
单总线数字温度传感器

18B20

产 品 手 册

目录

特点.....	3	Skip ROM [CCh]	12
典型应用	3	Alarm Search [ECh]	12
概述.....	3	18B20 功能指令	12
管脚图和管脚描述	3	CONVERT T [44h]	12
结构框图	4	WRITE SCRATCHPAD [4Eh].....	12
运行-测量温度	4	READ SCRATCHPAD [BEh]	12
运行-报警信号	5	COPY SCRATCHPAD [48h]	12
给 18B20 供电.....	6	RECALL E ² [B8h]	12
64 位 ROM 编码	7	READ POWER SUPPLY [B4h].....	13
存储.....	8	单总线信令	16
配置寄存器	9	初始化程——复位与存在脉冲	16
循环冗余校验码生成.....	9	读/写时隙.....	16
单总线系统	10	写时隙.....	16
硬件配置	10	读时隙.....	16
传输序列	11	18B20 运行示例 1	19
初始化.....	11	18B20 运行示例 2	20
ROM 操作指令	11	电气特性.....	21
SEARCH ROM [F0h]	11	绝对最大额定值.....	21
Read ROM [33h].....	11	直流电气特性	21
Match ROM [55h]	11	交流电气特性-非易失性存储器	22
		交流电气特性	22

概述

特点

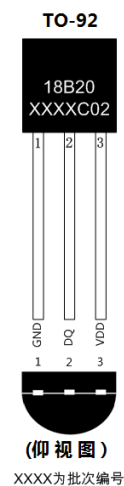
- 单总线接口,节约布线资源
- 应用简单,无需额外器件
- 转换温度时间 500ms
- 可编程 9~12 位数字输出
- 宽供电电压范围 2.7V-5.5V
- 每颗芯片有可编程的 ID 序列号
- 用户可自行设置报警值
- 超强 ESD 保护能力 (HBM>8000V)
- 典型待机电流功耗 1μA@3V
- 典型换电流功耗 0.6mA@3V

典型应用

- 温度监控
- 粮情监测
- 智能家电系统
- 水温测量
- 建筑、设备或机房内部温度监测系统
- 过程监测和控制系统

18B20 是一款高精度的单总线温度测量芯片。温度传感器的测温范围为-55℃ 到+125℃ ;根据用户需要通过配置寄存器可以设定数字转换精度和测温速度。芯片内置 4byte 非易失性存储单元供用户使用, 2byte 用于高低温报警, 另外 2byte 用于保存用户自定义信息。在-10℃ 到+85℃ 范围内最大误差为±0.5℃, 在全温范围内最大误差为±1.5℃。用户可自主选择电源供电模式和寄生供电模式。单总线接口允许多个设备挂在同一总线, 该特性使得 18B20 也非常便于部署分布型温度采集系统。

管脚图和管脚描述



引脚	名称	功能
1	GND	地线
2	DQ	单总线通信接口, 寄生模式时供电端口
3	VDD	电源线 (2 线通信时接地, 以保证芯片内正确识别 VDD 状态)

