DS1961S

带有 SHA-1 引擎保护的 1k位EEPROM iButton

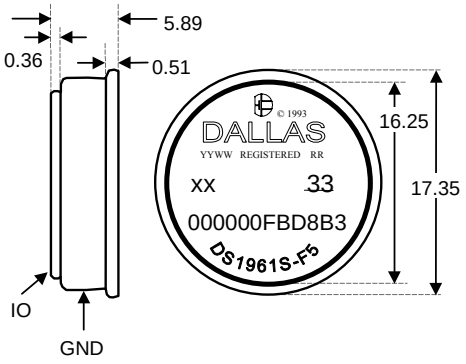
特性

- 1128 位 5V EEPROM 存储器，分为四页，每页 256 位，64 位只写密钥和多达五个通用读/写寄存器
- 写访问需要知道密钥，并且能够计算和传送 160 位MAC(信息认证码)，以便鉴别真伪
- 可以对密钥和数据存储器加写保护（所有页或者只是第 0 页），或者将它们置于 EPROM 仿真模式(“写入 0”，第 1 页)
- 内置 512 位 SHA-1 引擎，用于计算 160 位信息鉴定码（MAC）或生成密钥
- 读写操作可在 2.8V 至 5.25V 的很宽电压范围和-40ºC 至+85ºC 的温度范围内进行
- 使用 1-Wire® 协议通过一根数据线以 14.1kbps的速率与主机通信
- 内置 16 位循环冗余校验码（CRC）发生器，用于数据的安全传输
- 高速模式提高通信速率至 125kbps
- 工作温度范围-40°C 至+85°C
- +85°C 时，数据保存至少 10 年

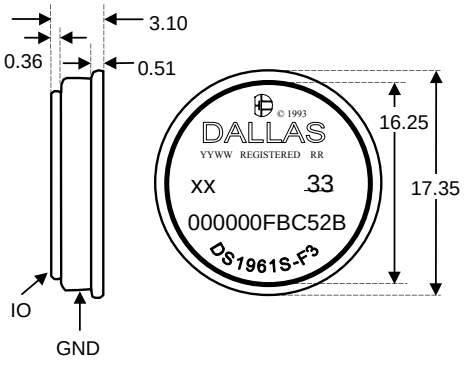
iButton共性

- 唯一的、由工厂光刻和测试的 64 位注册号（8 位家族码 + 48 位序列号 + 8 位 CRC 校验码）没有任何两个器件相同，保证绝对可溯
- 用于 1-Wire 网络的多节点控制器
- 短时间接触实现数字识别和信息获取
- 基于芯片的数据载体提供了一种紧凑的信息存储方案
- 可以安装在某一物体上并读取数据
- 钮扣外形可以自动对准杯状检测器
- 外刻注册号的不锈钢壳体能够经受住恶劣的环境
- 安装方便，可以使用自带粘性的背垫、固定其凸缘或利用压紧其边缘的圆环锁定
- 当读取器首次上电时，在线检测应答
- 符合 UL#913 (第四版)标准；本质安全设备：经过 I 级, 1 区, A、B、C 组和指定 D 区域场合的认证。

F5 MicroCan



F3 MicroCan



图中所有尺寸单位均为毫米。

iButton,1-Wire和MicroCan是 Dallas Semiconductor的注册商标。

定购信息

DS1961S-F5	F5 iButton
DS1961S-F3	F3 iButton

附件样例

DS1963S	SHA协处理器iButton
DS9096P	自粘胶垫
DS9101	多用途夹
DS9093RA	安装固定环
DS9093A	环扣
DS9092	iButton读取探头

iButton说明

DS1961S在坚固的iButton内集成了 1024 位EEPROM、64 位密钥、一个 8 字节的寄存器/控制页(其中包含五个用户读/写字节)、512 位SHA-1 引擎和一个全功能的 1-Wire接口。数据按照 1-Wire协议串行传送，只需一根数据线和返回地线。DS1961S有一个称为暂存器的辅助存储区，在向主存储器，寄存器写入数据时，或者在安装新密钥时充当缓冲器。数据首先被存入暂存器，并可从这里读回。经过验证后，假定DS1961S接收到了匹配的 160 位MAC，那么Copy Scratchpad（复制暂存器）命令将把数据传送到最终的存储单元。MAC的计算涉及到存储在DS1961S中(包含器件身份寄存器)的密钥和附加数据。只有加载新的密钥时才无需提供MAC。当读取存储页或是计算新密钥的时候，也可以激活SHA-1 引擎来计算 160 位的MAC，而不必加载它。

DS1961S 能够识别一个“Refresh Scratchpad”（更新暂存器）命令。在暂存器复制操作后正确地使用更新序列可以减少在接触条件下弱位失效的数目(见 *写入及验证*部分)。更新序列还提供了一种恢复弱状态位器件功能的方法。

每个 DS1961S 都有工厂光刻的、保证唯一的 64 位注册号，以确保实现产品跟踪。坚固耐用的不锈钢封装可防尘、防潮、防震。紧凑的硬币形状可自动对准配套的插座，这使 DS1961S 易于人工操作使用。各种附件使 DS1961S 可以安装在塑料钥匙链、图像身份卡等各种表面上。

应用

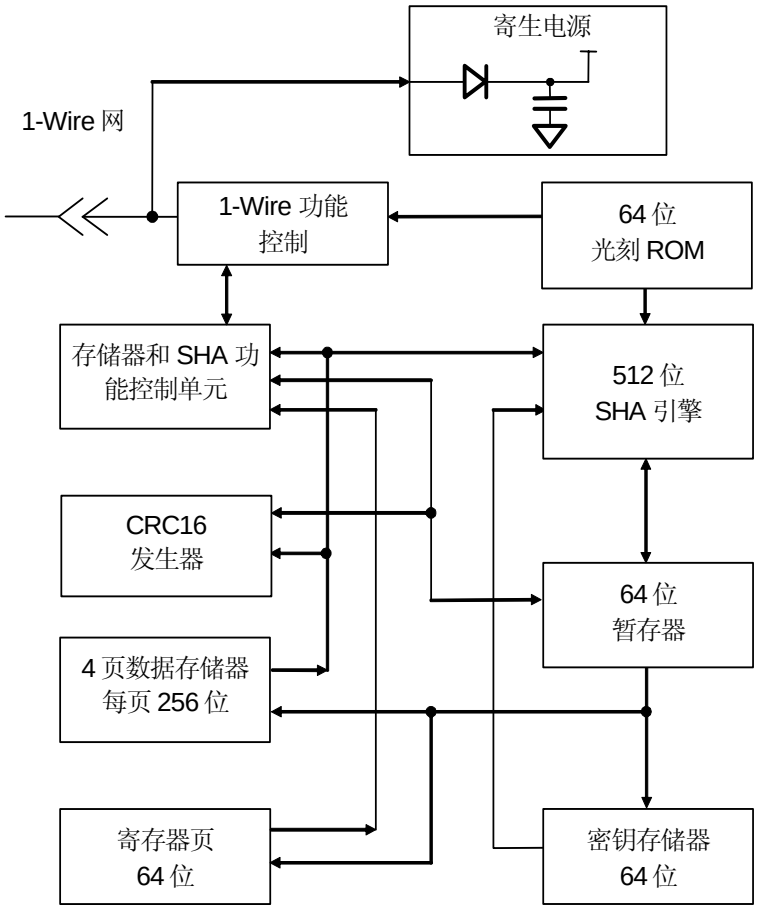
DS1961S 可有多种用途，例如安全访问控制，用户/产品认证，消费品售后服务管理，以及电子支付系统中的代用货币。作为电子货币（eCash）的载体，DS1961S 可存储同一个服务提供商的最多 3 个货币文件或“钱包”，这使得这一器件非常适合公司范围内的单密钥应用，如自助餐厅，复印机，游乐园或私人俱乐部的访问控制。当要实现更高级别的安全或主机微控制器处理能力不足时，可用 DS1963S 作协处理器校验 DS1961S 产生的 MAC 码或计算需要写入 DS1961S 的 MAC 码。

概述

图 1 中的框图说明了 DS1961S 的主控部分和存储单元之间的关系。DS1961S 有五个主要的数据部件：1）64 位光刻 ROM，2）64 位暂存器，3）四个 32 字节的 EEPROM 页，4）64 位寄存器页，5）64 位密钥存储器，6）一个 512 位 SHA-1（安全散列算法）引擎。1-Wire 协议分层结构见图 2。总线主机必须首先提供七个 ROM 操作命令中的一个：1）Read ROM, 2） Match ROM, 3）Search ROM, 4）Skip ROM, 5）Resume Communication, 6）Overdrive Skip ROM 或 7）Overdrive Match ROM。一旦以标准速率完成 Overdrive ROM 命令，器件就进入高速模式，随后的所有通信

都以高速进行。图 9 说明了协议所要求的这些 ROM 操作命令。成功地执行了 ROM 操作命令后，就可以进行存储器操作，主机可以发出八条存储器和 SHA 操作命令中的任何一个。图 7 说明了有关这些存储器和 SHA 操作命令的协议。所有数据读写都是 LSB 在前。

图 1. DS1961S 框图



64 位光刻 ROM

每个DS1961S都有一个 64 位的唯一ROM代码。前 8 位是 1-Wire家族代码。然后是 48 位的唯一序列号。最后 8 位是前 56 位的CRC检验码（图 3）。1-Wire CRC校验码由一个包含移位寄存器和异或门的多项式发生器产生，如图 4 所示。生成多项式为$X^8 + X^5 + X^4 + 1$。关于“Dallas 1-Wire CRC”的更多信息参见Dallas Semiconductor的*Book of DS19xx iButton Standards* 。移位寄存器初值为零。然后，从家族代码的LSB开始，每次移入一位。当家族代码第 8 位移入后，再移入序列号。当序列号第 48 位也移入后，留在移位寄存器中的就是CRC值。移入八位CRC校验码后，移位寄存器应该全部归零。

图 2. 1-Wire 协议的层次结构

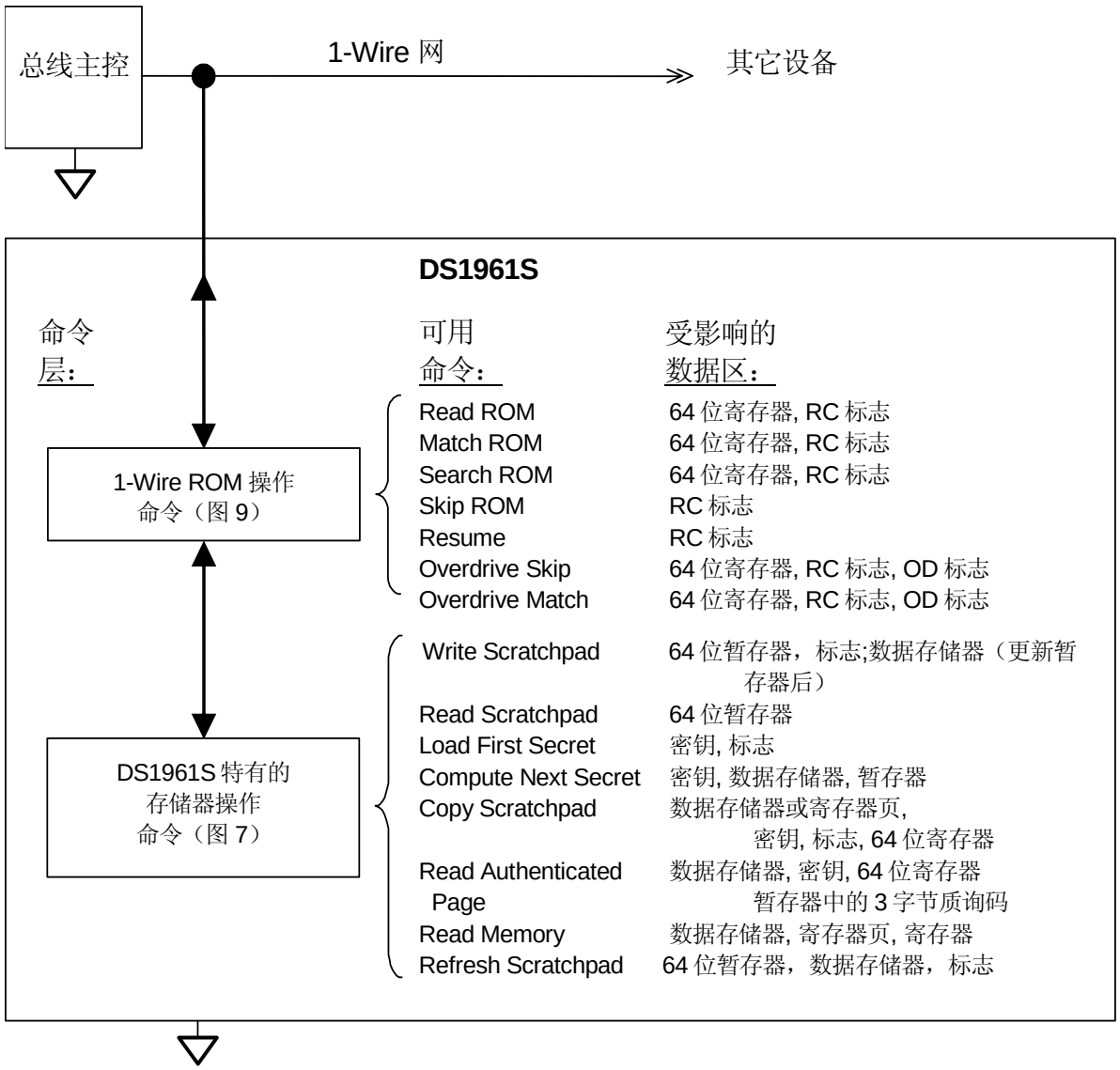
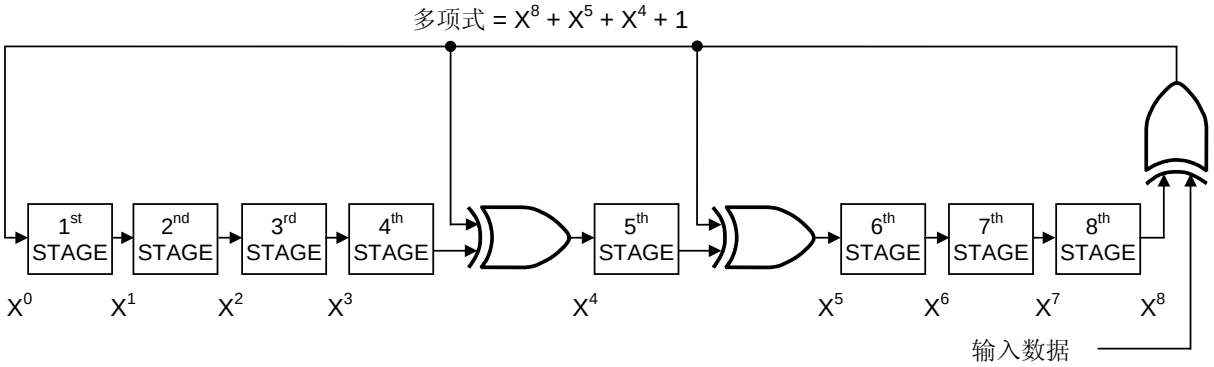


