1-wire 单总线的基本原理

一、概述

1-wire 单总线是 Maxim 全资子公司 Dallas 的一项专有技术。与目前多数标准串行数据通信方式，如 SPI/I^2C/MICROWIRE 不同，它采用单根信号线，既传输时钟，又传输数据，而且数据传输是双向的。它具有节省 I/O 口线资源、结构简单、成本低廉、便于总线扩展和维护等诸多优点。

1-wire 单总线适用于单个主机系统，能够控制一个或多个从机设备。当只有一个从机位于总线上时，系统可按照单节点系统操作；而当多个从机位于总线上时，则系统按照多节点系统操作。

为了较为全面地介绍单总线系统，将系统分为三个部分讨论：硬件结构、命令序列和信号方式（信号类型和时序）。

二、硬件结构

顾名思义，单总线只有一根数据线。设备（主机或从机）通过一个漏极开路或三态端口，连接至该数据线，这样允许设备在不发送数据时释放数据总线，以便总线被其它设备所使用。单总线端口为漏极开路，其内部等效电路如图 1 所示。

单总线要求外接一个约 5k 的上拉电阻；这样，单总线的闲置状态为高电平。不管什么原因，如果传输过程需要暂时挂起，且要求传输过程还能够继续的话，则总线必须处于空闲状态。位传输之间的恢复时间没有限制，只要总线在恢复期间处于空闲状态（高电平）。如果总线保持低电平超过 480μs，总线上的所有器件将复位。另外，在寄生方式供电时，为了保证单总线器件在某些工作状态下（如温度转换期间、EEPROM 写入等）具有足够的电源电流，必须在总线上提供强上拉（如图 1 所示的 MOSFET）。

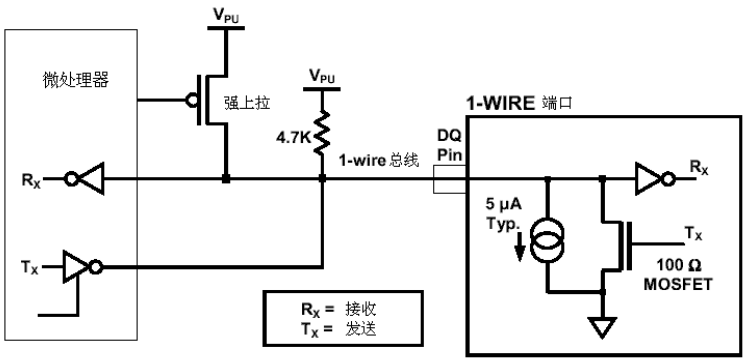

图1、单总线的硬件接口示意图

三、命令序列

典型的单总线命令序列如下：

第一步：初始化

第二步：ROM 命令（跟随需要交换的数据）

第三步：功能命令（跟随需要交换的数据）

每次访问单总线器件，必须严格遵守这个命令序列，如果出现序列混乱，则单总线器件不会响应主机。但是，这个准则对于搜索 ROM 命令和报警搜索命令例外，在执行两者中任何一条命令之后，主机不能执行其后的功能命令，必须返回至第一步。

3．1 初始化

基于单总线上的所有传输过程都是以初始化开始的，初始化过程由主机发出的复位脉冲和从机响应的应答脉冲组成。应答脉冲使主机知道，总线上有从机设备，且准备就绪。复位和应答脉冲的时间详见单总线信号部分。

3．2 ROM 命令

在主机检测到应答脉冲后，就可以发出 ROM 命令。这些命令与各个从机设备的唯一 64 位 ROM 代码相关，允许主机在单总线上连接多个从机设备时，指定操作某个从机设备。这些命令还允许主机能够检测到总线上有多少个从机设备以及其设备类型，或者有没有设备处于报警状态。从机设备可能支持 5 种 ROM 命令（实际情况与具体型号有关），每种命令长度为 8 位。主机在发出功能命令之前，必须送出合适的 ROM 命令。ROM 命令的操作流程如图 2 所示。下面将简要地介绍各个 ROM 命令的功能，以及使用在何种情况下。