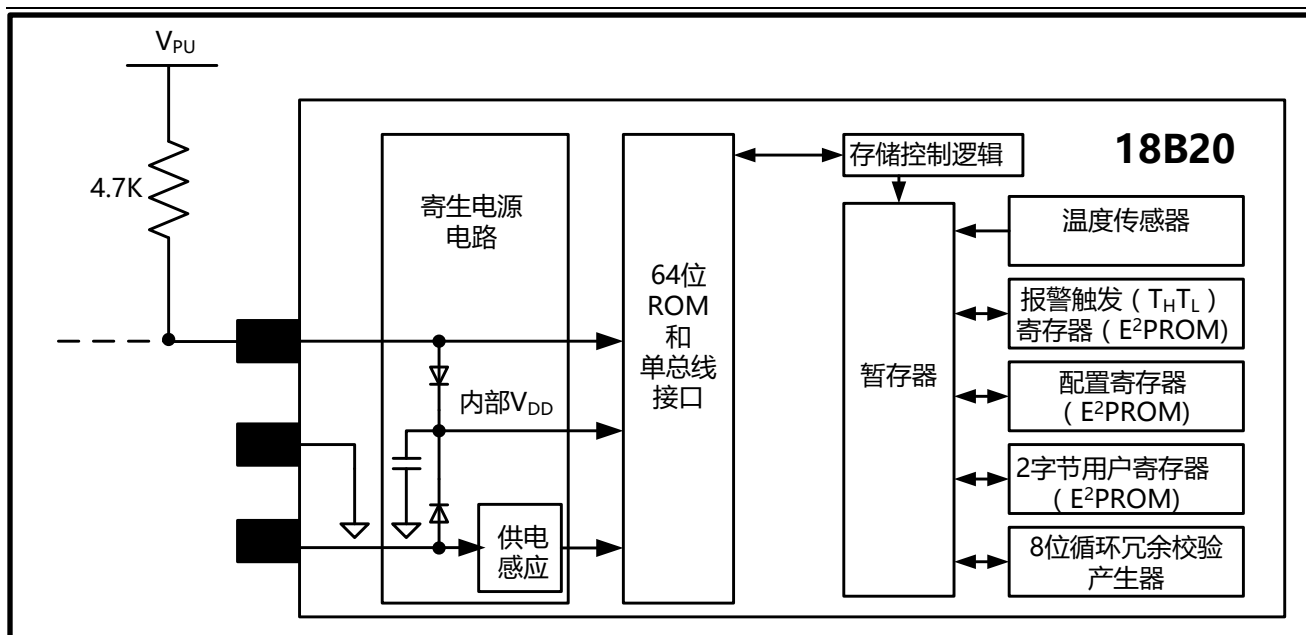


图 1 18B20 框图

结构框图

图 1 是 18B20 的原理框图。64 位 ROM 存储了器件的唯一序列码。暂存器包含了两个字节的温度寄存器，存储来自于温度传感器的数字输出。另外，暂存器提供了一高一低两个报警触发阈值寄存器 (TH 和 TL)。配置寄存器允许用户设定温度数字转换的分辨率为 9,10, 11 或 12 位。2 个字节的用户可编程 E2PROM 是非易失性存储，器件掉电时数据不会失去。

18B20 使用单总线协议，总线通讯通过一根控制信号线实现。控制线需要一个弱上拉电阻这样

所有的器件都通过三态或者开漏极端口（就是 18B20 的 DQ 引脚）连接到总线上。在这个总线系统中，单片机（主机）通过每个器件的唯一 64 位编码识别并寻址总线上的器件。因为每个器件都有唯一的编码，实际上挂在总线上并可以被寻址的设备数量是无限的。单总线协议，包括详细的指令与时隙描述在[单总线系统](#)章节有详细描述。

18B20 的另一个特点是其可以不需要额外供电运行。这种情况下供电是总线为高的时候，通过单总线在 DQ 引脚上的上拉电阻提供给器件的。总线高信号对一个内部电容充电，然后在总线低的时候，内部电容就会维持对器件供电。这种从单总线获取电源的方法被称为“寄生供电”。当然 18B20 也可以通过 VDD 由外部供电。

运行-测量温度

18B20 的核心功能是直接数字测温。温度传感器的分辨率是 9,10, 11,12 位用户可配置的，对

应的温度分度分别是 0.5°C, 0.25°C, 0.125°C, 和 0.0625°C。上电后的默认分辨率是 12 位。

18B20 在低功耗空闲状态下上电，要启动温度测量和模数转换，主机必须发出 Convert T [44h] 指令。转换之后，产生的温度数据被存储在暂存器

的 2 个字节的温度寄存器中然后 18B20 返回空闲状态。如果 18B20 从外部供电，主机可以在 Convert T [44h] 指令后发布“读时隙”指令(参考[单总线系统](#)章节)，然后 18B20 发回响应，若温度转换还在进行中为 0，若已完成温度转换为 1。如果 18B20 由寄生电源供电，则不能使用此通知技术，因为总线在整个温度转换期间必须被强上拉。寄生电源下对总线的要求，在[给 18B20 供电](#)章节中有详细解释。18B20 的温度输出数据是摄氏温度格式，对于华氏度的应用，必须使用查表或转换子程序。温度数据存储为 16 位符号扩展温度寄存器

中的二进制补码(见图 2)。符号位(S)指示温度为正或负：对于正数 $S = 0$ ，对于负数 $S = 1$ 。如果 18B20 配置为 12 位分辨率，温度寄存器中的所有位都将包含有效数据。

对于 11 位分辨率，位 0 未定义。对于 10 位分辨率，位 1 和 0 未定义，对于 9 位

分辨率位 2，1 和 0 未定义。表 1 给出了在 12 位分辨率转换条件下，数字输出数据的示例以及相应的温度读数

	ADDR	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
LS 字节	0	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
		bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
MS 字节	1	S	S	S	S	S	2^6	2^5	2^4

图 2 温度寄存器格式

温度(°C)	数字输出 (二进制)	数字输出 (16 进制)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