图 3. 64 位光刻 ROM

图 4. 1-Wire CRC 发生器



存储器映像

DS1961S 有四个存储区：数据存储器，密钥存储器，含有特定功能和用户字节的寄存器页和暂存器。数据存储器每页 32 个字节。密钥、寄存器页和暂存器均为 8 字节。向数据存储器写数据，装载初始密钥，或者向寄存器页写入数据时，暂存器作为缓存器使用。

如图 5 所示，数据存储器、密钥存储器和寄存器页位于一个线性地址空间中。数据存储器和寄存器页对读访问没有限制。但向数据存储器和寄存器页写数据则需要知道密钥。

图 5. DS1961S 存储器映像

地址	说明	注释
0000h 至 001Fh	数据存储器页 0	没有密钥不可写入
0020h 至 003Fh	数据存储器页 1	没有密钥不可写入
0040h 至 005Fh	数据存储器页 2	没有密钥不可写入
0060h 至 007Fh	数据存储器页 3	没有密钥不可写入
0080h 至 0087h	密钥存储器	不可读；写入无需密钥
0088h[1)]	写保护密钥；008Ch 至 008Fh	代码 AAh 或 55h 激活保护
0089h[1)]	写保护页 0 至 3	代码 AAh 或 55h 激活保护
008Ah[1)]	用户字节，自保护	代码 AAh 或 55h 激活保护
008Bh	工厂字节（只读）	读出为 AAh 或 55h；见内文
008Ch[1)]	用户字节/页 1 的 EPROM 模式控制	代码 AAh 或 55h 激活模式
008Dh[1)]	用户字节/仅写保护页 0	代码 AAh 或 55h 激活保护
008Eh 至 008Fh	用户字节/制造商 ID	功能取决于工厂字节
0090h 至 0097h	64 位身份寄存器	只读访问

[1)] 一旦编程为AAh或 55h，该地址就成为只读。可以存储所有其它的代码，但既不能对地址加写保护，也不激活任何功能。

密钥的安装有两种方法，一是把数据从暂存器复制到密钥存储器，二是通过当前密钥和暂存器内容经过运算后生成新的密钥。密钥不能直接读取；只有 SHA 引擎能够访问它，以计算信息鉴定码。

地址 0088h 至 008Fh，也被称为寄存器页，含有特定功能寄存器，通用用户字节，以及一个工厂字节。一旦编程为 AAh 或 55h，这些字节中的大多数将被写保护而不能再更改。其它所有代码既不能写保护，也不能激活与这个特定字节相关的特殊功能。特殊功能为：1）仅写保护密钥，2）同时写保护四个数据存储器页，3）仅激活数据存储器页 1 的 EPROM 模式，4）仅激活数据存储器页 0 的 EPROM 模式。一旦 EPROM 模式被激活，在数据存储器未加写保护的情况下，地址 0020h 至 003Fh 中的位只能从逻辑 1 改为逻辑 0。

工厂字节读取结果为 55h 或 AAh。通常这个地址读取到的是 55h，表明地址 008E 和 008F 是可读/写的用户字节，没有任何特定功能和锁定机制。代码 AAh 表明这两个字节被编程为 16 位制造商 ID，并在工厂内加了写保护。制造商 ID 是一个由用户提供的识别码，用来协助应用软件识别 DS1961S 所在的产品，以及快速找到可用的密钥。设置和注册制造商 ID 请与工厂联系。

地址 0090h 至 0097h 被称为身份寄存器，通常身份寄存器存有该器件 ROM 注册号的一个拷贝，家族代码存在较低地址，随后是 48 位的序列号和存储在地址 0097h 的 8 位 CRC 校验码。从这些地址（0090h 至 0097h）读取数据时，总线主机接收到的注册号每一位顺序都与使用 ROM 操作命令时相同。对用户定制的版本而言，身份寄存器内容可以是任何用户指定的常数。更多的用户定制信息请与工厂联系。

图 6. 地址寄存器

位号	7	6	5	4	3	2	1	0
目的地址 (TA1)	T7	T6	T5	T4	T3	T2 (0)	T1 (0)	T0 (0)
目的地址 (TA2)	T15	T14	T13	T12	T11	T10	T9	T8
结束地址及数据状态(E/S) (只读)	AA	1	PF	1	1	E2 (1)	E1 (1)	E0 (1)

地址寄存器和传输状态

DS1961S 使用三个地址寄存器：TA1，TA2 和 E/S（图 6）。这些寄存器普遍用于许多其它 1-Wire 器件，但在 DS1961S 中的工作略有不同。寄存器 TA1 和 TA2 装载写入数据的目的地址或读取数据的源地址。寄存器 E/S 是一个只读的传输状态寄存器，用于验证写命令的数据完整性。因为 DS1961S 的暂存器只接收 8 字节的数据块，所以 TA1 的低三位总为 0，E/S 寄存器（结束偏移量）的低三位总是 1。这意味着暂存器中的所有数据随后都要复制到主存储器或密钥中。E/S 寄存器的第 5 位称为 PF 或“字节不全标志（partial byte flag）”，该位如果为逻辑 1 则意味着主机发送的数据位数不是 8 的整数倍，或者暂存器中的数据由于掉电的关系而成为无效数据。有效的写

暂存器操作将清除 PF 位。第 3，4 和 6 位没有功能；读出时总为 1。利用 PF 标志，主机可以在写命令之后检验数据的完整性。E/S 寄存器的最高位称为 AA 或授权许可（authorization accepted），用以指示暂存器中的数据已复制到目的存储器地址。向暂存器中写入数据将清除该标志。

写入及验证

为了向 DS1961S 写入数据，需要把暂存器用作中间存储器。首先，主机发 Write Scratchpad（写暂存器）命令并指定目的地址和要写入暂存器的数据。需要注意的是，数据必须写入存储器的 8 字节边界内，也就是说，目的地址的三个最低有效位（T2—T0）必须等于 000b。如果发送的 T2—T0 为非零值，器件将把这些位强制置为零，并使用更改后的地址作为目的地址。发送八个字节的数据后，主机可以接收 Write Scratchpad 命令取反的 CRC16 校验码，该校验码计算时使用的地址和数据均为主机实际发送的值。主机将接收到的 CRC 与自己计算的结果进行比较来判断通信是否成功。暂存器数据被写入后，主机总是应该执行一次 Read Scratchpad 来验证写入数据是否正确。读暂存器时，DS1961S 会重新发回目的地址 TA1 和 TA2，以及 E/S 寄存器的内容。如果 DS1961S 在 Write Scratchpad 或 Refresh Scratchpad 命令中接收到的最后一个字节不完整，或上一次写暂存器后发生过掉电故障，则字节不全标志（partial flag）（E/S 寄存器的第 5 位）被置为 1。授权许可（AA）标志（E/S 寄存器的第 7 位）通常会被 Write Scratchpad 或 Refresh Scratchpad 命令清零；如果被置为 1，则说明 DS1961S 未能正确识别前一个 Write (或 Refresh) Scratchpad 命令。无论哪种情况，主机都应重写暂存器。收到 E/S 寄存器数据后，主机还会收到暂存器数据。Write Scratchpad 或 Refresh Scratchpad 的说明解释了暂存器数据在各种情况下可能发生的变化。在暂存器数据后面收到的是 Read Scratchpad 命令、目的地址、E/S 寄存器和暂存器数据取反的 CRC 校验码。与 Write Scratchpad 命令一样，此 CRC 被主机用来与自己计算的结果进行比较，以判断通信是否成功。完成数据验证后，主机可以发 Copy Scratchpad 命令将暂存器中的数据复制到存储器中。此外，还可以发送 Load First Secret 或 Compute Next Secret 命令来改变密钥。详细信息见相关命令的说明。

接触环境下并不能保证电接触的质量。如果电接触效果很差或发生间断，就有可能没有足够能量执行 Copy Scratchpad 命令，使 EEPROM 某一位的浮栅电压介于 0 和 1 之间。这种情况发生时，这一位的逻辑值是不确定的。取决于电压和/或温度条件。同一位可能被主机读为一种极性而被内部的 SHA-1 引擎读为相反的极性。主机不能正确计算 SHA-1 MAC，从而无法对该位进行重写，导致进入死锁模式。为了修复写入效果较差的位从而恢复器件正常功能，引入了一个 Refresh Scratchpad 命令。Refresh Scratchpad 与 Load First Secret 命令联合提供了一个将 EEPROM 各位恢复为正常值，解除死锁，允许器件被重新写入的途径。

为了防止写入效果较差的位，应该在每一个 Copy Scratchpad 命令后执行一次更新序列。更新序列定义为一个 Refresh Scratchpad（与前一个 Copy Scratchpad 命令目的地址相同），后面跟随一个 Load First Secret 命令。EN_LFS 标志被 Refresh Scratchpad 命令置位。EN_LFS 标志允许 Load First Secret 命令使用地址 0000h-007Fh。使用 Load First Secret 命令允许主机将暂存器数据复制到存储器中而不必执行 Copy Scratchpad 命令所需的 MAC 校验。如果主机在更新暂存器后试图执行任何有可能改变暂存器数据或目的地址的命令，EN_LFS 被复位为 0。从而保证 Load First Secret 命令仅能将更新的存储器数据装载到 Refresh Scratchpad 命令指定的地址。Refresh Scratchpad 命令行为

与 Write Scratchpad 命令对 0080h 及更高地址的操作相同。这种情况下 EN_LFS 并不置位，因此不能更新密钥（0080h）和寄存器页（0088h）中的数据。这能防止通过 Refresh Scratchpad 命令后跟随一个 Read Scratchpad 命令来获取密钥。

存储器和 SHA 操作命令

作为一个安全器件，DS1961S与其它iButton存储器使用稍有不同。DS1961S的大多数存储器可以像其它所有iButton存储器一样读取，但在尝试读取密钥时只能读到FFh字节，而不是真实数据。图 7 所示的*存储器和SHA功能流程*描述了访问存储器和操作SHA引擎的协议。主机与DS1961S之间的通信或者以标准速率（默认，OD ＝ 0），或者以高速模式（OD ＝ 1）进行。如果没有明确设定为高速模式，DS1961S默认为标准速率。

Write Scratchpad [0Fh]

Write Scratchpad（写暂存器）适用于数据存储器、密钥和寄存器页中的可写地址。如果总线主机发送的目的地址大于 90h，将不执行该命令。

发出 Write Scratchpad 命令后，主机必须首先提供 2 个字节的目的地址，随后是要写入暂存器的数据。数据将从暂存器的开头部分开始写入。值得注意的是，不论主机传送了多少个字节，结束偏移量（E2..E0,见图 6）的值总是 111b。由于这个原因，主机应该总是发送 8 个字节的数据，尤其是载入的数据被用作密钥时。如果主机发送的数据少于 8 个字节，并且也没有读回暂存器进行验证，那么新密钥的一部分可能是主机所不知道的随机数。只有完整的数据字节才能被接受。如果最后一个数据字节不完整，该字节将被忽略，并置位字节不全标志（PF）。

执行 Write Scratchpad 命令时，DS1961S 内部的 CRC 发生器（见图 12）随着主机的发送过程，计算整个数据流的 CRC 校验码，始于命令码，止于最后一个数据字节。该 CRC 校验码利用 CRC16 多项式产生，它首先清除 CRC 发生器，然后移入 Write Scratchpad 的命令代码（0Fh），接着是目的地址（TA1 和 TA2），以及所有的数据字节。要注意的是，尽管 DS1961S 在实际的 Write Scratchpad 命令中将设置 TA1 的位 T2..T0 为 000b，但是 CRC16 是根据主机发送的实际 TA1 来作计算的。主机可以随时终止 Write Scratchpad 命令。但是，如果暂存器已装满，主机可再发 16 个读时隙接收由 DS1961S 产生的 CRC 校验码。如果主机在读取 CRC 后继续读取数据，读到的所有数据均是 FFh。

收到目的地址（TA1 和 TA2）后，DS1961S 将清除 EN_LFS 标志。如果 EPROM 模式被激活，并且试图在页 1（0020h-003Fh）执行 Write Scratchpad 命令，暂存器实际装入的数据将是主机发送的暂存器数据与目的地址现存数据进行逻辑与的结果。如果试图在寄存器页（0088h-008Fh）执行 Write Scratchpad 命令，写保护字节现存的值将覆盖主机发送的相应暂存器数据。其它所有情况下，主机发送的所有数据都会无改变的写入暂存器。

Read Scratchpad [AAh]

Read Scratchpad（读暂存器）可以用来验证目的地址和暂存器数据的完整性。发出命令码后，主机开始读数据。开头的两个字节是目的地址，其中 T2 至 T0 = 0。下一个字节是结束偏移量/数据状态字节（E/S），跟在后面的便是暂存器数据，它可能与主机最初发送的数据不同，尤其是当目的地址为密钥存储器，寄存器页，存储器页 1（处于 EPROM 模式）时，或使用 Refresh Scratchpad 时。这些情况下，暂存器数据有可能与 Write Scratchpad 或 Refresh Scratchpad 命令实际发送的数据不同。主机应该读到暂存器的最后一个字节，随后，就可以收到反码的 CRC。它基于 DS1961S 所发送的数据计算产生。如果主机读取 CRC 校验码后继续读，那么读到的所有数据都将是 FFh。

