DS2432 is ready to send or receive data. The bus master transmits (TX) a reset pulse (t_{RSTL}, minimum 480 µs at regular speed, 48 µs at Overdrive Speed). The bus master then releases the line and goes into receive mode (RX). The 1-Wire bus is pulled to a high state via the pull-up resistor. After detecting the rising edge on the data pin, the DS2432 waits (t_{PDH}, 15-60 µs at regular speed, 2-6 µs at Overdrive speed) and then transmits the Presence Pulse (t_{PDL}, 60-240 µs at regular speed, 8-24 µs at Overdrive Speed). A Reset Pulse of 480 µs or longer will exit the Overdrive Mode returning the device to regular speed. If the DS2432 is in Overdrive Mode and the Reset Pulse is no longer than 80 µs the device will remain in Overdrive Mode.

INITIALIZATION PROCEDURE "RESET AND PRESENCE PULSES" Figure 10



REGULAR SPEED	OVERDRIVE SPEED
480 µs ≤ t_{RSTL} < ∞*	48 µs ≤ t_{RSTL} < 80 µs
480 µs ≤ t_{RSTH} < ∞**	48 µs ≤ t_{RSTH} < ∞ **
15 ≤ t_{PDH} < 60 µs	15 ≤ t_{PDH} < 6 µs
60 µs ≤ t_{PDL} < 240	8 µs ≤ t_{PDL} < 24

* In order not to mask interrupt signaling by other devices on the 1-Wire bus, t_{RSTL} + t_R should always be less than 960 µs.

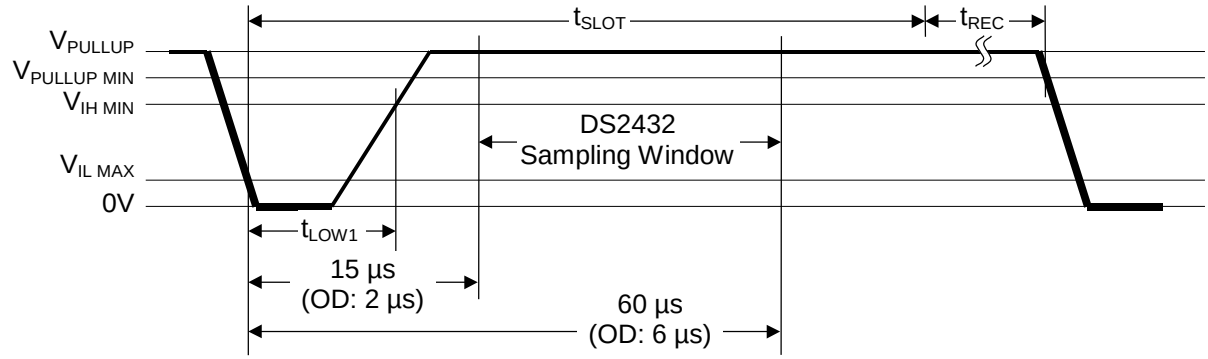
** Includes recovery time

Read/Write Time Slots

The definitions of write and read time slots are illustrated in Figure 11. The master initiates all time slots by driving the data line low. The falling edge of the data line synchronizes the DS2432 to the master by triggering an internal delay circuit. During write time slots, the delay circuit determines when the DS2432 will sample the data line. For a read data time slot, if a "0" is to be transmitted, the delay circuit determines how long the DS2432 will hold the data line low. If the data bit is a "1", the DS2432 will not hold the data line low at all.

READ/WRITE TIMING DIAGRAM Figure 11

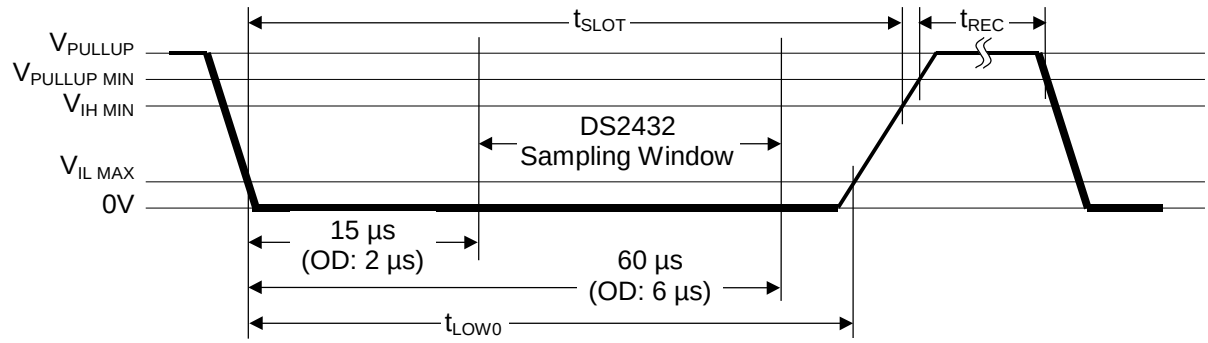
Write-one Time Slot



REGULAR SPEED
60 µs ≤ t_{SLOT} < 120 µs
1 µs ≤ t_{LOW1} < 15 µs
1 µs ≤ t_{REC} < ∞