表 1 温度/数据对应关系

运行-报警信号

18B20 完成一次温度转换后，就将该温度值和用户定义的二进制补码报警触发值比较，该值存储在 T_H 和 T_L 寄存器中(见图 3)。符号位 S 表示该值是正还是负：正数 $S=0$ 负数 $S=1$ 。 T_H 和 T_L 寄存器是非易失性的(E²PROM)所以当设备掉电时数据会保持。 T_H 和 T_L 可以通过暂存器的第 2 和第 3

字节访问，详细请参考[存储](#)章节。

由于 T_H 和 T_L 是 8 位寄存器，因此在温度比较中只有只用到 11 到 4 位。如果测量到的温度小于等于 T_L 或大于等于 T_H ，则报警条件成立，18B20 内部会设立一个标志位。该标志位每次温度测量都会被更新，所以如果报警条件不再满足，标志位就会在下次温度转换之后被关闭。

主机可以通过发布一个 *Alarm Search[ECh]* 指令来检查所有挂在总线上的 18B20 的标志位状况。任何设立了标志位的 18B20 都会响应该指令，这样主机可以知道具体哪一个 18B20 进

入了报警条件。如果报警条件成立并且 T_H 或者 T_L 设置被更改，则应该再进行一次温度转换来验证报警条件。

	ADDR	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
高温报警 (TH)	2	S	2^6	2^5	2^4	2^3	2^2	2^1	2^0
		bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
低温报警 (TL)	3	S	2^6	2^5	2^4	2^3	2^2	2^1	2^0

图 3 T_H 和 T_L 寄存器格式

给 18B20 供电

18B20 可以通过 V_{DD} 引脚由外部电源供电，也可以工作在“寄生电源”模式，这个模式可以让 18B20 在没有本地外围供电的情况下仍然工作。寄生电源对于远程测温或者空间受限的应用非常有用。图 1 显示了 18B20 的寄生电源控制电路，这种情况下通过 DQ 脚在总线高的情况下从单总线“窃取”电源。窃取的电荷在总线高的情况下对 18B20 供电，同时一部分电荷存储在寄生电源电容上用于在总线低的情况下提供电源。当 18B20 工作在寄生电源模式下， V_{DD} 引脚必须连接到地。

在寄生电源模式下，只要满足时序和电压要求(参考[直流与交流电气特性](#))，单总线和寄生电源电容就可以为 18B20 提供足够的电流。但是当 18B20 执行温度转换或者从暂存器到 E^2PROM 复制数据时，运行电流可以高至 1.5 毫安。这个电流可能引起单总线的弱上拉电阻的压降超限，而且这个电流超过了寄生电源电容可以提供的范围。为了确保 18B20 有足够的供电电流，在任何发生温度转换或者数据从暂存器复制到 E^2PROM 的情况下，都有必要对单总线提供一个强上拉。这个强上拉可以如图 4 中所示，用 MOSFET 把总线直接拉

到电源轨来实现。单总线在发布完 *Convert T [44h]* 或者 *Copy Scratchpad [48h]* 指令后，必须在 $10\mu s$ (最多) 内切换到强上拉，并且总线在温度转换(t_{CONV})或数据传输($t_{WR} = 10ms$)的过程中必须被上拉保持高。当上拉被启动，单总线上不可以发生其他活动。

18B20 也可以用传统模式通过外部电源供电到 V_{DD} 引脚，如图 5 所示。这种模式的优点是不需要 MOSFET 上拉，而且单总线在温度转换过程中可以任意进行其他操作。

在温度大于 $+100^\circ C$ 时不建议使用寄生电源因为 18B20 的漏电流在这种温度下会很高，可能难以让芯片保持通信。对于类似这样极限温度的应用，强烈建议使用外部电源对 18B20 供电。

有些情况下总线主机可能不知道 18B20 是寄生电源供电还是外部电源供电。而主机需要这个信息以决定温度转换期间是否应该使用强上拉。为了得到这一信息，主机可以发布一个 *Skip ROM [CCh]* 指令后跟随 *Read Power Supply [B4h]* 指令，后跟一个“读时隙”。在读时隙期间，寄生电源 18B20 会把总线拉低，外部供电的 18B20 会让总线保持高。如果总线拉低，主机就知道它一定要在温度转换过程中对单总线保持强上拉。