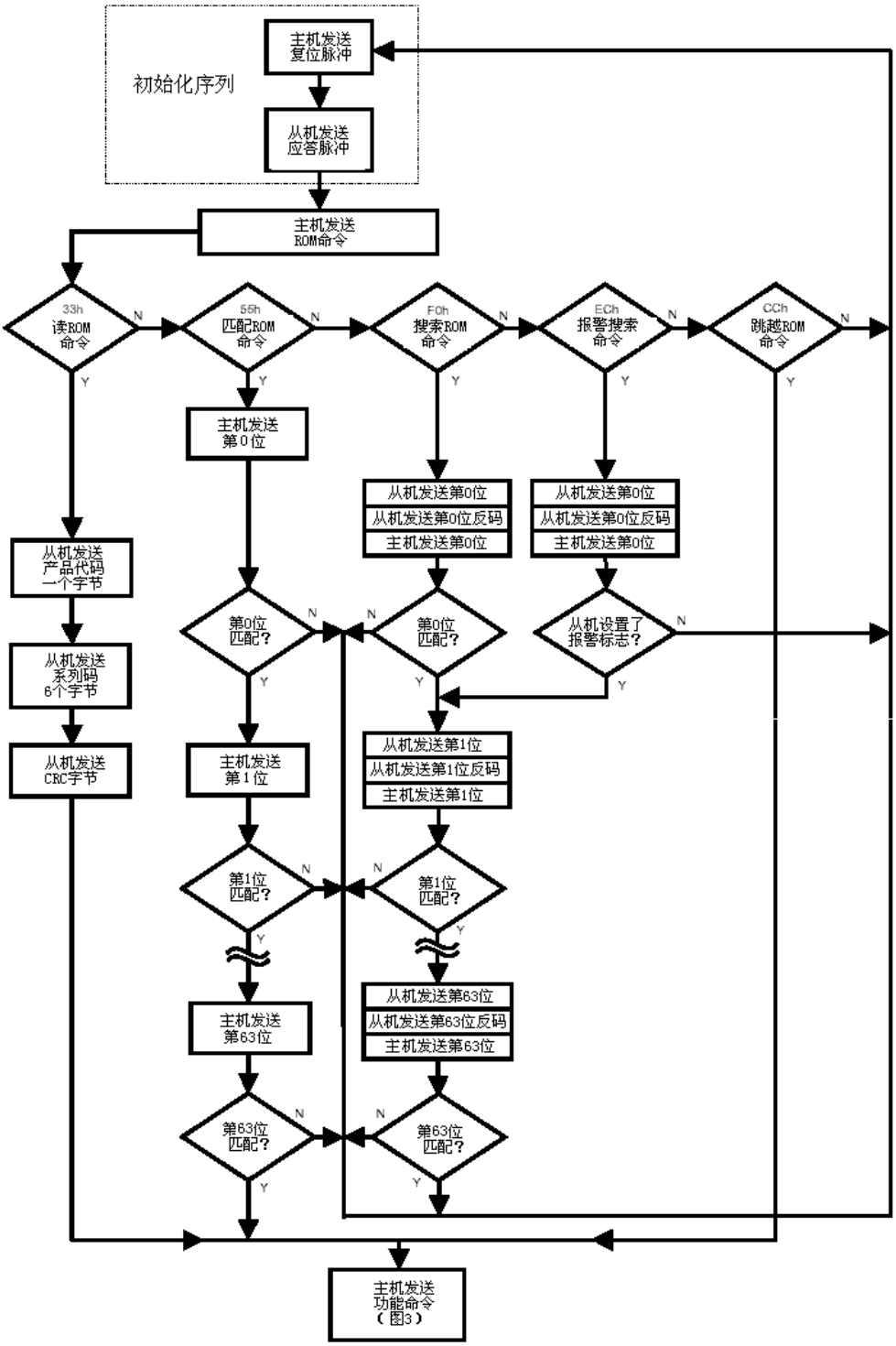

图2、ROM命令流程图

3.2.1 搜索 ROM[F0h]

当系统初始上电时，主机必须找出总线上所有从机设备的 ROM 代码，这样主机就能够判断出从机的数目和类型。主机通过重复执行搜索 ROM 循环（搜索 ROM 命令跟随着位数据交换），以找出总线上所有的从机设备。如果总线只有一个从机设备，则可以采用读 ROM 命令来替代搜索 ROM 命令。关于搜索 ROM 命令的详细解释，请参见附录 A。在每次执行完搜索 ROM 循环后，主机必须返回至命令序列的第一步（初始化）。