OVERDRIVE SPEED
6 µs ≤ t_{SLOT} < 16 µs
1 µs ≤ t_{LOW1} < 2 µs
1 µs ≤ t_{REC} < ∞

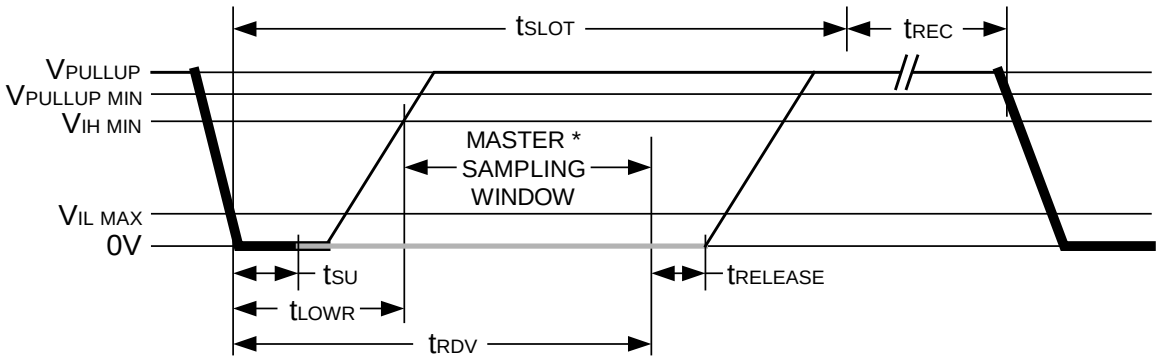
Write-zero Time Slot



REGULAR SPEED
60 µs ≤ t_{LOW0} < t_{SLOT} < 120 µs
1 µs ≤ t_{REC} < ∞

OVERDRIVE SPEED
6 µs ≤ t_{LOW0} < t_{SLOT} < 16 µs
1 µs ≤ t_{REC} < ∞

Read-data Time Slot



Waveform Legend:
- RESISTOR
- MASTER
- DS2432

REGULAR SPEED	OVERDRIVE SPEED
$60\mu s \le t_{SLOT} < 120\mu s$	$6\mu s \le t_{SLOT} < 16\mu s$
$1\mu s \le t_{LOWR} < 15\mu s$	$1\mu s \le t_{LOWR} < 2\mu s$
$0\mu s \le t_{RELEASE} < 45\mu s$	$0\mu s \le t_{RELEASE} < 4\mu s$
$1\mu s \le t_{REC} < \infty$	$1\mu s \le t_{REC} < \infty$
$t_{RDV} = 15\mu s$ *	$t_{RDV} = 2\mu s$ *
$t_{SU} < 1\mu s$	$t_{SU} < 1\mu s$

*The optimal sampling point for the master is as close as possible to the end time of the t_{RDV} period without exceeding t_{RDV}. For the case of a Read-one time slot, this maximizes the amount of time for the pull-up resistor to recover the line to a high level. For a Read-zero time slot it ensures that a read will occur before the fastest 1-Wire device releases the line ($t_{RELEASE}$ = 0).

CRC GENERATION

With the DS2432 there are two different types of CRCs (Cyclic Redundancy Checks). One CRC is an 8-bit type. It is computed at the factory and lasered into the most significant byte of the 64-bit ROM. The equivalent polynomial function of this CRC is $X^8 + X^5 + X^4 + 1$. To determine whether the ROM data has been read without error the bus master can compute the CRC value from the first 56 bits of the 64-bit ROM and compare it to the value read from the DS2432. This 8-bit CRC is received in the true form (non-inverted) when reading the ROM.