暂存器数据可由 Write Scratchpad 或 Refresh Scratchpad 装入。暂存器中实际存在的数据取决于使用的命令，目的地址，以及是否激活了 EPROM 模式。有关解释见 Write Scratchpad 或 Refresh Scratchpad 命令的说明。

Load First Secret [5Ah]

Load First Secret（首次装载密钥）命令有两种受 EN_LFS 标志控制的运行模式。EN_LFS=0 时，如果密钥未加写保护，该命令用暂存器数据替换器件现有的密钥。EN_LFS=1 时，该命令允许重写存储器数据（地址 0000h 至 007Fh），而不经过 Copy Scratchpad 命令所需的 SHA-1 计算。在 Load First Secret 之前执行 Refresh Scratchpad 命令可将 EN_LFS 置为 1，否则 EN_LFS 一直为 0。

EN_LFS = 0 时

此模式下，在执行 Load First Secret 命令前，主机必须使用密钥起始地址（0080h）将新的密钥写入暂存器。发出 Load First Secret 后，主机必须提供一个 3 字节的授权码（依次为 TA1，TA2，E/S），此数据应该通过紧邻此条命令的前一个 Read Scratchpad 命令获得。这 3 个字节的数据必须与三个地址寄存器中的数据完全匹配（见图 6）。如果数据匹配，而且密钥未加写保护，AA 标志将置位，并开始复制数据。暂存器内容的所有 8 个字节的数据都将被复制到密钥存储单元。

EN_LFS = 1 时

此模式下执行 Load First Secret 命令，必须先执行 Refresh Scratchpad 命令，将存储器中 8 个字节数据装入暂存器（地址为从 0000h 至 007Fh），同时将 EN_LFS 置为 1。发出 Load First Secret 后，主机必须提供一个 3 字节的授权码（依次为 TA1，TA2，E/S），此数据应该通过紧邻此条命令的前一个不影响 EN_LFS 值的 Read Scratchpad 命令获得。这 3 个字节的模式数据必须与三个地址寄存器中的数据完全匹配（见图 6）。如果模式匹配，而且存储器未加写保护，AA 标志将置位，并开始复制数据。暂存器内容的所有 8 个字节的数据都将被复制到存储单元。

主机必须使用密钥起始地址（0080h）将新的密钥写入暂存器。发出 Load First Secret 后，主机必须提供一个 3 字节的授权码（依次为 TA1，TA2，E/S），此数据应该通过紧邻此条命令的前一个 Read Scratchpad 命令获得。这 3 个字节的数据必须与三个地址寄存器中的数据完全匹配（见图 6）。如果数据匹配，而且密钥未加写保护，AA 标志将置位，并开始复制数据。暂存器所有 8 个字节的数据都将被复制到密钥存储单元。

无论使用那种模式，复制操作时间为t_{PROG}，期间 1-Wire总线上的电平一定不要低于 2.8V。主机在复制延时结束后应至少读取 1 个字节。读到AAh说明复制成功，读到FFh说明复制不成功。除了在EN_LFS=0 时使用Load First Secret命令，新密钥还可以通过Copy Scratchpad命令装载。不过，这个方法需要知道当前密钥，并要计算 160 位的MAC。