3.2.2 读 ROM[33h]（仅适合于单节点）

该命令仅适用于总线上只有一个从机设备。它允许主机直接读出从机的 64 位 ROM 代码，而无须执行搜索 ROM 过程。如果该命令用于多节点系统，则必然发生数据冲突，因为每个从机设备都会响应该命令。

3.2.3 匹配 ROM[55h]

匹配 ROM 命令跟随 64 位 ROM 代码，从而允许主机访问多节点系统中某个指定的从机设备。仅当从机完全匹配 64 位 ROM 代码时，才会响应主机随后发出的功能命令；其它设备将处于等待复位脉冲状态。

3.2.4 跳越 ROM[CCh]（仅适合于单节点）

主机能够采用该命令同时访问总线上的所有从机设备，而无须发出任何 ROM 代码信息。例如，主机通过在发出跳越 ROM 命令后跟随转换温度命令[44h]，就可以同时命令总线上所有的 DS18B20 开始转换温度，这样大大节省了主机的时间。值得注意，如果跳越 ROM 命令跟随的是读暂存器[BEh]的命令（包括其它读操作命令），则该命令只能应用于单节点系统，否则将由于多个节点都响应该命令而引起数据冲突。

3.2.5 报警搜索[ECh]（仅少数 1-wire 器件支持）

除那些设置了报警标志的从机响应外，该命令的工作方式完全等同于搜索 ROM 命令。该命令允许主机设备判断那些从机设备发生了报警（如最近的测量温度过高或过低等）。同搜索 ROM 命令一样，在完成报警搜索循环后，主机必须返回至命令序列的第一步。

3．3 功能命令（以 DS18B20 为例）

在主机发出 ROM 命令，以访问某个指定的 DS18B20，接着就可以发出 DS18B20 支持的某个功能命令。这些命令允许主机写入或读出 DS18B20 暂存器、启动温度转换以及判断从机的供电方式。DS18B20 的功能命令总结于表 1 中，并在图 3 流程图中作了说明。

表 1、DS18B20 功能命令集

命令	描述	命令代码	发送命令后，单总线上的响应信息	注释
温度转换命令				
转换温度	启动温度转换	44h	无	1
存储器命令				
读暂存器	读全部的暂存器内容，包括 CRC 字节	BEh	DS18B20 传输多达 9 个字节至主机	2
写暂存器	写暂存器第 2、3 和 4 个字节的数据（即 T_H，T_L 和配置寄存器）	4Eh	主机传输 3 个字节数据至 DS18B20	3
复制暂存器	将暂存器中的 T_H，T_L 和配置字节复制到 EEPROM 中	48h	无	1
回读 EEPROM	将 T_H，T_L 和配置字节从 EEPROM 回读至暂存器中	B8h	DS18B20 传送回读状态至主机	