The other CRC is a 16-bit type, generated according to the standardized CRC16-polynomial function $X^{16} + X^{15} + X^2 + 1$. This CRC is used for error detection with the Read Authenticated Page command, when reading the scratchpad and for fast verification of a data transfer when writing to the scratchpad. It is the same type of CRC as is used for error detection within the iButton Extended File Structure. In contrast to the 8-bit CRC, the 16-bit CRC is always returned or sent in the complemented (inverted) form. A CRC-generator inside the DS2432 chip (Figure 12) will calculate a new 16-bit CRC as shown in the command flow chart of Figure 7. The bus master may compare the CRC value read from the device to the one it calculates from the data and decide whether to continue with an operation or to re-read the portion of the data with the CRC error.

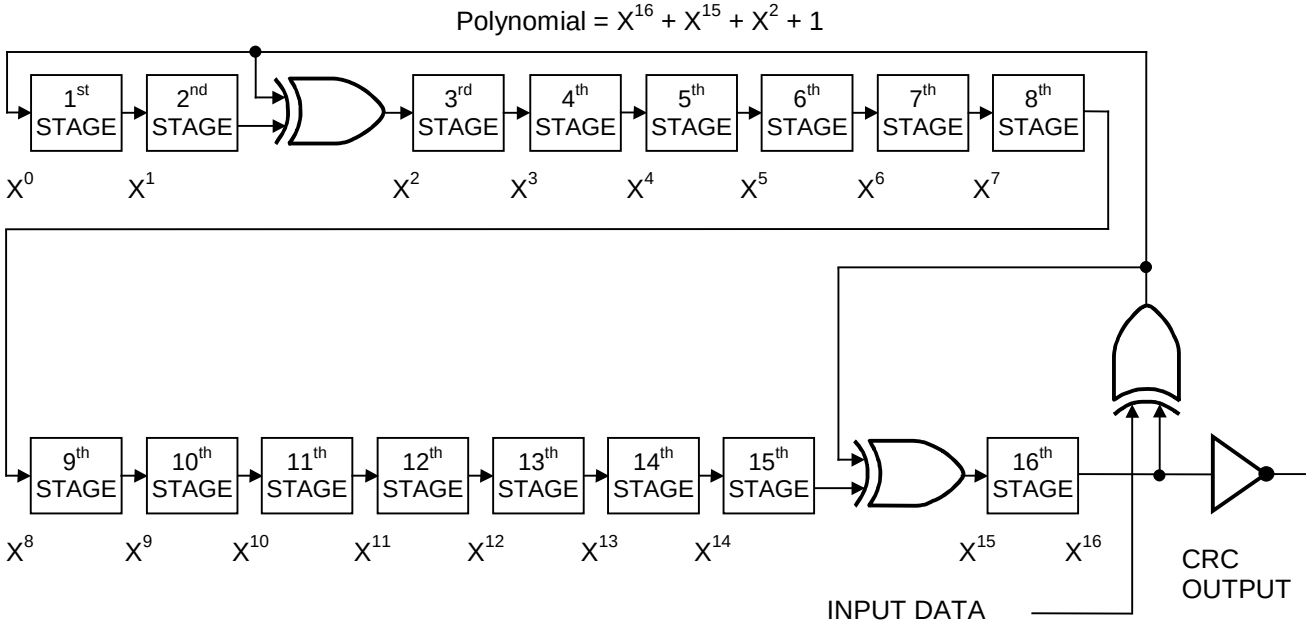
With the Write Scratchpad command the CRC is generated by first clearing the CRC generator and then shifting in the command code, the Target Addresses TA1 (with T2 to T0 set to 0) and TA2, and all data bytes as sent by the master. The DS2432 will transmit this CRC only if the scratchpad is filled to its capacity.

With the Read Scratchpad command the CRC is generated by first clearing the CRC generator and then shifting in the command code, the Target Addresses TA1 and TA2, the E/S byte, and the scratchpad data, which may have been modified by the DS2432 (see Write Scratchpad command). The DS2432 will transmit this CRC only if the reading continues through the end of the scratchpad.

With the Read Authenticated Page command the 16-bit CRC value is the result of shifting the command byte into the cleared CRC generator, followed by the two address bytes, the data bytes, and the FFh byte. The CRC that follows the Message Authentication Code (MAC) results from clearing the CRC generator and then shifting in the 160-bit MAC in the same bit sequence as the master receives it.

For more details on generating CRC values including example implementations in both hardware and software, see the "Book of DS19xx iButton Standards".