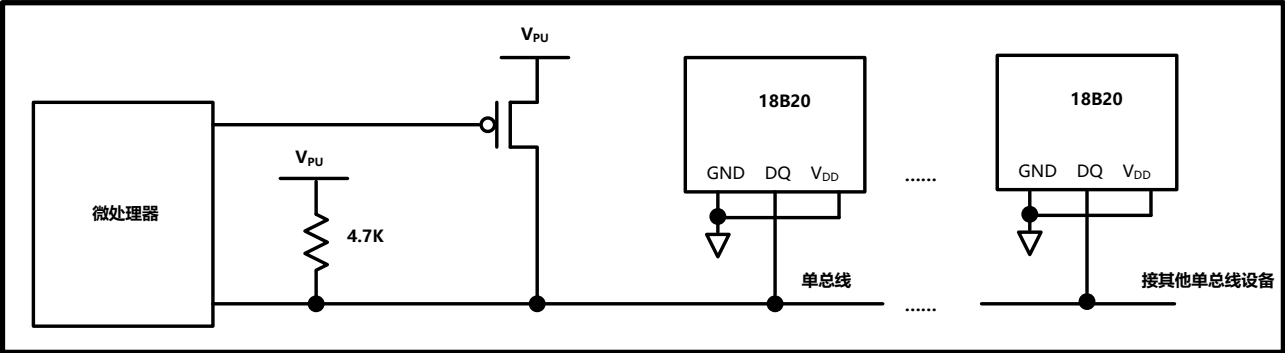


图 4 在 18B20 温度转换期间通过寄生电源供电

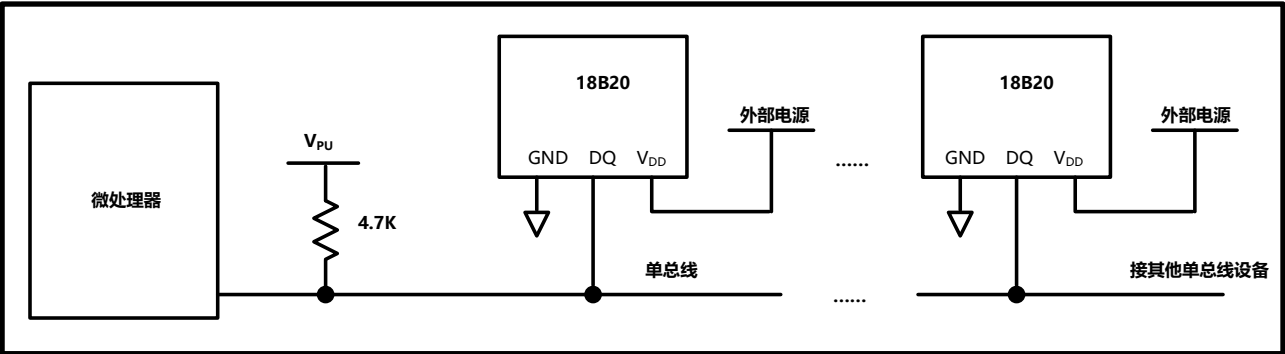


图 5 用外部电源对 18B20 供电

64 位 ROM 编码

每一个 18B20 都包含一个唯一的 64 位编码 (见图 6) 存储在 ROM 里。最低位 8 位 ROM 编

码包含了 18B20 的单总线系列代码 : 28h。接下来 48 位包含一个唯一的序列码。最高位 8 位包含了从前面 56 位 ROM 编码产生的循环冗余校验码。64 位 ROM 编码和相关的 ROM 功能控制逻辑使得 18B20 可以作为一个单总线器件使用单总线协议。单总线协议在[单总线系统](#)章节中有详述。

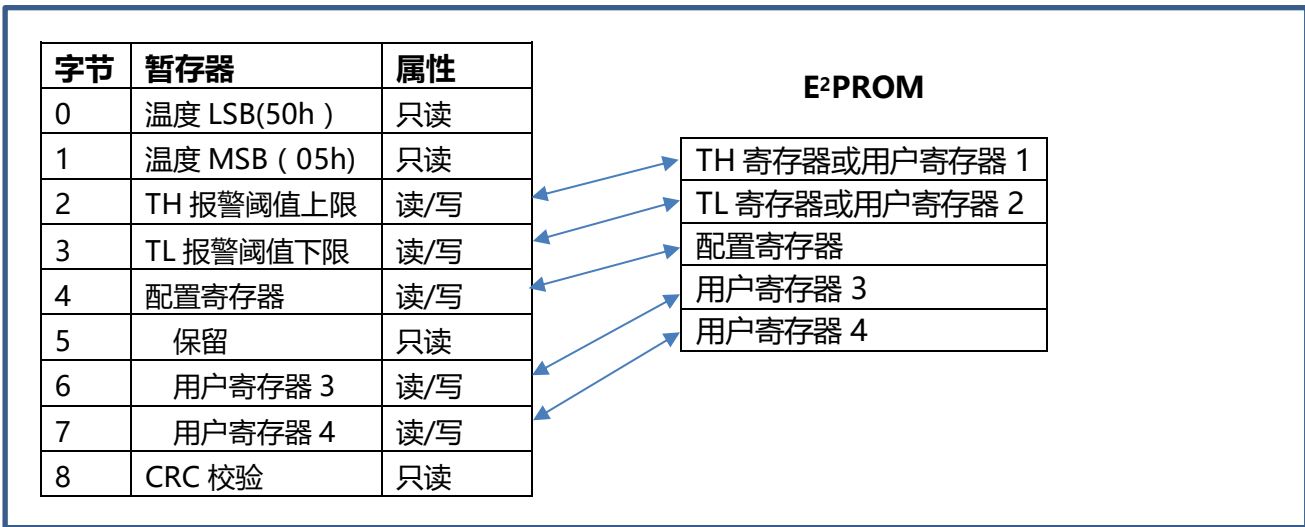
8 位循环冗余校验			48 位序列号			8 位产品系列码		
MSB		LSB	MSB		LSB	MSB		LSB