注释：

1．在温度转换和复制暂存器数据至 EEPROM 期间，主机必须在单总线上允许强上拉。并且在此期间，总线上不能进行其它数据传输；

2．通过发出复位脉冲，主机能够在任何时候中断数据传输；

3．在复位脉冲发出前，必须写入全部的三个字节。

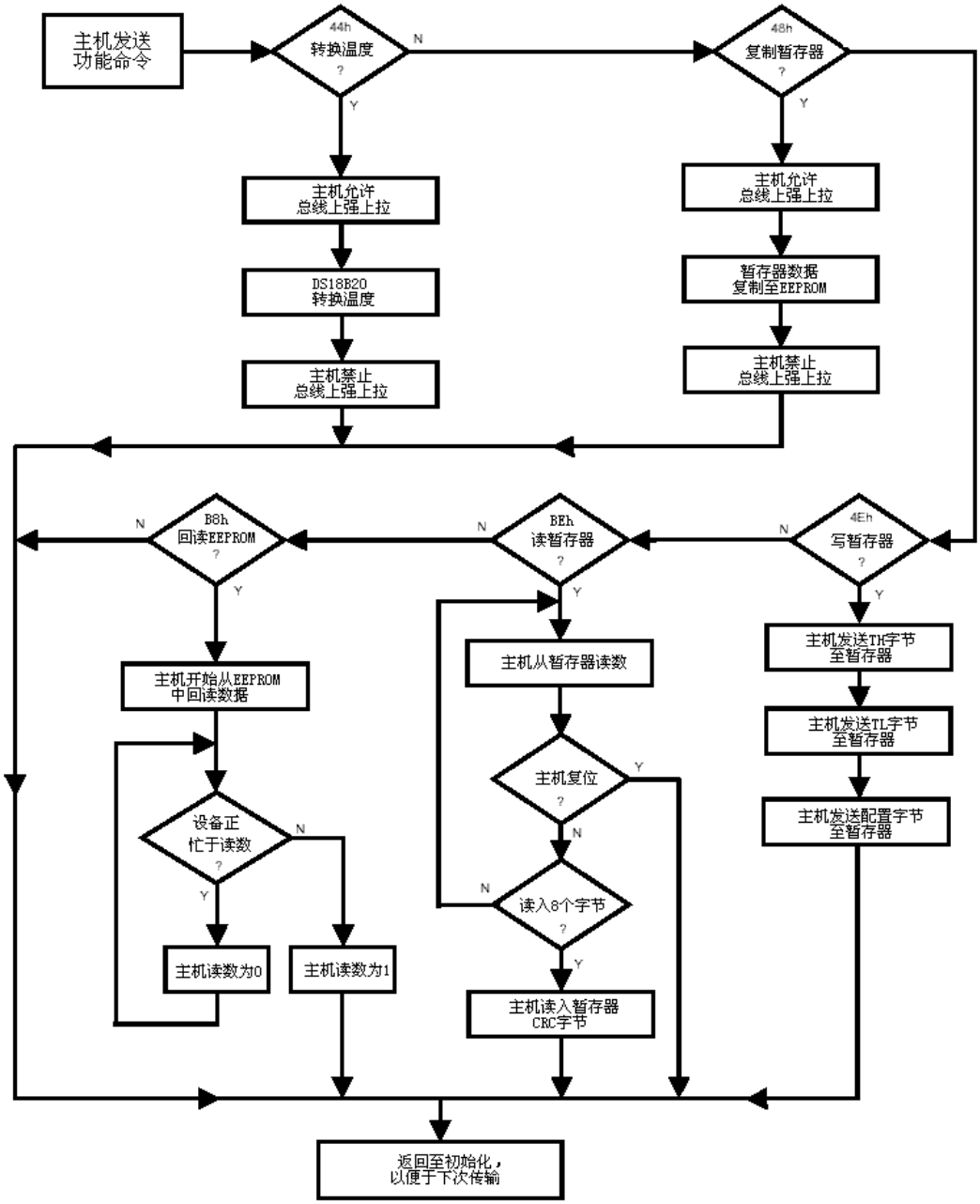

图3、DS18B20的功能命令流程图

四、信号方式

所有的单总线器件要求采用严格的通信协议，以保证数据的完整性。该协议定义了几种信号类型：复位脉冲、应答脉冲、写 0、写 1、读 0 和读 1。所有这些信号，除了应答脉冲以外，都由主机发出同步信号。并且发送所有的命令和数据都是字节的低位在前，这一点与多数串行通信格式不同（多数为字节的高位在前）。

4．1 初始化序列：复位和应答脉冲

单总线上的所有通信都是以初始化序列开始，包括：主机发出的复位脉冲及从机的应答

脉冲，如图 4 所示。当从机发出响应主机的应答脉冲时，即向主机表明它处于总线上，且工作准备就绪。在主机初始化过程，主机通过拉低单总线至少 480μs，以产生（Tx）复位脉冲。接着，主机释放总线，并进入接收模式（Rx）。当总线被释放后，5k 上拉电阻将单总线拉高。在单总线器件检测到上升沿后，延时 15-60μs，接着通过拉低总线 60-240μs，以产生应答脉冲。