图 7-1. 存储器和 SHA 功能流程图

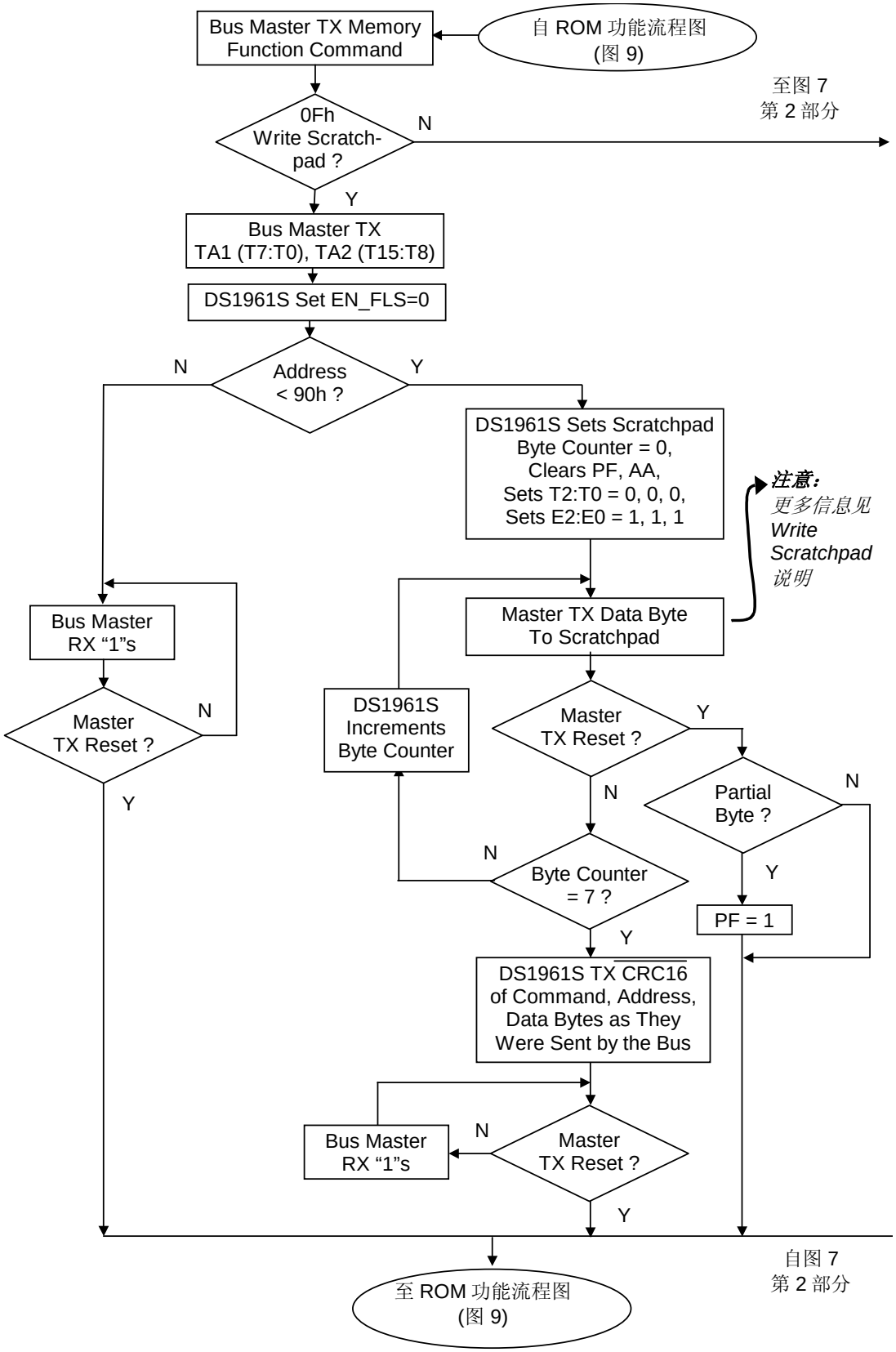


图 7-2. 存储器和 SHA 功能流程图（续）

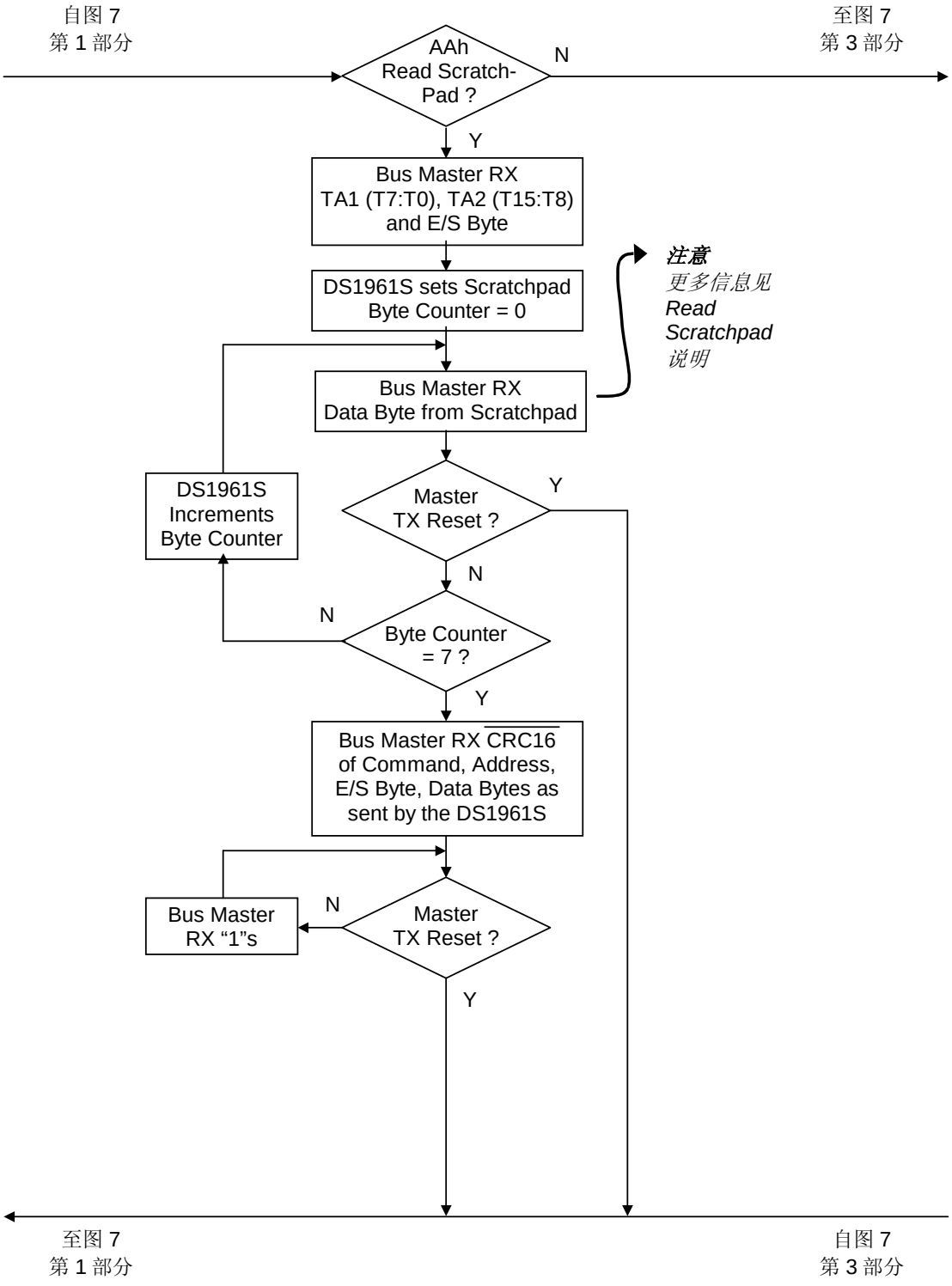


图 7-3. 存储器和 SHA 功能流程图（续）

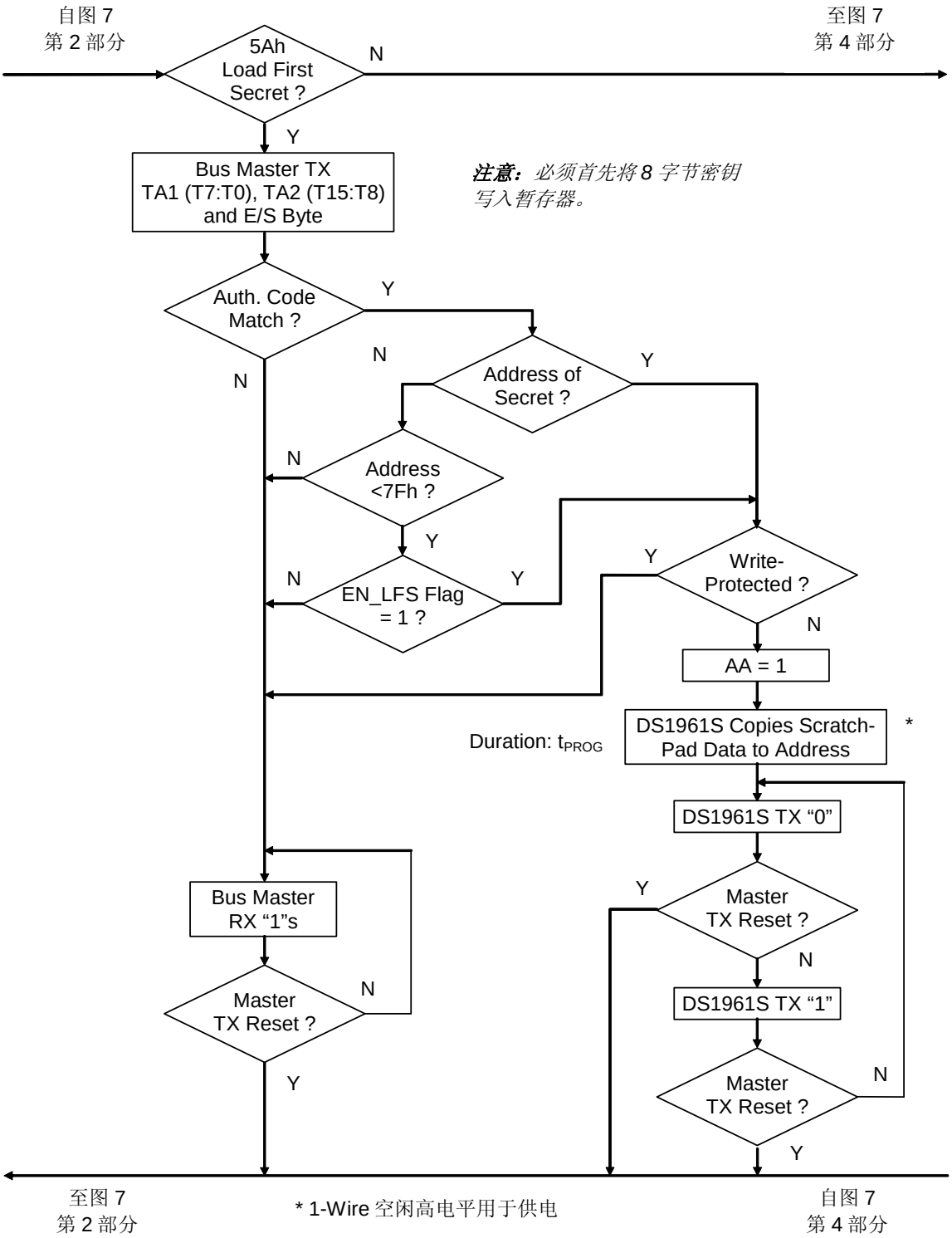


图 7-4. 存储器和 SHA 功能流程图（续）

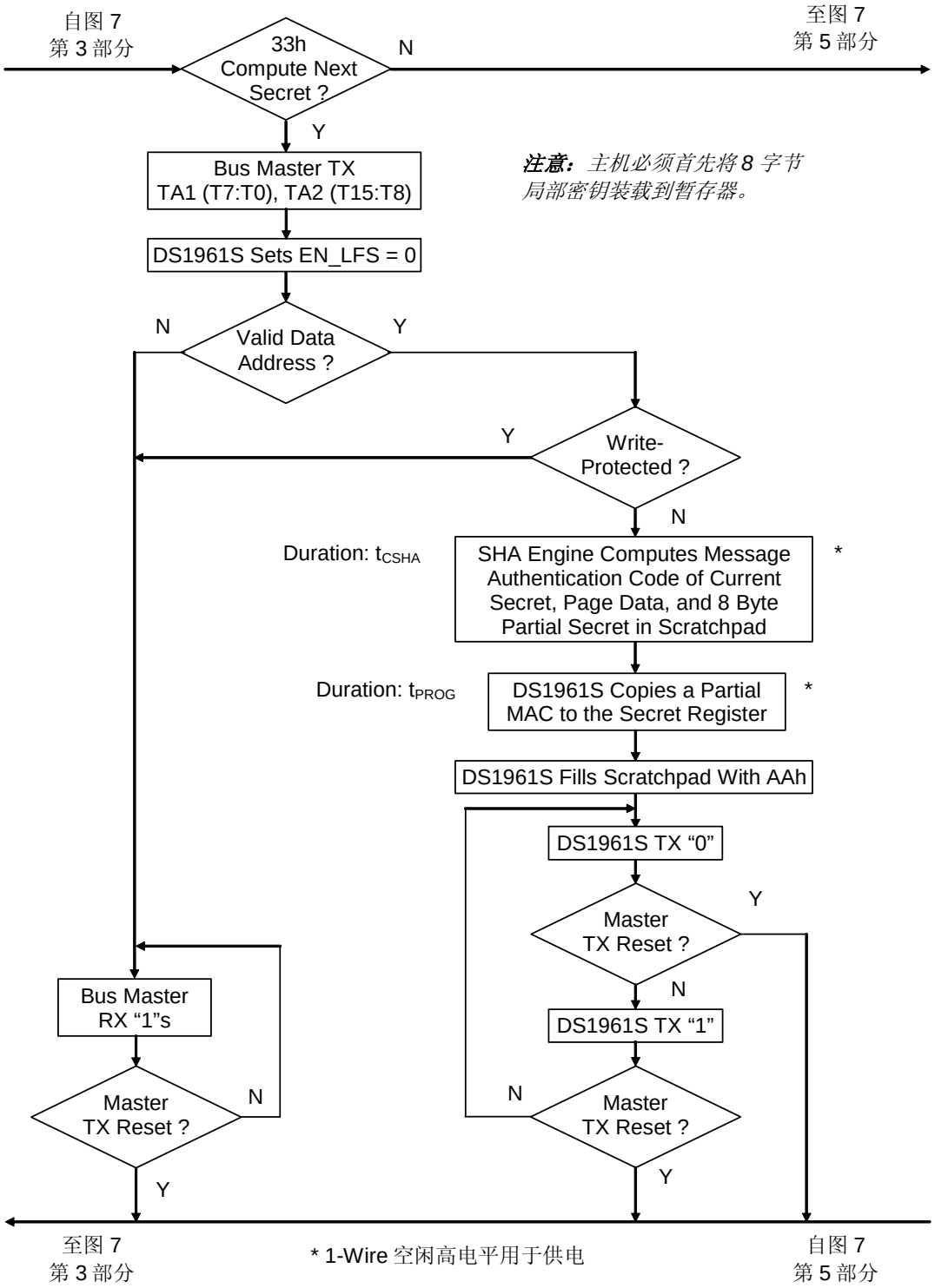


图 7-5. 存储器和 SHA 功能流程图（续）

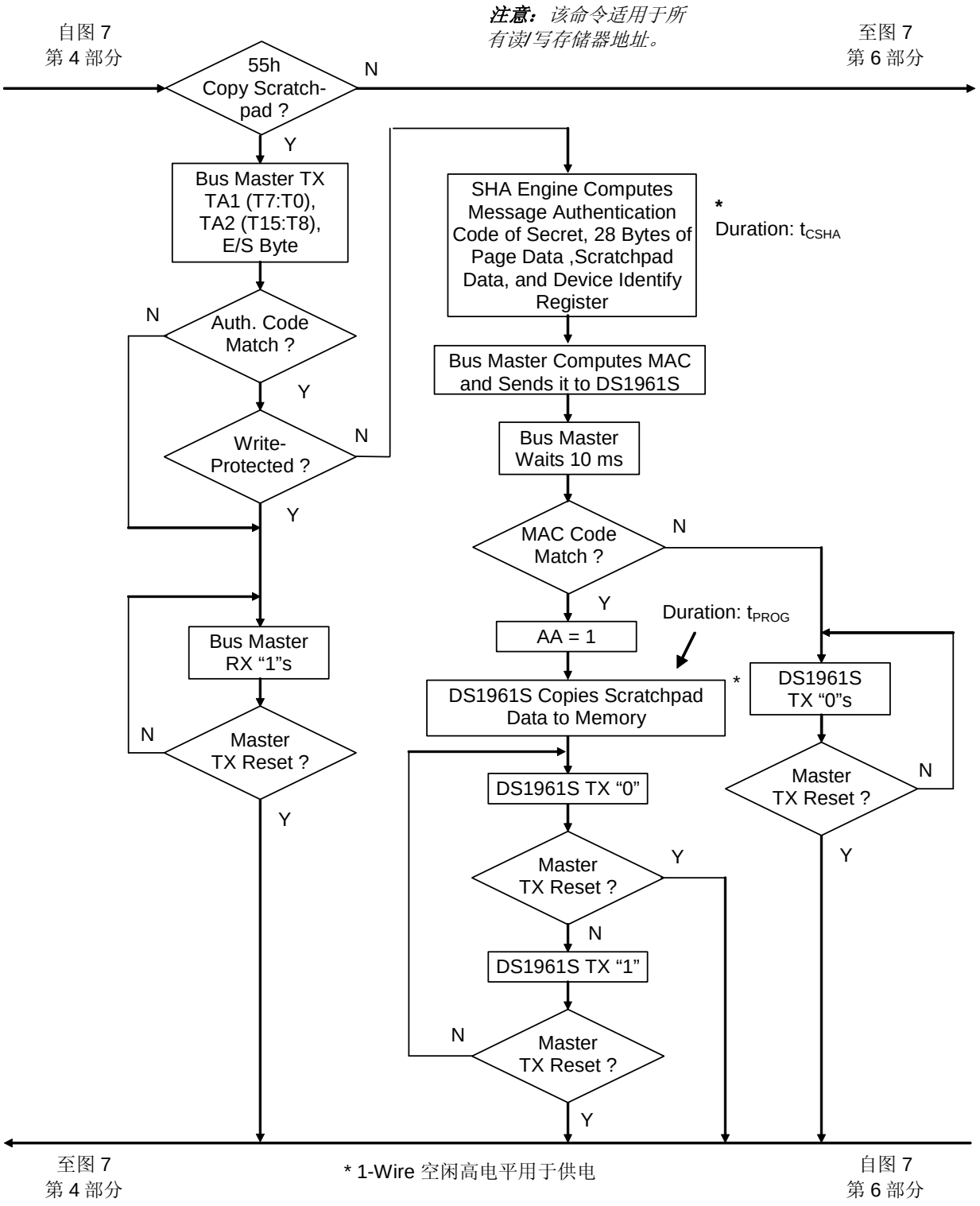


图 7-6. 存储器和 SHA 功能流程图（续）

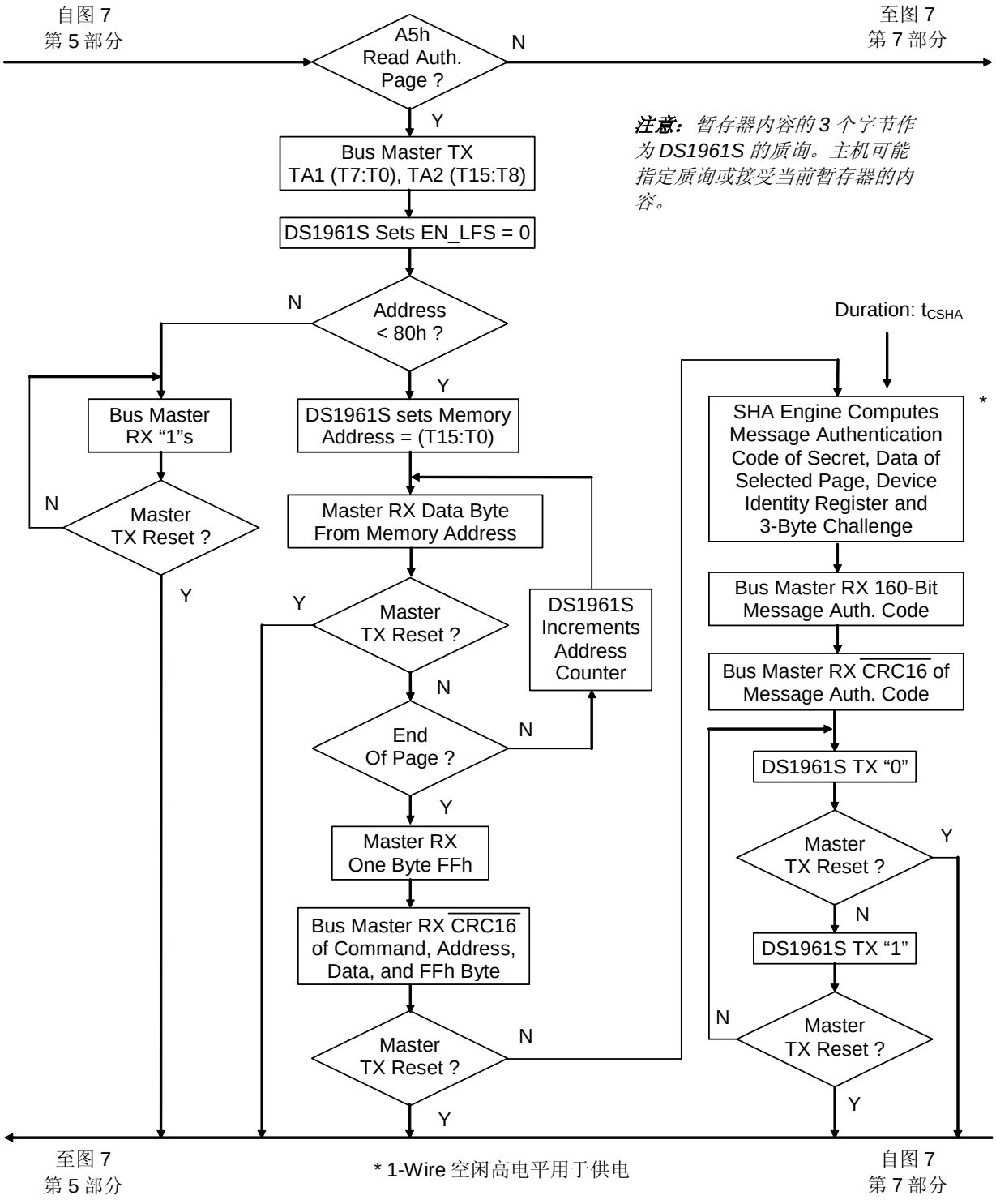


图 7-7. 存储器和 SHA 功能流程图（续）

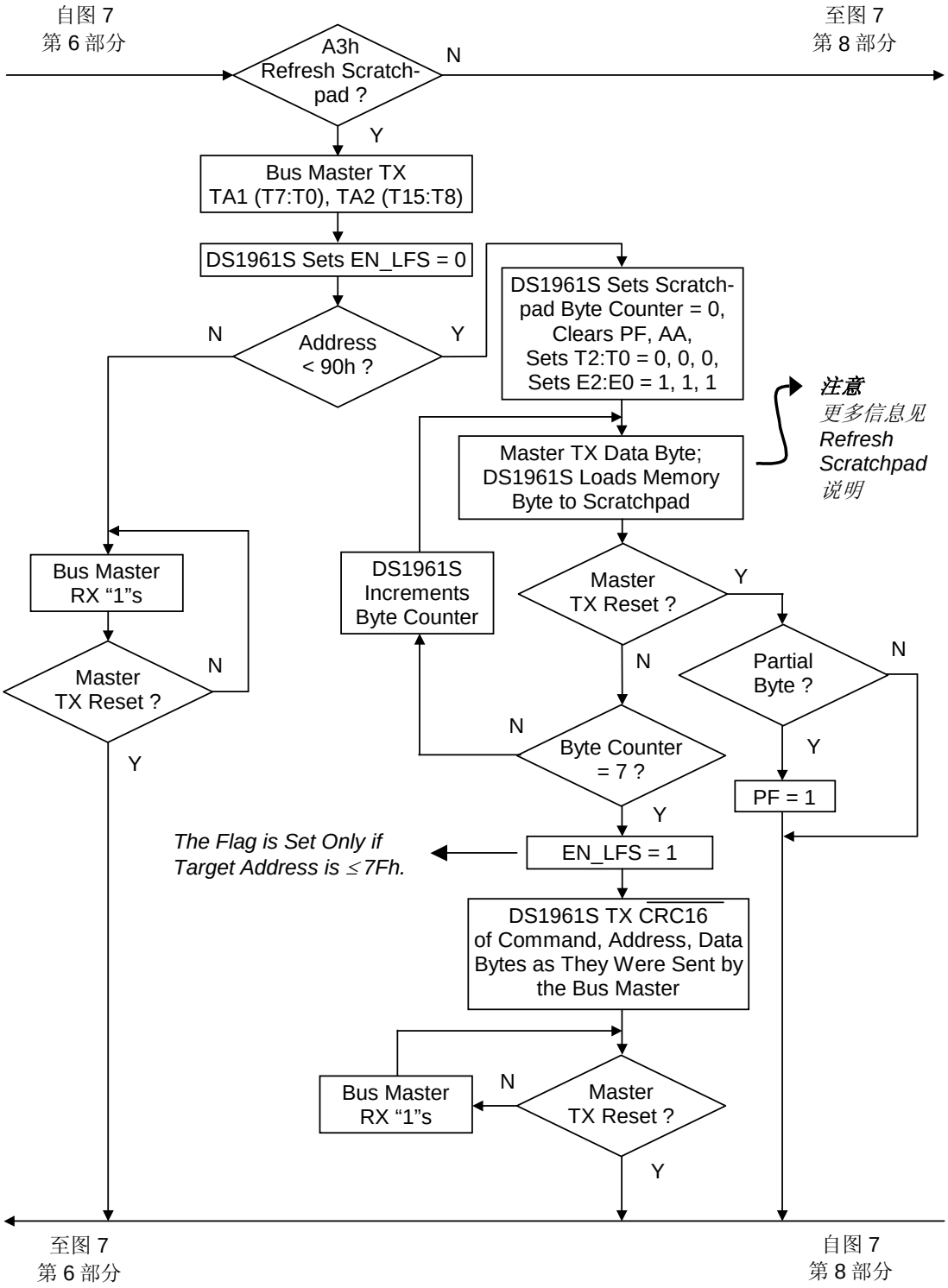
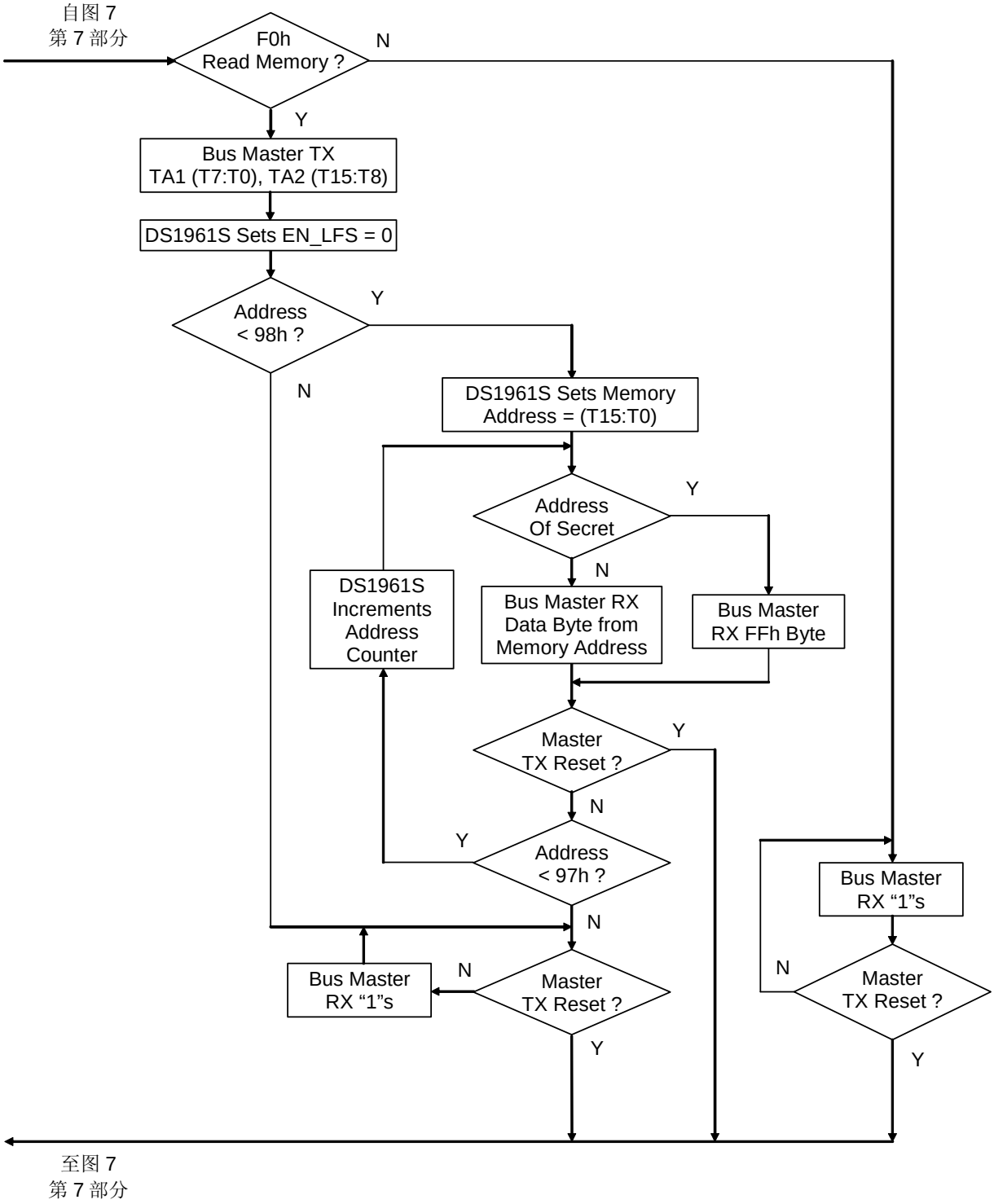


图 7-8. 存储器和 SHA 功能流程图（续）



Compute Next Secret [33h]

一些应用对安全性的要求要比利用单一的、直接写入密钥所能达到的安全水平要高。为增加安全性，DS1961S 能够基于当前密钥、一个指定的存储器页的内容、以及暂存器中所有数据组成的部分密钥计算出一个新的密钥。在密钥未被写保护的情况下，要安装计算出来的密钥，主机需要发 Compute Next Secret（计算下一密钥）命令，这条命令将激活 512 位 SHA-1 引擎。表 1 说明了有关的各种数据是如何进入 SHA 引擎的，以及 SHA 结果的一部分是如何载入密钥存储单元的。稍后，本文将介绍 SHA 的算法。Compute Next Secret 命令可以根据需要多次使用，以便提高安全性水平。总线主机不必知道器件的当前密钥，就可以成功计算出一个新密钥，并用它覆盖现存的密钥。

表 1. Compute Next Secret 命令所需的 SHA-1 输入数据

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

Compute Next Secret 结果

(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]

符号说明

Mt	**SHA 引擎的输入缓冲器** 0 ≤ t ≤ 15；32 位字
(SS + N)	**密钥的第 N 字节，密钥起始地址为 0080h** （见*存储器映像*）
(PP + N)	**存储器页的第 N 字节；存储器页起始于 0000h，0020h，0040h 和 0060h**（见*存储器映像*）
(SP + N)	**暂存器第 N 字节**
MPX	MPX[7] = 0; MPX[6] = 0; MPX[5:0] = (SP + 0)[5:0]
D, E	32 位字，160 位 SHA 结果的一部分

发出Compute Next Secret命令后，主机必须提供一个 2 字节的目的地址，用于指定提供 256 位 SHA输入数据的存储器页。收到目的地址（TA1 和TA2）后，DS1961S将清除EN_LFS标志。由于

仅页号有效，所以目的地址TA1 低五位将被忽略。如果主机送出的目的地址有效（如在 0000h至007Fh范围内），而且密钥未加写保护，SHA引擎将启动。主机必须等待t_{CSHA}以计算出一个新的密钥，在SHA延时之后，主机还要等待t_{PROG}以将新密钥复制到密钥寄存器中。在t_{CSHA}和t_{PROG}这段时间内，1-Wire总线上的电平一定不能低于 2.8V。复制完成后，DS1961S用AAh字节填充暂存器，但如果SHA引擎由于地址不正确或由于写保护没有启动，将不会修改暂存器的值。复制延时结束后主机应至少读取一个字节。如果读到AAh，说明复制成功，读到FFh则说明由于地址不正确或由于写保护导致复制失败。

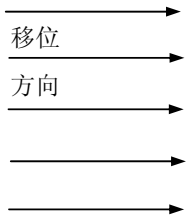
由于暂存器的内容被用做部分密钥，因此，暂存器必须在发 Compute Next Secret 命令之前，用 Write Scratchpad 命令给暂存器写入 8 字节已知数据。否则的话，新密钥将取决于以前的操作留在暂存器中的数据。

Copy Scratchpad [55h]