CRC-16 HARDWARE DESCRIPTION AND POLYNOMIAL Figure 12



ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground	-0.5V to +5.5V
Operating Temperature	-40°C to +85°C
Storage Temperature	-55°C to +125°C
Soldering Temperature	See J-STD-020A specification

* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

DC ELECTRICAL CHARACTERISTICS (V_{PUP} =2.8V to 5.25V; -40°C to +85°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
1-Wire Input High	V_{IH}	2.2			V	1, 7
1-Wire Input Low	V_{IL}	-0.3		TBD	V	1, 8
1-Wire Output Low @ 4 mA	V_{OL}			0.4	V	1
1-Wire Output High	V_{OH}		V_{PUP}		V	1, 2
Input Load Current	I_L		5		µA	3
Programming Current	I_{LPROG}		500		µA	9

CAPACITANCE (t_A = 25°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
1-Wire I/O	$C_{IN/OUT}$		100	800	pF	5

ENDURANCE (V_{PUP} =5.0V; T_A = 25°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Write/Erase Cycles	N_{CYCLE}	50k			—	
Data Retention	t_{DRET}	10			years	

AC ELECTRICAL CHARACTERISTICS REGULAR SPEED (V_{PUP}=2.8V to 5.25V; -40°C to +85°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Time Slot	t_{SLOT}	60		120	µs	
Write 1 Low Time	t_{LOW1}	1		15	µs	
Write 0 Low Time	t_{LOW0}	60		120	µs	
Read Low Time	t_{LOWR}	1		15	µs	
Read Data Valid	t_{RDV}		15		µs	10
Release Time	$t_{RELEASE}$	0	15	45	µs	
Read Data Setup	t_{SU}			1	µs	4
Recovery Time	t_{REC}	1			µs	
Reset High Time	t_{RSTH}	480			µs	
Reset Low Time	t_{RSTL}	480			µs	6
Presence Detect High	t_{PDHIGH}	15		60	µs	
Presence Detect Low	t_{PDLOW}	60		240	µs	
Programming Time	t_{PROG}			10	ms	
SHA Computation Time	t_{CSHA}		1.0	2.0	ms	9

AC ELECTRICAL CHARACTERISTICS

OVERDRIVE SPEED (V_{PUP}=2.8V to 5.25V; -40°C to +85°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Time Slot	t_{SLOT}	6		16	µs	
Write 1 Low Time	t_{LOW1}	1		2	µs	
Write 0 Low Time	t_{LOW0}	6		16	µs	
Read Low Time	t_{LOWR}	1		2	µs	
Read Data Valid	t_{RDV}		2		µs	10
Release Time	$t_{RELEASE}$	0	1.5	4	µs	
Read Data Setup	t_{SU}			1	µs	4
Recovery Time	t_{REC}	1			µs	
Reset High Time	t_{RSTH}	48			µs	
Reset Low Time	t_{RSTL}	48		80	µs	
Presence Detect High	t_{PDHIGH}	2		6	µs	
Presence Detect Low	t_{PDLOW}	8		24	µs	
Programming Time	t_{PROG}			10	ms	
SHA Computation Time	t_{CSHA}		1.0	2.0	ms	9

NOTES:

1. All voltages are referenced to ground.
2. V_{PUP} = external pull-up voltage.
3. Input load is to ground.
4. Read data setup time refers to the time the host must pull the 1-Wire bus low to read a bit. Data is guaranteed to be valid within 1 µs of this falling edge.
5. Capacitance on the data pin could be 800 pF when power is first applied. If a 5 kΩ resistor is used to pull up the data line to V_{PUP}, 5 µs after power has been applied the parasite capacitance will not affect normal communications.
6. The reset low time (t_{RSTL}) should be restricted to a maximum of 960 µs, to allow interrupt signaling, otherwise, it could mask or conceal interrupt pulses.
7. V_{IH} is a function of the external pull-up resistor and V_{PUP}.
8. Under certain low voltage conditions V_{ILMAX} may have to be reduced to as much as 0.5V to always guarantee a Presence Pulse. V_{IL} is a function of V_{PUP} and the reset low time.
9. During write operations to the EEPROM and during the computation of Message Authentication Codes (MAC) the voltage on the 1-Wire bus must not fall below 2.8V. The computation of a MAC takes maximum 2.0 ms. Copying scratchpad data to the EEPROM takes max. 10 ms.
10. The optimal sampling point for the master is as close as possible to the end time of the t_{RDV} period without exceeding t_{RDV}. For the case of a Read-one time slot, this maximizes the amount of time for the pull-up resistor to recover the line to a high level. For a Read-zero time slot it ensures that a read will occur before the fastest 1-Wire device releases the line ($t_{RELEASE}$ = 0).