图 6 64 位激光 ROM 编码

存储

18B20 的存储组织如图 7 所示。存储器包含一个 SRAM 暂存器以及非易失性存储 E²PROM 寄存器，用于存储高低报警触发值（TH 和 TL），配置寄存器和 2 字节用户可编程 E²PROM。需要注意的是，如果 18B20 的报警功能没有被使用，则 TH 和 TL 寄存器可以用作通用存储。所有存储指令都在 [18B20 功能指令](#) 章节有详细描述。暂存器字节 0 和字节 1 是只读的。字节 2 和字节 3 用于访问 TH 和 TL 寄存器。字节 4 包含配置寄存器数据，在 [配置寄存器](#) 章节中有详细解释。字节 5 是内部用途不可以被改写。字节 6 和 7 是用户可以自由使用的。字

节 8 为只读寄存器，是字节 0 到 7 产生的循环冗余校验码。18B20 通过 [循环冗余校验码生成](#) 章节中所述的规则来产生这个码。数据可以通过 Write Scratchpad [4Eh] 指令写入到字节 2,3,4,5,6, 和 7 中。数据传输必须始于字节 2 的最低位。为了验证数据完整性，可以在数据写操作之后读取暂存器（通过 *Scratchpad [BEh]* 指令）。读取暂存器的时候，单总线的数据传输始于字节 0 的最低位。要把 TH, TL 和配置数据从暂存器传输到 E²PROM，主机必须发出 *Copy Scratchpad [48h]* 指令。E²PROM 寄存器中的数据掉电会保持，上电时会自动加载到相应的暂存器位置。数据也可以通过 *Recall E 2 [B8h]* 指令随时重新加载。主机可以在 *Recall E 2 [B8h]* 指令随后发送读时隙，然后 18B20 会返回状态，如果重新加载还在进行中返回 0，如果已经完成返回 1。



存储映射

配置寄存器

暂存器的字节 4 是配置寄存器，组织形式如图 8。用户可以通过设定表 2 中的 R0 和 R1 位来配置

18B20 的转换分辨率。上电默认这些位是 R0=1 和 R1=1(12 位分辨率)。请注意分辨率和转换时间是直接折衷关系(分辨率越高, 转换时间越长; 分辨率越低, 转换时间越短)。Bit7 和 bit0 到 4 是为器件内用途保留的不能够被重写。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	R1	R0	1	1	1	1	1

图 8 配置寄存器

R1	R0	分辨率	最大转换时间	
0	0	9-bit	93.75ms	$t_{CONV}/8$
0	0	10-bit	187.5ms	$t_{CONV}/4$
1	0	11-bit	375ms	$t_{CONV}/2$
1	1	12-bit	750ms	t_{CONV}

表 2 温度计分辨率配置

循环冗余校验码生成

循环冗余校验字节是 18B20 的 64 位 ROM 编码的一部分，位于暂存器的第 9 个字节。ROM 编码循环冗余校验是从 ROM 编码的前 56 位计算得来，存储在 ROM 的最高位。暂存器的循环冗余校验码是在暂存器内计算的来，因此随暂存器内数据改变而改变。循环冗余校验码给主机从暂存器读取数据提供了一个数据校验方法。要验证数据读取是否正确，主机必须自己通过接收到的数据进行计算并将此数据与 ROM 里的循环冗余校验码比较（来自于 ROM 读操作）或者与暂存器里的循环冗余校验码比较（来自于暂存器读操作）。如果计算的循环冗余校验码与读取的温和，数据就是正确无误的接收了。循环冗余校验的比较以及是否继续操