DS1961S的数据存储器可以随意读取。然而，执行Copy Scratchpad（复制暂存器）要向存储器或寄存器页写入新的数据，就需要知道器件的密钥，并且能够执行SHA-1 运算，以产生 160 位的MAC，这样才可启动由暂存器到存储器的数据传送过程。主机可以在软件中计算MAC，或者用DS1963S作为协处理器。协处理器的好处是密钥可以隐藏在协处理器iButton中。向DS1961S发送MAC运算结果的顺序如表 2 所示。表 3A和 3B说明了各种数据元素是如何进入SHA引擎的。有关SHA算法的说明，参见本文档的后续部分。

表 2. 信息鉴定码传送顺序

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]



传送始于寄存器E，最低有效位优先。

发出Copy Scratchpad命令后，主机必须提供一个 3 字节的授权码，这个数据应该通过紧邻此条命令的前一个Read Scratchpad命令获得。这 3 个字节的数据必须与三个地址寄存器（依次为TA1，TA2，E/S）中的数据完全匹配。如果数据匹配，而且目标存储器未加写保护，DS1961S将启动它的SHA引擎，基于当前密钥、暂存器中的所有数据、所寻址的存储器页的前 28 个字节数据、以及身份寄存器的前 7 个字节（地址 0097h处的字节未被使用，见表 3A）计算一个 160 位的MAC。计算所需时间为t_{CSHA}，期间 1-Wire总线上的电平一定不能低于 2.8V。同时，主机也利用同样的数据计算一个MAC，待t_{CSHA}结束后，把它发送给DS1961S，以便证明它有权写EEPROM。然后，主机需要等待t_{PROG}，在此期间，1-Wire总线上的电平一定不能低于 2.8V。如果DS1961S生成的MAC与主机计算的MAC相匹配，DS1961S将置位AA标志，并将整个暂存器的内容复制到数据EEPROM。复制延时结束后主机应至少读取一个字节。如果读到AAh说明复制成功。读到 00h说明由于计算出的MAC与主机发送的MAC不匹配导致复制失败。读到FFh则说明由于写保护或错误的授权码导致复制失败。

表 3a. 复制数据到数据存储器页时 Copy Scratchpad 命令的 SHA-1 输入数据

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

符号说明

Mt	**SHA 引擎的输入缓冲器** 0 ≤ t ≤ 15；32 位字
(SS + N)	**密钥的第 N 字节；密钥的起始地址为 0080h** （见*存储器映像*）
(PP + N)	**存储器页的第 N 字节；存储器页起始于 0000h，0020h，0040h 和 0060h**（见*存储器映像*）
(SP + N)	**暂存器第 N 字节**
MP	MP[7:3] = 00000b， MP[2:0] = T7:T5
(ID + N)	**身份寄存器第 N 个字节** 器件身份寄存器最后一个字节未被使用

在复制数据到寄存器页的时候需要特别小心。为了防止无意中锁定某个特定功能寄存器或用户字节，建议首先读取寄存器页，然后在暂存器中修改后再全部写回。在向寄存器页复制数据（或通过 Copy Scratchpad 命令建立密钥）时，SHA 引擎的 M1 至 M7 输入数据将是当前密钥（M1，M2），寄存器页的当前内容（M3，M4），身份寄存器的全部内容（M5，M6），和 4 个字节 FFh（M7），见表 3B 所示。因此，如果使用 DS1963S 作为由暂存器向寄存器页传输数据时计算 MAC 的协处理器，密钥必须被当作页数据。如果需要写寄存器页，这会禁止使用部分（计算的）密钥。在使用 DS1961S 作为代用货币的实际应用中，部分密钥比对密钥或器件其他部分进行写保护更重要。

表 3b. 复制数据到寄存器页或密钥时 Copy Scratchpad 命令的 SHA-1 输入数据

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] = (PP + 6)	M2[7:0] = (PP + 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

符号说明

Mt	**SHA 引擎的输入缓冲器** 0 ≤ t ≤ 15；32 位字
(SS + N)	**密钥的第 N 字节；密钥的起始地址为 0080h** （见*存储器映像*）
(RP + N)	**寄存器页的第 N 字节；页起始于 0088h** （见*存储器映像*）
(SP + N)	**暂存器第 N 字节**
MP	MP[7:0] = 04h
(ID + N)	**身份寄存器第 N 个字节**

Read Authenticated Page [A5h]