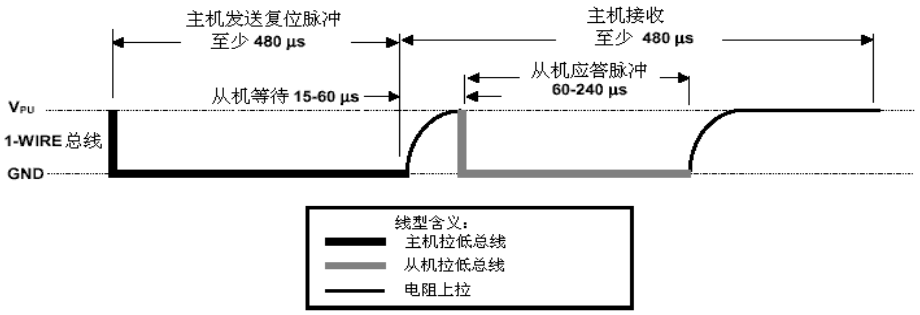

图4、单总线初始化时序图

4．2 读/写时隙

在写时隙期间，主机向单总线器件写入数据；而在读时隙期间，主机读入来自从机的数据。在每一个时隙，总线只能传输一位数据。

4．2．1 写时隙

存在两种写时隙：“写 1”和“写 0”。主机采用写 1 时隙向从机写入 1，而采用写 0 时隙向从机写入 0。所有写时隙至少需要 60μs，且在两次独立的写时隙之间至少需要 1μs 的恢复时间。两种写时隙均起始于主机拉低总线（图 5 所示）。产生写 1 时隙的方式：主机在拉低总线后，接着必须在 15μs 之内释放总线，由 5k 上拉电阻将总线拉至高电平；而产生写 0 时隙的方式：在主机拉低总线后，只需在整个时隙期间保持低电平即可（至少 60μs）。

在写时隙起始后 15-60μs 期间，单总线器件采样总线电平状态。如果在此期间采样为高电平，则逻辑 1 被写入该器件；如果为 0，则写入逻辑 0。

4．2．2 读时隙

单总线器件仅在主机发出读时隙时，才向主机传输数据，所以，在主机发出读数据命令后，必须马上产生读时隙，以便从机能够传输数据。所有读时隙至少需要 60μs，且在两次独立的读时隙之间至少需要 1μs 的恢复时间。每个读时隙都由主机发起，至少拉低总线 1μs（图 5 所示）。在主机发起读时隙之后，单总线器件才开始在总线上发送 0 或 1。若从机发送 1，则保持总线为高电平；若发送 0，则拉低总线。当发送 0 时，从机在该时隙结束后释放总线，由上拉电阻将总线拉回至空闲高电平状态。从机发出的数据在起始时隙之后，保持有效时间 15μs，因而，主机在读时隙期间必须释放总线，并且在时隙起始后的 15μs 之内采样总线状态。