作都完全取决于主机。如果 18B20 的循环冗余校验（ROM 或暂存器）与总线主机计算的值不匹配，18B20 内没有电路阻止指令进一步执行。循环冗余校验码（ROM 或暂存器）的等效多项式函数是：

$$CRC = X^8 + X^5 + X^4 + 1$$

总线主机可以重新计算循环冗余校验码并与 18B20 产生的循环冗余校验数值比较，该值通过图 9 的多项式发生器产生。该电路包含一个移位寄存器和若干异或门，移位寄存器的各位都初始化为 0。从 ROM 数据的最低位或暂存器中字节 0 的最低位开始，每次移动一个位到移位寄存器。直到 ROM 的第 56 位或暂存器的字节 7 的最高位移完，多项式发生器里就会存储重新计算过的循环冗余校验。接下来，18B20 的 8 位 ROM 或者暂存器循环冗余校验数据必须被移入电路。到这里，如果重新计算的循环冗余校验是正确的，移位寄存器内应该全是 0。

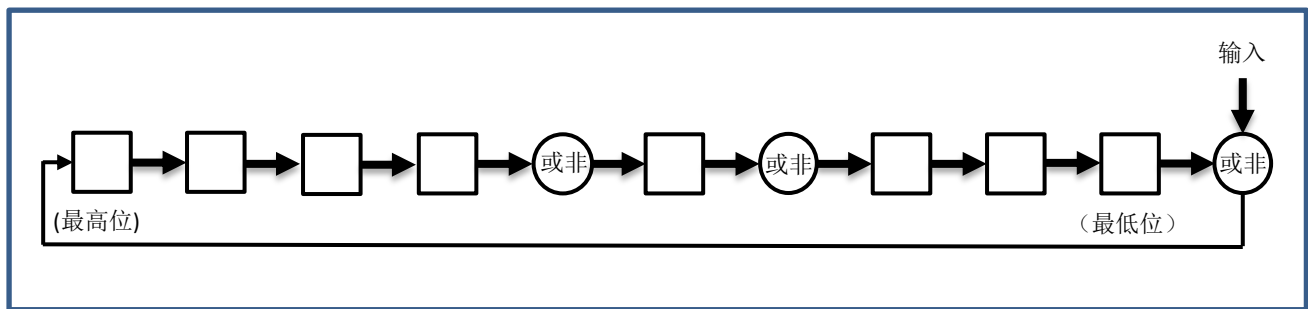


图 9 循环冗余校验生成器

单总线系统

单总线系统使用单一总线主机控制一个或多个从设备。18B20 永远位从设备。当总线上只有

一个从设备，系统被称为“单点”系统；当总线上有多个从设备，称为“多点”系统。所有数据和指令在单总线上传输都始于最低位。以下单总线系统的描述被分为三个题目：硬件配置，传输序列，以及单总线信令（信号类型和时序）

硬件配置

根据定义，单总线只有一根数据线。每个设备（主或从）通过开漏极或三态端口与数据线相连。这就可以允许设备在没有传输数据的时候“释放”数据线，因而总线可以给其他设备使用。

18B20 的单总线端口（DQ 引脚）内部等效电路是漏极开路，如图 10 所示。单总线要求一个大约 $5K\Omega$ 的外部上拉电阻；这样，单总线的空闲状态就是高电平。如果出于任何原因传输需要暂停，在传输未返回之前，总线必须保持空闲状态。在恢复过程中，只要单总线保持在非活动（高电平）状态，数据位之间的恢复时间可以无限长。如果总线被拉低超过 $480\mu s$ ，所有挂在总线上的器件将被复位。