利用命令 Read Authenticated Page（读验证页），主机可以获得全部或部分存储器页的数据和一个 MAC。通过 MAC，主机能够判定存储在 DS1961S 中的密钥是否对于某特定应用有效。DS1961S 利用自己的密钥、指定存储器页的所有数据、身份寄存器的前 7 个字节和一个 3 字节的质询来计算 MAC，这个 3 字节质询是由主机在发 Read Authenticated Page 命令之前提前写入暂存器的。为此，主机可以使用 Write Scratchpad 命令，采用数据存储器内的任意目的地址，将质询写入暂存器。有关质询的部分为第 5，第 6 和第 7 个字节。作为另外一种选择，主机也可以将执行前一命令时，偶然留在暂存器中的数据作为一个质询。160 位 MAC 的传送方法与 Copy Scratchpad 命令中的情况完全一样，见表 2，只是数据流向改为由 DS1961S 至主机。执行 Read Authenticated Page 命令时输入 SHA 引擎的数据见表 4 所示。

主机发出命令代码并指定了目的地址（TA1 和TA2）后，DS1961S首先清除EN_LFS标志。如果目的地址有效（< 0080h）,主机将收到从目的地址开始，一直到数据页末尾的存储器页数据、一个FFh字节和一个反码的CRC，该CRC码由命令代码、目的地址、已传送的数据和FFh字节产生。如果目的地址无效（≥ 0080h），主机将接收到FFh而不是页数据。CRC校验码接收完毕后，主机等待t_{CSHA}，在此期间，1-Wire总线上的电平不能低于 2.8V。在这段时间内，DS1961S的SHA引擎利用密钥、选定页的 32 个数据字节、器件的注册号（不包括CRC校验码）和 3 字节质询计算MAC。然后，主机就可读取 160 位MAC，随后是一个反码的CRC，以确保数据传输的可靠性。如果在CRC校验码后主机继续读取数据，它将收到AAh。

表 4. Read Authenticated Page 命令的 SHA-1 输入数据

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

符号说明

Mt	**SHA 引擎的输入缓冲器** 0 ≤ t ≤ 15；32 位字
(SS + N)	**密钥的第 N 字节；密钥的起始地址为 0080h** （见*存储器映像*）
(PP + N)	**存储器页的第 N 字节；存储器页起始于 0000h，0020h，0040h 和 0060h**（见*存储器映像*）
(SP + N)	**暂存器第 N 字节**
MP	MP[7:3] = 01000b， MP[2:0] = T7:T5
(ID + N)	**身份寄存器第 N 个字节** 器件身份寄存器最后一个字节未被使用

Refresh Scratchpad [A3h]

Refresh Scratchpad (更新暂存器)将存贮器中的数据装入暂存器并将 EN_LFS 标志置位，从而允许使用 Load First Secret 命令重写从存储器中读出的数据，而不经过复制暂存器的 MAC 计算。

Refresh Scratchpad 的命令流程图与 Write Scratchpad 非常相似。但如果目的地址位于 0000h-007Fh 之间，会有两个主要差异。1）主机发送的位于目的地址后面的数据字节将被忽略；暂存器将装入位于存储器目的地址上的原始数据，即使存储器页处于 EPROM 模式。2）主机发送八个空字节后，EN_LFS 被置 1。在 Write Scratchpad, Compute Next Secret, Read Authenticated Page, Refresh Scratch, Read Memory 命令或上电复位收到 TA1 和 TA2 后 EN_LFS 标志清 0，因为这些命令可能改变目标地址和/或暂存器中的数据。

在地址 0080h-008Fh 处，Refresh Scratchpad 命令与 Write Scratchpad 命令操作相同。以保护密钥不会被随后的 Read Scratchpad 命令获取。

Read Memory [F0h]

Read Memory（读存储器）可以用来读取除密钥之外的所有存储器。尝试读取密钥时将会获得 FFh 而不是真实密钥。主机发出命令代码和目的地址（TA1 和 TA2）后，DS1961S 会首先清除 EN_LFS 标志。如果目的地址有效，主机读取从目的地址开始的数据，可以一直读到地址 0097h。如果继续读，结果将全是逻辑 1。应该注意的是，目的地址寄存器将指向最后一个读取的字节。结束偏移量/数据状态字节和暂存器不受影响。

DS1961S 提供的硬件手段能够保证写入存储单元的数据正确无误。为了保证在 1-Wire 环境下读取数据的可靠性，同时提高数据传输的速率，建议将数据按照存储器页的大小进行分组。然后，在每个分组内包含一个由主机计算的、针对每页数据的 16 位 CRC 校验码。这样，主机就不必多次重复地读取一页数据来检验数据的正确与否，从而保证了快速、无误地传输数据（推荐的文件结构参见*应用笔记 114*，有时也称之为 TMEX 格式）。

SHA-1 算法

以下有关SHA算法的说明译自安全散列标准（Secure Hash Standard）SHA-1 文档，该文档可从NIST网站下载（www.itl.nist.gov/fipspubs/fip180-1.htm）。该算法采用十六个 32 位字M_t（$0 \le t \le 15$）作为输入数据，如表 1，3A和 3B所示，分别被用于Compute Next Secret，Copy Scratchpad和Read Authenticated Page命令。SHA算法涉及到一个称为W_t（$0 \le t \le 79$）的八十个 32 位字的序列，一个称为K_t（$0 \le t \le 79$）的八十个 32 位字的序列，一个布尔函数f_t（B，C，D）（$0 \le t \le 79$），其中B，C和D为 32 位字，以及另外三个 32 位字，称为A，E和TMP。SHA算法用到的操作有不带进位的算术加（“+”），逻辑反或 1 的补码（“\”），异或（“⊕”），逻辑与（“∧”），逻辑或（“∨”），赋值（“:=”），以及 32 位字的循环移位。表达式“$S^n(X)$”表示将X向左循环移n位，X是一个 32 位字。

函数f_t定义如下：

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

序列W_t（$0 \le t \le 79$）定义如下：

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

序列K_t（$0 \le t \le 79$）定义如下：

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

变量 A，B，C，D，E 初始化如下：

$$
\begin{array}{lll}
A & := & 67452301h \\
B & := & EFCDAB89h \\
C & := & 98BADCFEh \\
D & := & 10325476h \\
E & := & C3D2E1F0h
\end{array}
$$

当 t 从 0 循环至 79，执行了下面的一系列计算后，160 位 MAC 是 A，B，C，D 和 E 的串联（不考虑任何进位）：

$$
\begin{array}{lll}
TMP & := & S^5(A) + f_t(B,C,D) + W_t + K_t + E \\
E & := & D \\
D & := & C \\
C & := & S^{30}(B) \\
B & := & A \\
A & := & TMP
\end{array}
$$

主机可以按照表 3 所示的寄存器和位顺序，通过 Read Authenticated Page 命令读取 MAC。与 Copy Scratchpad 命令相比，位的传送顺序是一样的,不过，主机必须计算 MAC，并将其发送给 DS1961S。在执行 Compute Next Secret 命令时 MAC 不会暴露。SHA 运算寄存器 D 和 E 的内容被直接复制到密钥寄存器，如表 1 所示。

1-Wire 总线系统

1-Wire 单总线系统是用一根数据线连接单个主机和一台或多台从机设备的系统。任何情况下，DS1961S 都作为从机设备来使用。总线上的主机典型为微处理器。在小型系统中，可利用软件控制一个单独端口引脚产生 1-Wire 通信信号。在大一些的系统中，推荐使用 DS2480B 1-Wire 线驱动器芯片或基于这一芯片的串口适配器（DS9097U 系列）。这能简化硬件设计并将微控制器从对实时信号的响应中解脱出来。

对单总线系统的论述分为以下三个部分：硬件结构、处理流程和 1-Wire信令（信号类型和时序）。1-Wire通信协议规定总线的收发按照特殊时隙下的总线状态进行，由主机发出的同步脉冲下降沿初始化。需要了解更多关于通讯协议详细描述，请参考*Book of DS19xx iButton Standards*第 4 章。

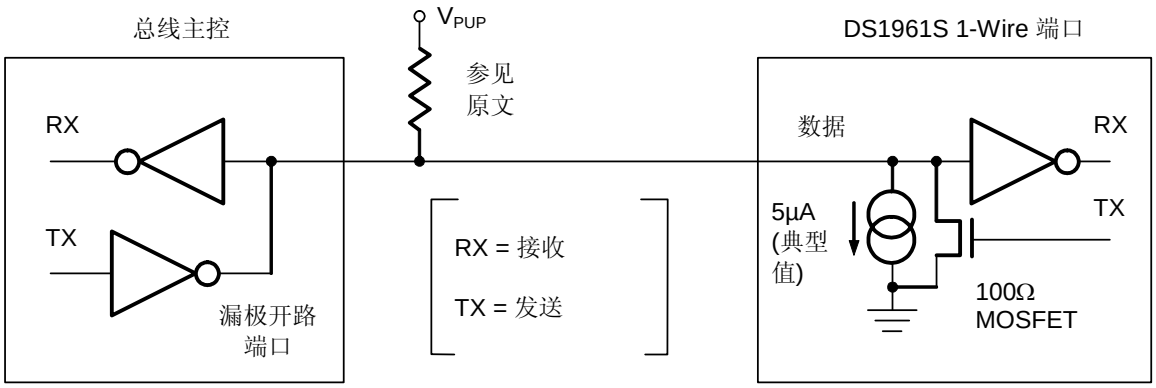
硬件结构

1-Wire 单总线系统中只定义了一根数据线，所以，保证在适当的时间驱动总线上的每个设备是非常重要的。为使上述操作易于实现，接在 1-Wire 总线上的每个装置必须都带有一个漏极开路或三态端口连接数据线。DS1961S 的 1-Wire 端口是漏极开路的，其内部等效电路如图 8 所示。

多点总线由连接了多个从机设备的 1-Wire总线组成。在标准速率下，1-Wire总线的最大速率为 16.3kbps。在高速模式下，速率可达 142kbps。DS1961S并不保证与iButton标准完全兼容。其最大速率在标准速率模式下为 14.1kbps，在高速模式下 125kbps。为了在任意速率下执行存储器和 SHA操作命令，DS1961S需要的 1-Wire上拉电阻最大值为 2.2kΩ。当与几个DS1961S同时通信时，例如安装同样的密钥给几个器件，在器件从暂存器向EEPROM传送数据时，应该利用一个上拉至V_{PUP}的低阻抗上拉旁路这个电阻。

1-Wire 总线的空闲状态是高电平。如因某种原因需要暂停通信，稍候要恢复通信的话，总线必须保持在空闲状态。如果不是这样，当总线处于低电平状态超过 16µs（高速模式）或 120µs（常规速率）时，总线上的一个或多个器件将被复位。当 DS1961S 连在总线上时，高速模式下总线电平为低的时间不能大于 15.2µs，以保证总线上没有从机器件被复位。与 DS2480B 1-Wire 驱动器和基于这一驱动器芯片的串口适配器一起使用时，除了灵活性受限，DS1961S 能够正常进行通信。

图 8. 硬件结构



处理流程

通过 1-Wire 端口访问 DS1961S 的协议如下：

- 初始化
- ROM 操作命令
- 存储器或 SHA 操作命令
- 交易/数据

初始化

1-Wire 总线上所有的传输操作均从初始化过程开始。初始化过程由主机发出的复位脉冲和从机发出的在线应答脉冲（presence pulse）组成。在线应答脉冲使主机检测到 DS1961S 挂接在总线上，并且已经准备就绪。详细内容请参阅 *1-Wire 信令*一节。

ROM 功能命令

一旦主机检测到在线应答脉冲，就可以发出 DS1961S 支持的七条 ROM 功能命令。所有 ROM 操作命令的长度为八位。以下列出了这些命令的简要介绍（见图 9 中的流程图）：

Read ROM [33h]

此条命令允许主机读取 DS1961S 的 8 位家族码、48 位唯一的序列号和 8 位 CRC 校验码。此命令适用于总线上只有一个从机的情况。如果总线上连接了多个从机设备，当同一时间每个从机设备都响应此条命令时，就必然要发生数据冲突（漏极开路输出将产生一个线与结果）。结果导致主机读取的家族码和 48 位序列号无效。

Match ROM [55h]

Match ROM 命令后面跟随 64 位注册号，允许主机访问多从机总线系统中某个特定的 DS1961S。只有与 64 位注册号完全匹配的 DS1961S 才会响应主机随后发出的存储器功能命令。所有其它从机将等待复位脉冲。这条命令既适用于单从机系统，也适用于多从机系统。