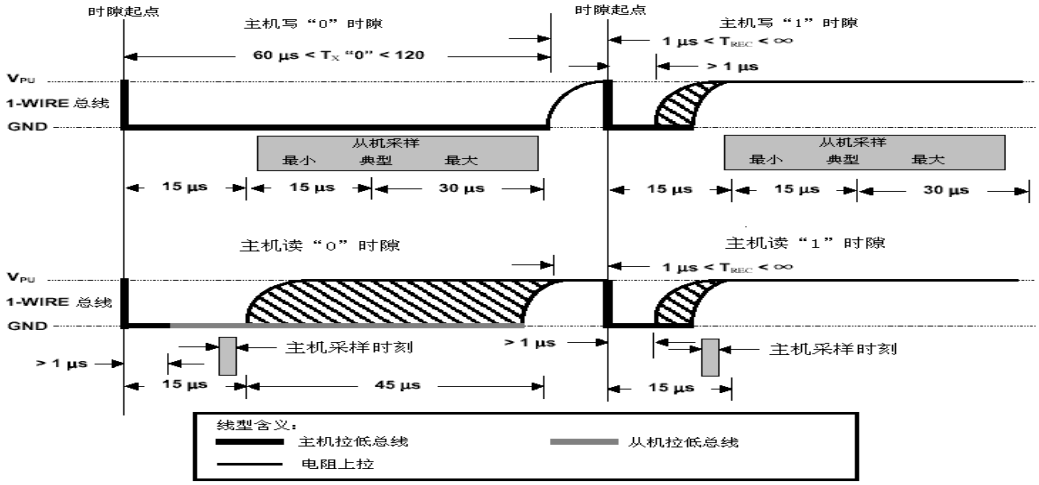

图5、主机读/写时隙的时序示意图

附录 A：ROM 搜索实例

ROM 搜索过程只是一个简单的三步循环程序：读一位、读该位的补码、写入一个期望的数据位。总线主机在 ROM 的每一位上都重复这样的三步循环程序。当完成某个器件后，主机就能够知晓该器件的 ROM 信息。剩下的设备数量及其 ROM 代码通过相同的过程即可获得。

下面的 ROM 搜索过程实例假设四个不同的器件被连接至同一条总线上，它们的 ROM 代码如下所示：

ROM1　00110101…
ROM2　10101010…
ROM3　11110101…
ROM4　00010001…

具体搜索过程如下：

1．主机发出复位脉冲，启动初始化序列。从机设备发出响应的应答脉冲；
2．接着主机在总线上发出 ROM 搜索命令；
3．主机从总线上准备读入一个数据位，这时，每个响应设备分别将 ROM 代码的第一位输出到单总线上。ROM1 和 ROM4 输出 0 至总线，而 ROM2 和 ROM3 输出 1 至总线。线上的输出结果将是所有器件的逻辑“与”，所以，主机从总线上读到的将是 0。接着，主机开始读另一位，即每个器件分别输出 ROM 代码中第一位的补码，此时，ROM1 和 ROM4 输出 1 至总线，而 ROM2 和 ROM3 输出 0 至总线。这样，主机读到的该位补码还是 0。主机由此判定，总线上有些器件的 ROM 代码第一位为 0，有些则为 1；

两次读到的数据位具有以下含义：

00　在该位处，存在设备冲突；
01　在该位处，所有器件为 0；
10　在该位处，所有器件为 1；
11　单总线不存在任何设备。

4．主机写入 0，从而禁止了 ROM2 和 ROM3 响应余下的搜索命令，仅在总线上留下了 ROM1 和 ROM4；
5．主机再执行两次读操作，依次收到 0 和 1，这表明 ROM1 和 ROM4 在 ROM 代码的第二位都是 0；
6．接着主机写入 0，在总线上继续保持 ROM1 和 ROM4；
7．主机又执行两次读操作，收到两个 0，表明所连接的设备的 ROM 代码在第三位既有 0，也有 1；
8．主机再次写入 0，从而禁止了 ROM1 响应余下的搜索命令，仅在总线上留下了 ROM2；
9．主机读完 ROM4 余下的 ROM 数据位。这样就完成了第一次搜索，并找到了位于总线上的第一个设备；
10．重复执行第 1 至第 7 步，开始新一轮的 ROM 搜索命令；
11．主机写入 1，使 ROM4 离线，仅在总线上留下 ROM1；
12．主机读完 ROM1 余下的 ROM 数据，这样就完成了第二次的 ROM 搜索，找到了第二个 ROM 代码；
13．重复执行第 1 至第 3 步，开始新一轮的 ROM 搜索命令；
14．主机写入 1，这次禁止了 ROM1 和 ROM4 响应余下的搜索命令，仅在总线上留下了 ROM2 和 ROM3；
15．主机又执行两次读操作时隙，读到两个 0；
16．主机写入 0，这样禁止了 ROM3，而留下了 ROM2；

17．主机读完 ROM2 余下的 ROM 数据，这样就完成了第三次的 ROM 搜索，找到了第三个 ROM 代码；
18．重复执行第 13 至第 15 步，开始新一轮的 ROM 搜索命令；
19．主机写入 1，这次禁止了 ROM2，而留下了 ROM3；
20．主机读完 ROM3 余下的 ROM 数据，这样就完成了第四次的 ROM 搜索，找到了第四个 ROM 代码。
说明：

每次搜索 ROM 操作，主机只能找到某一个单总线器件的 ROM 代码，所需要的最短时间为：

960μs+(8+3×64) ×61μs=13.16ms

所以，主机能够在 1 秒之内读出 75 个单总线的 ROM 代码。