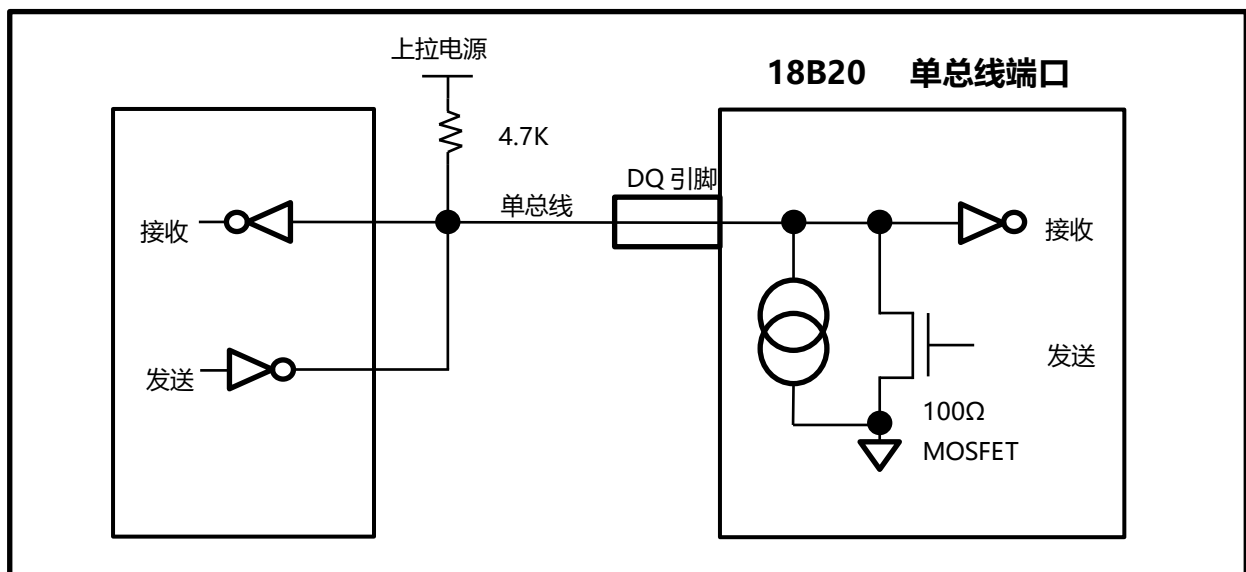


图 10 硬件配置

传输序列

访问 18B20 的传输序列如下：

步骤 1 初始化

步骤 2 ROM 指令（跟随任意必需的数据

交换）

步骤 3 18B20 功能指令（跟随任意必需的数据交换）

每次访问 18B20 都确保这个顺序非常重要，因为任何步骤缺失或者顺序不对都会导致

18B20 不响应。只有 Search ROM [F0h] 和 Alarm Search [ECh] 指令可以例外。当这些 ROM 指令发布之后，主机必须返回序列中的步骤 1。

初始化

通过单线总线的所有执行（处理）都从一个初始化序列开始。初始化序列包括一个由总线控制器

发出的复位脉冲和跟随其后由从机发出的存在脉冲。存在脉冲让总线主机知道从设备（如

18B20）在总线上且已准备好运行。复位与存在脉冲的时序在[单总线信令](#)章节详述。

ROM 操作指令

当总线主机检测到存在脉冲，就可以发布 ROM 指令。这些指令运作于每个从设备唯一的 64 位 ROM 编码而且如果有多个从设备挂在单总线上，主机可以单独寻址特定的从设备。这些指令也使得主机可以决定总线上有多少什么类型的设备，以及是否有任何设备满足了报警条件。共有 5 个 ROM 指令，每个 8 位长。主设备必须在发布

18B20 的功能指令之前发布一个合适的 ROM 指令。图 11 显示了 ROM 指令运行的流程图。

SEARCH ROM [F0h]

当一个系统初始化上电后，主机必须识别总线上所有从设备的 ROM 编码，这样主机才能决定从设备的数量和类型。主机通过排除过程来识别

ROM 编码，这一过程需要主机执行 Search ROM 循环（如，Search ROM 指令跟随数据交换）反复执行直到识别所有从设备。如果总线上只有一个从设备，则可以使用简单的 Read ROM（如下）指令代替 Search ROM 指令。每次 Search ROM 循环之后，总线主机必须返回传输序列的步骤 1（初始态）

Read ROM [33h]

这个指令允许总线主机读到 18B20 的 4 位 ROM 编码。只有在总线上存在单只 18B20 的时候才能使用这个指令。如果总线上有不止一个从机，当所有从机试图同时响应时就会发生数据冲突。

Match ROM [55h]

匹配 ROM 指令后跟 64 位 ROM 编码序列，让总线主机在多点或单点总线上寻址一个特定的

18B20。只有 64 位 ROM 编码序列完全匹配的 18B20 才会响应主机发出的功能指令。所有其他从机都将等待一个复位脉冲。

Skip ROM [CCh]

可以主机可以通过该指令同时寻址总线上所有设备而无需发送任何 ROM 编码。例如，主机可以令总线上所有 18B20 同时执行温度转换，只需发布 Skip ROM 指令跟随一个 Convert T [44h] 指令。

需要注意 Read Scratchpad [BEh] 指令只能在单一从设备挂在总线上时才能跟随 Skip ROM 指令。这种情况下，通过允许主机无需发送 64 位设备

18B20 功能指令

当总线主机使用一个 ROM 指令寻址一个它希望与之通讯的 18B20 之后，主机可以发布 18B20 的功能指令之一。这些指令允许主机从 18B20 的暂存器写或读数据，发动温度转换以及了解供电模式。18B20 的功能指令，如下所述，总结于表 3 并且由图 12 的流程图详细阐述。

CONVERT T [44h]