图 9-1. ROM 功能流程

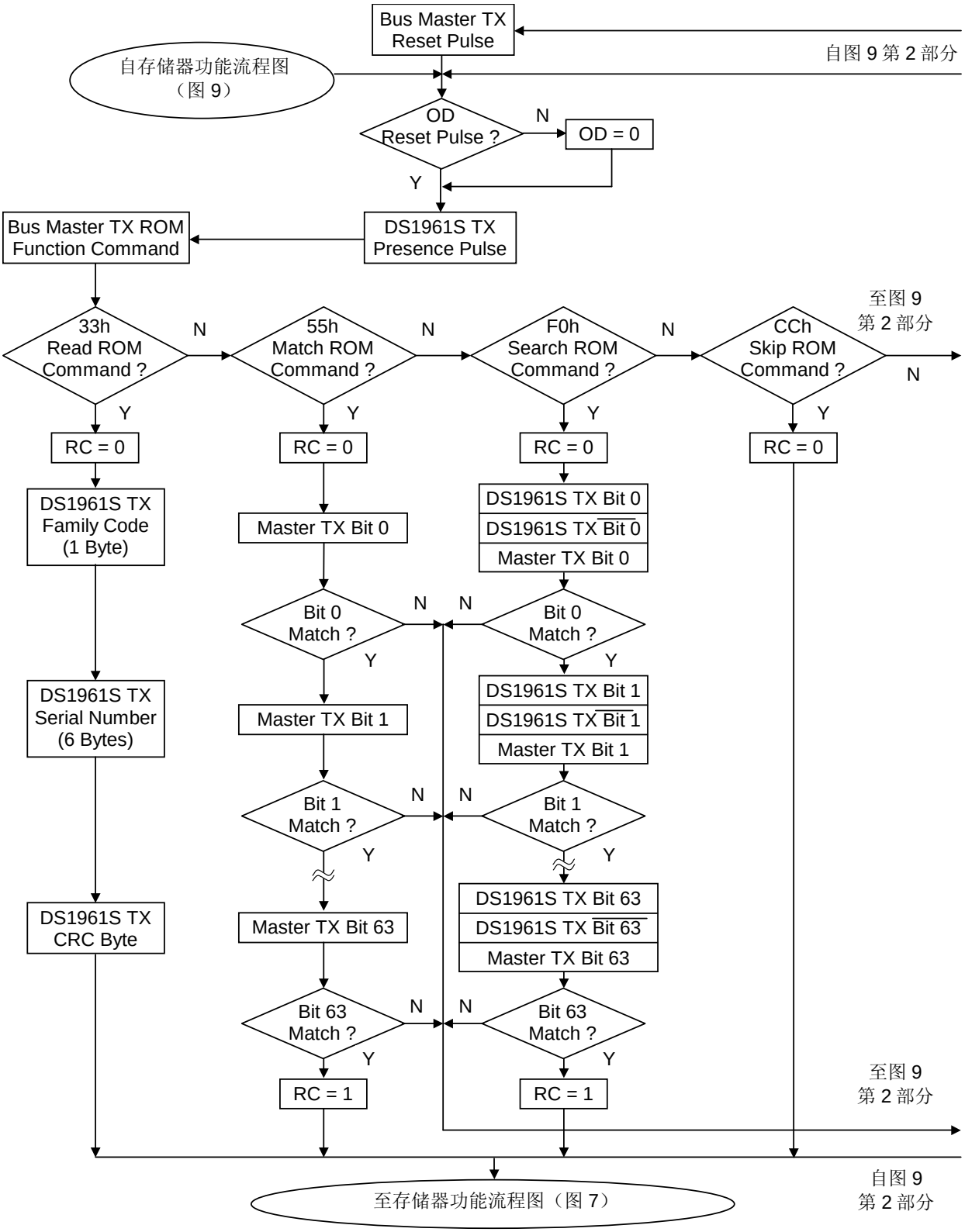
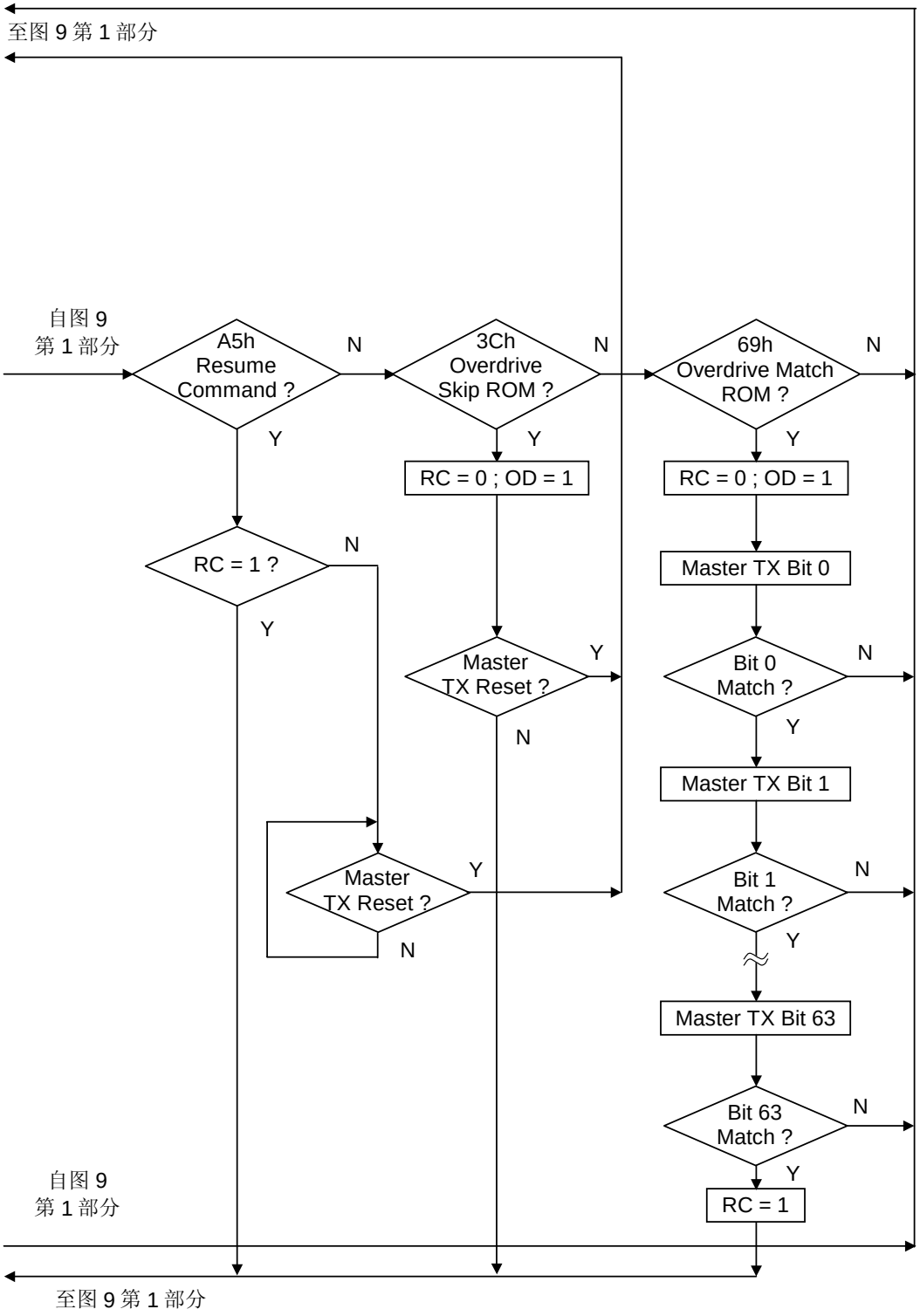


图 9-2. ROM 功能流程



Search ROM [F0h]

系统初次上电时，总线主机可能并不知道 1-Wire总线上从机设备的数目和它们的 64 位注册号，而Search ROM命令能够使得总线主机通过排除法来检测出总线上所有从机设备的 64 位注册号。Search ROM过程其实只是简单的 3 步骤重复：读一位、读此位的补码，然后写这一位的期望值，主机对注册号的每一位数据都执行这简单的 3 步骤操作。在完全通过一次审查操作后，总线主机就能读出一台从机设备的 64 位内容。其余从机设备的注册号可经由另外的操作检测出来。关于Search ROM命令更全面的讨论，请参考*Book of DS19xx iButton Standards*第 5 章，并且在此章中还包括一个实例。

Skip ROM [CCh]

Skip ROM 命令在单从机总线系统中允许主机直接访问存储器和 SHA 功能，而无须提供 64 位注册号，节省时间。如果总线上挂接了不止一个从机设备，而且在 Skip ROM 命令后发出了一条 Read 命令，总线上的从机设备就会同时传送数据，从而引起数据冲突（漏极开路输出将产生一个线与结果）。

Resume Command [A5h]

在一个典型应用中，要写满一个 32 字节的存储器页，往往需要多次访问 DS1961S。这意味着在多点环境中，每次访问都要重复执行 Match ROM 命令和发送 64 位注册号。为了提高多点环境中的数据吞吐率，设置了 Resume Command 功能。该功能检测 RC 位的状态，如果置位，就直接传递控制给存储器和 SHA 功能，类似于 Skip ROM 命令。设置 RC 位的唯一方法是成功地执行 Match ROM，Search ROM 或 Overdrive Match ROM 命令。一旦设置了 RC 位，利用 Resume Command 功能就可重复访问同一器件。对于总线上另一器件的访问将清除 RC 位，以防两个或更多的器件同时响应 Resume Command 功能。

Overdrive Skip ROM [3Ch]

在单点总线上发出该命令的时候，总线主机不需要 64 位的注册号就可以访问存储器和 SHA 功能，从而节省了时间。不同于通常的 Skip ROM 命令， Overdrive Skip ROM 命令将 DS1961S 设置成高速模式（OD = 1）。该命令代码后面的所有通信都发生在高速模式下，直到有一个最短持续 480µs 的复位脉冲把总线上的所有器件都复位到标准速率（OD = 0）。在多点总线上发出该命令时，所有支持高速模式的器件都被置为高速模式。随后，为了寻址特定的高速模式器件，必须发出一个高速模式的复位脉冲，接着运用 Match ROM 或 Search ROM 命令。这将加速搜索过程。如果总线上有多于一个支持高速模式的从机，并且 Overdrive Skip ROM 命令后接着就是 Read 命令，那么由于多个从机同时发送，总线上就会发生数据冲突（多个开漏输出下拉将产生线与结果）。

Overdrive Match ROM [69h]

通过 Overdrive Match ROM 命令，后接以高速模式发送的 64 位注册号，总线主机可以在多点总线上找到某个特定的 DS1961S，并将它设置成高速模式。只有 64 位注册号精确匹配的 DS1961S 才会响应后续的存储器或 SHA 操作命令。那些通过前面的 Overdrive Skip 或 Overdrive Match 命令已被置为高速模式的从机将继续保持高速模式。直到有一个最短持续时间 480µs 的复位脉冲发出后，所有高速模式的器件将返回常规速率。命令 Overdrive Match ROM 适用于总线上有单个或多个器件的情况。

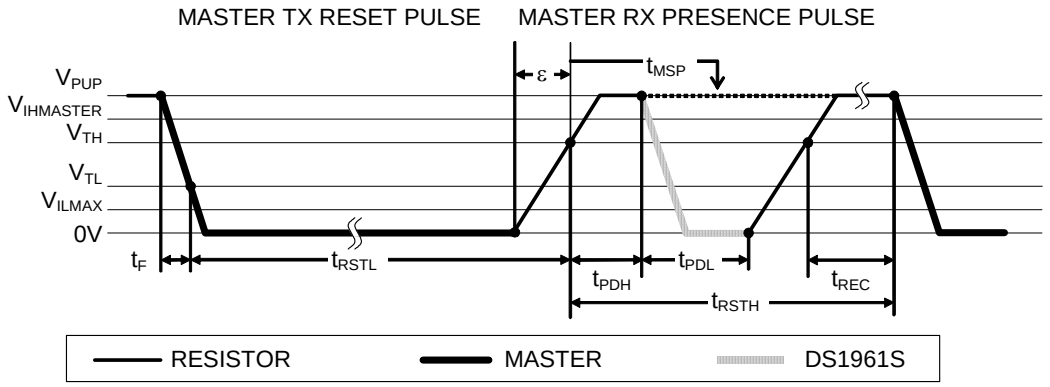
1-Wire 信令

为了保证数据的完整性，DS1961S 具有一个严格的信号协议。该协议在一条线上定义了四种类型的信号：包括复位脉冲和在线应答脉冲的复位序列，写 0，写 1，和读数据。除了在线应答脉冲以外，所有其它信号均由总线主机发出。DS1961S 能够以两种不同速率通信：标准速率和高速模式。如果没有明确设定为高速模式，DS1961S 就以标准速率通信。高速模式下，所有波形均采用快速定时。

从空闲状态被激活时，1-Wire线上的电平需要从V_{PUP}降到阈值电平V_{TL}以下。从活动状态返回空闲状态，电平需要从V_{ILMAX}升至阈值电平V_{TH}以上。电平V_{ILMAX}仅与DS1961S决定逻辑电平有关，而不触发任何事件。

与DS1961S进行通信所需的初始化时序见图 10。收到正确的ROM或存储器功能命令后，复位脉冲后面跟随一个在线应答脉冲表明DS1961S已经准备好发送或接收数据。在一个混合设备组成的网络中，复位脉冲低电平t_{RSTL}时间必须足够长，以保证速率最慢的从机识别出这是一个复位脉冲。t_{RSTL}时间取决于通信速率和 1-Wire上拉电平（见电气特性）。如果主机在下降沿采用摆率控制，作为补偿，需要将拉低总线电平的时间延长到t_{RSTL} +t_F。如果DS1961S处于高速模式，一个标准速率的复位脉冲会将其恢复成标准速率。如需DS1961S仍保持高速模式，t_{RSTL}时间不能超过高速模式定义的最大值。

图 10. 初始化过程（复位和在线应答脉冲）



总线主机释放数据线后进入接收模式（RX）。1-Wire总线电平被上拉电阻或DS2480B驱动器等有源上拉电路拉升至V_{PUP}。当电平高于门限V_{TH}时，DS1961S在等待t_{PDH}时间后通过将总线电平拉低并保持t_{PDL}时间，发出一个在线应答脉冲。为了检测该应答脉冲，主机必须在t_{MSP}时刻检测总线的逻辑电平。

t_{RSTH}窗口时间必须至少等于t_{PDHMAX} , t_{PDLMAX}, 与t_{RECMIN}的总和。一旦t_{RSTH}结束，DS1961S即可开始数据通信。在一个混和设备组成的网络中，为了兼容其它 1-Wire设备，t_{RSTH}在标准速率下最小应为 480µs，在高速模式下最小应为 48µs。

读/写时隙

DS1961S 的数据通信在时隙内进行，每时隙传输一位。数据在写时隙由主机传输到从机。数据在读时隙由从机传输到主机。图 11 说明了读时隙和写时隙的定义。

所有通信均以主机拉低数据线电平开始。当 1-Wire总线上的电平降至门限电平V_{TL}以下时，DS1961S启动内部时基，从机时基的偏差使从机采样窗口从t_{SLSMIN}延伸到t_{SLSMAX}。采样点的总线电平决定了从机将该时隙解码为 1 还是 0。为了保证可靠的通信，在整个采样窗口内电平值应该一直保持低于V_{ILMAX}或高于最大的V_{TH}。

主机到从机

对于一个“写 1”时隙，主机拉低总线电平的时间（$t_{MPD1} = t_{W1L} - \varepsilon + t_F$）必须足够短，以保证 1-Wire总线电平在DS1961S最早的采样点t_{SLSMIN}能达到V_{TH}，在最迟的采样点（t_{SLSMAX}）过后，在下一个时隙开始之前要保证一定的恢复时间（t_{REC}）。

对于一个“写 0”时隙，主机拉低总线电平的时间（$t_{MPD0} = t_{W0L} + t_F$）必须足够长，以保证数据总线电平在较慢的DS1961S采样时（为t_{SLSMAX}）仍保持低于V_{ILMAX}。在下一个时隙开始前，数据线电平需要首先升至高于V_{TH}，并一直保持到恢复时间t_{REC}结束。

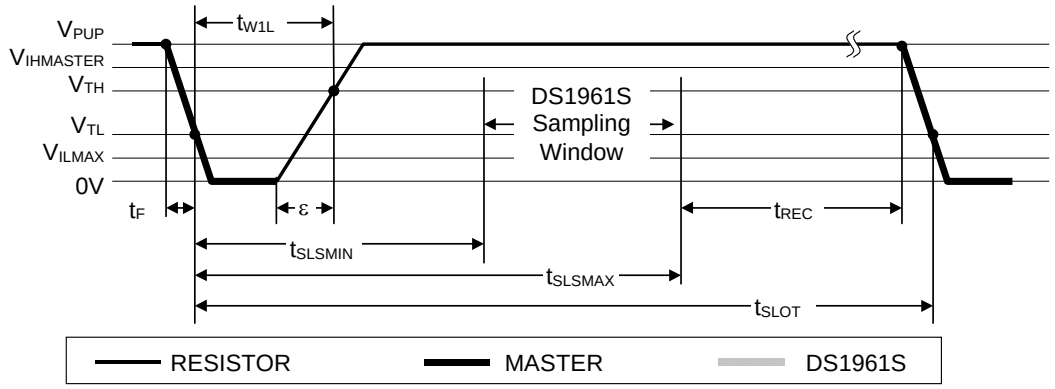
从机到主机

“读数据”时隙与“写 1”时隙非常类似。主机将总线电平拉低启动一个读数据时隙。当 1-Wire总线电平降至门限V_{TL}以下时，DS1961S启动内部时基，主机拉低总线电平的时间（$t_{MPDR} = t_{RL} + t_F$） 必须足够长，以保证足够的建立时间t_{SU}，此后DS1961S通过其 1-Wire端口发送一位。发送 0 时，DS1961S将总线低电平保持t_{SPD}时间，发送 1 时，DS1961S并不保持总线电平。

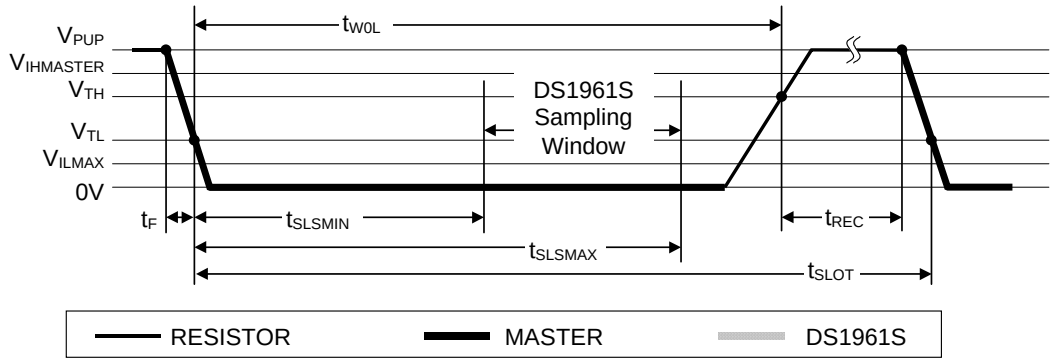
主机在t_{MSR}时刻采样数据线，该时刻位于一个既受t_{RL}和上升时间（δ）之和，又受t_{SPDMIN}决定的窗口内。读 0 时最佳采样点不迟于t_{SPDMIN}，读 1 时 1-Wire总线电平在t_{MSR}时刻必须能够达到$V_{IHMASTER}$。这决定了最大的主机下拉时间。为实现最可靠的通信，主机下拉时间应尽可能短，以最大化数据线电平到达V_{IHMIN}的时间。在下一个时隙开始前，t_{SPDMAX}应已结束，数据线上的电平必须已经升至高于V_{TH}，并保持该电平直到t_{REC}结束。

图 11. 读/写时序图

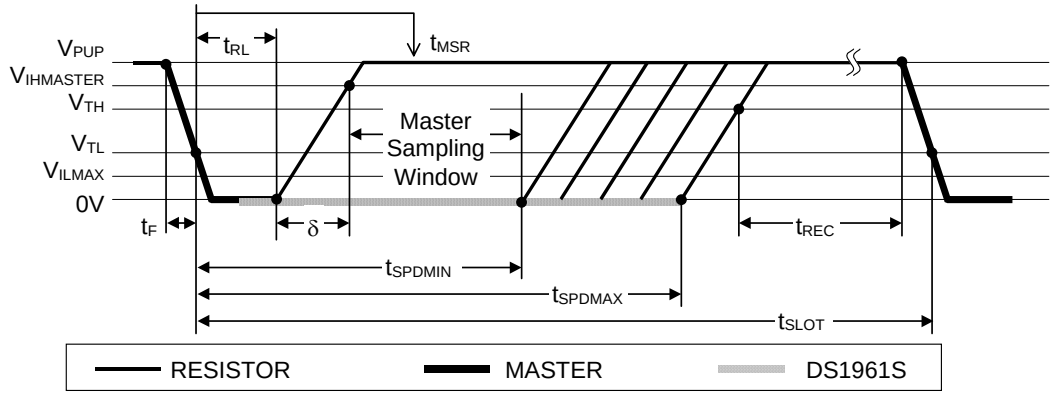
写 1 时隙



写 0 时隙



读数据时隙



CRC 生成

DS1961S有两种类型的循环冗余校验（CRC）。其中一种类型是 8 位的，在出厂时就已经计算好了，并用激光写入 64 位ROM的最高字节中。该CRC的等价多项式是$X^8 + X^5 + X^4 + 1$。为了确定ROM数据是否被无差错地读取，总线主机可用 64 位ROM的前 56 位计算CRC值，并将其与从

DS1961S读来的值相比较。读ROM的时候，接收到的是 8 位CRC校验码的原码形式（未求反的）。

另一类CRC是 16 位的，根据标准的CRC16 多项式函数$X^{16} + X^{15} + X^2 + 1$产生。该CRC校验码用于检测执行Read Authenticated Page命令时的错误，或者在读、写或更新暂存器的时候，快速检验数据传送的正确性。在iButton扩展文件结构中用于差错检验的也是同一种CRC。与 8 位CRC校验码不同的是，16 位CRC校验码通常是以反码的形式发送或回送。DS1961S芯片内部的CRC发生器（图 12）用于在图 7 所示的命令流程中计算一个新的 16 位CRC校验码。总线主机通过比较由器件读来的CRC校验码和自己根据数据计算出的CRC校验码，来决定是继续某一操作还是重读有CRC错误的数据部分。

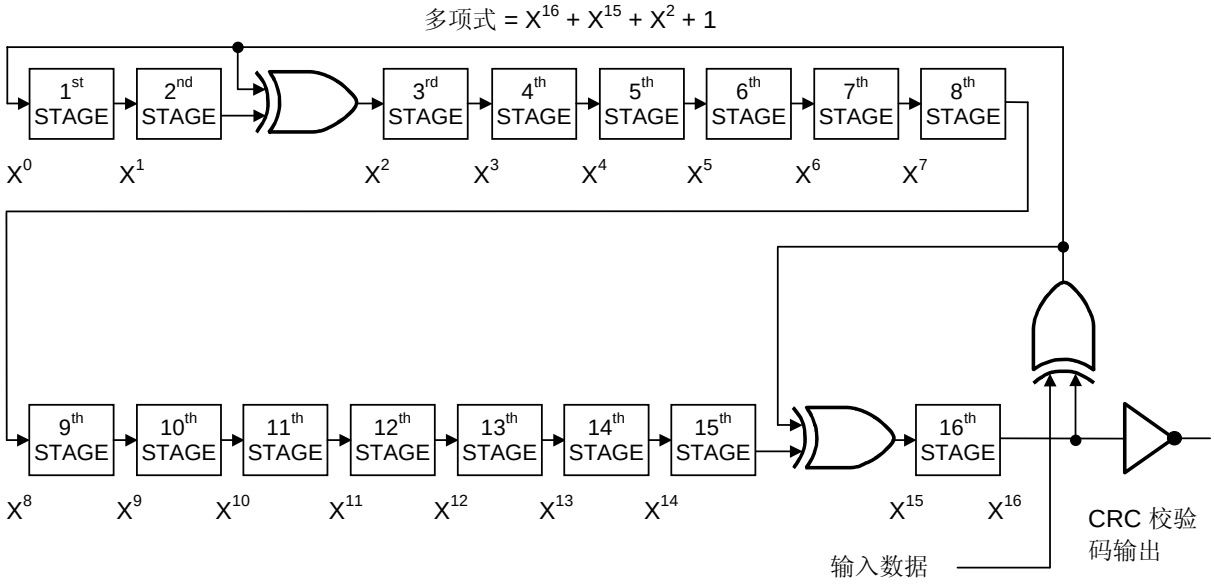
Write Scratchpad 或 Refresh Scratchpad 命令执行时，首先清除 CRC 发生器，然后移入命令代码，目的地址 TA1（T2 至 T0 均置 0）和 TA2，以及所有主机发送的数据字节。只有当主机正确发送八个字节时，DS1961S 才发送该 CRC 校验码。

在 Read Scratchpad 命令中，首先清空 CRC 发生器，然后移入命令代码，目的地址 TA1 和 TA2，E/S 字节，和暂存器数据，它们可能已被 DS1961S 调整过（见 Write Scratchpad 命令），最后产生了 CRC 校验码。只有读到暂存器末尾的时候，DS1961S 才发送该 CRC 校验码。

在 Read Authenticated Page 命令中，16 位 CRC 校验码是清空 CRC 发生器并移入命令字节、两个地址字节、数据字节、和 FFh 字节后的结果。跟在 MAC 结果后面的 CRC 校验码是在清空 CRC 发生器后，按照主机接收的位序移入 160 位 MAC 后产生的。

关于产生CRC校验码的详细资料，以及用硬件和软件实现的具体实例，参见*Book of DS19xx iButton Standards*。

图 12. CRC-16 硬件和多项式描述



极限参数*

I/O 对地电压	-0.5V，+6V
I/O 吸入电流	20mA
温度范围	-40°C 至+85°C
结温度	+150°C
存储温度范围	-55°C 至+85°C

** 这只是一个应力条件下的参数，并不意味着器件可以在符合或超出该规定所提及的工作条件下执行功能。长期暴露器件于极限条件会影响其可靠性。*

电气特性

参数	符号	条件	最小	典型	最大	单位	注释
工作温度	T_A	除 EEPROM 编程以外的全部	-40		85	°C	1
		全部	-20		85		
1-Wire 上拉	V_{PUP}	标准速率	2.8		5.25	V	1
		高速	3.3		5.25		
I/O 引脚通用数据							
1-Wire 上拉电阻	R_{PUP}				2.2	kΩ	1, 2
输入电容	C_{IO}			100	800	pF	3, 14
输入负载电流	I_L	在V_{PUP} 时的I/O口	1		10	µA	4
高到低的开关门限	V_{TL}			1.5		V	5, 6, 7, 14
输入低电平	V_{IL}				0.30	V	1, 5, 8
低到高的开关门限	V_{TH}			1.5		V	5, 6, 9, 14
4mA 时输出低电平	V_{OL}				0.4	V	5, 10
恢复时间	t_{REC}	标准速率, R_{PUP} = 2.2kΩ	5			µs	1, 14
		高速, R_{PUP} = 2.2kΩ	2				
		高速, 在复位脉冲之前; R_{PUP} = 2.2kΩ	5				
时隙时间	t_{SLOT}	标准速率	65			µs	1, 13
		高速, V_{PUP} > 4.5V	7				
		高速	9				
I/O 引脚, 1-Wire 复位, 在线应答检测周期							
复位低电平时间	t_{RSTL}	标准速率 V_{PUP} > 4.5V	480		640	µs	1, 13
		标准速率	720		960		
		高速, V_{PUP} > 4.5V	60		80		
		高速	68		80		

参数	符号	条件	最小	典型	最大	单位	注释
在线检测高电平时间	t_{PDH}	标准速率	15		60	µs	13
		高速, V_{PUP} > 4.5V	1		5		
		高速	1		6.7		
在线检测低电平时间	t_{PDL}	标准速率	60		285	µs	13
		高速, V_{PUP} > 4.5V	7.3		24		
		高速	7.3		28		
在线检测采样时间	t_{MSP}	标准速率	60		75	µs	1, 14
		高速, V_{PUP} > 4.5V	5		8.3		
		高速	6.7		8.3		
I/O 引脚, 1-Wire 写							
写 0 低电平时间	t_{W0L}	标准速率	60		120		13, 1
		高速, V_{PUP} > 4.5V	5		14		
		高速	7		14		
写 1 低电平时间	t_{W1L}	标准速率	5		15 - ε	µs	1, 11, 13
		高速, V_{PUP} > 4.5V	1		2 - ε		
		高速	1		1.85 - ε		
写采样时间 (从机采样)	t_{SLS}	标准速率	15		60	µs	13
		高速, V_{PUP} > 4.5V	2		5		
		高速	1.85		7		
I/O 引脚, 1-Wire 读							
读低电平时间	t_{RL}	标准速率	5		15 - δ	µs	1, 12,13
		高速, V_{PUP} > 4.5V	1		2 - δ		
		高速	1		1.85 - δ		
读 0 低电平 (数据来自从机)	t_{SPD}	标准速率	15		60	µs	13
		高速, V_{PUP} > 4.5V	2		5		
		高速	1.85		7		
读采样时间	t_{MSR}	标准速率	t_{RL} + δ		15	µs	1, 12, 13
		高速, V_{PUP} > 4.5V	t_{RL} + δ		2		
		高速	t_{RL} + δ		1.85		
EEPROM							
编程电流	I_{LPROG}				700	µA	14
编程时间	t_{PROG}				10	ms	
写/擦除周期	N_{CYCLE}		50k			—	14
数据保持时间	t_{RET}	+85°C, 不加电	10			年	