该指令发动一次温度转换。转换之后，采集的热数据存储在暂存器中的 2 字节的温度寄存器然后 18B20 返回低功耗空闲状态。如果设备使用于寄生电源模式下，本指令发布后最多 10 μ s 之内，主机必须在整个温度转换期间 (tCONV) 启动单总线的强上拉，如[给 18B20 供电](#)章节所述。如果 18B20 由外部电源供电，主机可以在 Convert T 指令后发布读时隙，然后 18B20 会回复 0 或者 1 表示温度转换正在进行中或者已完成。在寄生电源模式这个通知技术不能应用，因为总线在整个转换过程中被强上拉到高电平。

WRITE SCRATCHPAD [4Eh]

该指令允许主机对 18B20 暂存器写入最多 5 个数据。第一个字节被写入 TH 寄存器 (暂存器的字节 2)，第二个字节被写入 TL 寄存器 (字节 3)，第三个字节写入配置寄存器 (字节 4)，最后两个字节写入用户字节 3 和 4。数据必须先发送最低位。

ROM 编码而读取从设备，可以节约时间。如果总线上有超过一个从设备，一个 Skip ROM 指令跟随一个 Read Scratchpad 指令会导致数据冲突，因为多个设备会试图同时传送数据。

Alarm Search [ECh]

这条指令的流程图和 Search ROM 相同，然而，只有置位了报警标志位的 18B20 才会响应这条指令。本指令允许主机设备可以知道是否任何 18B20 在最近的温度转换达到了温度报警条件。每一次 Alarm Search 循环 (如，Alarm Search 指令跟随数据交换) 之后，总线主机必须返回传输序列的步骤 1 (初始化)。[运行-报警信号](#)章节解释了报警标志位的操作。

所有 5 个字节必须在主机发出复位信号前写入，否则数据可能损坏。(意味主机可以在任意时刻通过复位中止写入)

READ SCRATCHPAD [BEh]

该指令允许主机读取暂存器中的内容。数据传输始于字节 0 的最低位并延续遍历暂存器直到第 9 个字节 (字节 8-循环冗余验证码) 被读取。如果只需要暂存器中的部分数据，主机可以随时发布一个复位信号终止读取。

COPY SCRATCHPAD [48h]

该指令将暂存器中的 TH, TL, 配置寄存器和用户字节 3 和 4 (字节 2,3,4,5,6,7) 写入到 E2PROM。如果设备使用于寄生电源模式下，本指令发布后最多 10 μ s 之内，主机必须启动单总线的强上拉并保持最少 10ms，如[给 18B20 供电](#)章节所述。

RECALL E² [B8h]

该指令从 E2PROM 中调用报警触发值 (TH 和 TL)，配置寄存器和用户字节 4 和 5，并替换暂存器中字节 2,3,4,6 和 7 中对应的数据。主机设备可以跟随 Recall E² 指令之后发布一个读时隙，然后 18B20 会指示调用的状态，传送 0 表示调用正在进行中传送 1 表示调用已经结束。调用操作上电时自动执行，所以设备上电之后暂存器中就立即具备有效数据。

READ POWER SUPPLY [B4h]

主机发布本指令跟随一个读时隙以了解总线上是否有任何 18B20 在使用寄生电源供电。在读

时隙期间,寄生电源供电的 18B20 会拉低总线,而外部电源供电的 18B20 会让总线保持高。请参考[给 18B20 供电](#)章节获取对本指令有用的信息。

指令	描述	协议	指令发出后单总线的活动	备注
温度转换指令				
Convert T	发动温度转换	44h	18B20 传输转换状态给主机 (对寄生电源供电的 18B20 不适用)	1
存储指令				
Read Scratchpad	读包含循环冗余校验在内的所有暂存器内容	BEh	18B20 传输最多 9 个字节给主机	2
Write Scratchpad	写数据到暂存器的字节 2, 3, 和 6, 7 (TH, TL, 配置寄存器和用户字节)	4Eh	主机传输 3 或 4 或 5 个字节数据给 18B20	3
Copy Scratchpad	从暂存器复制 TH, TL, 配置寄存器和用户字节数据到 E ² PROM	48h	无	1
Recall E ²	从 E ² PROM 调用 TH, TL, 配置寄存器和用户字节数据到暂存器	B8h	18B20 传输调用状态给主机	
Read Power Supply	发送 18B20 的供电模式给主机	B4h	18B20 传输供电状态给主机	

表 3 18B20 功能指令集

备注 1: 对于寄生电源供电的 18B20, 主机必须在整个温度转换期间对单总线启动强上拉并将数据从暂存器复制到 E²PROM。在此期间总线不会发生任何活动。

备注 2: 主机可以通过发布一个复位信号随时打断数据传输。

备注 3: 所有字节必须在复位信号发布之前写完。

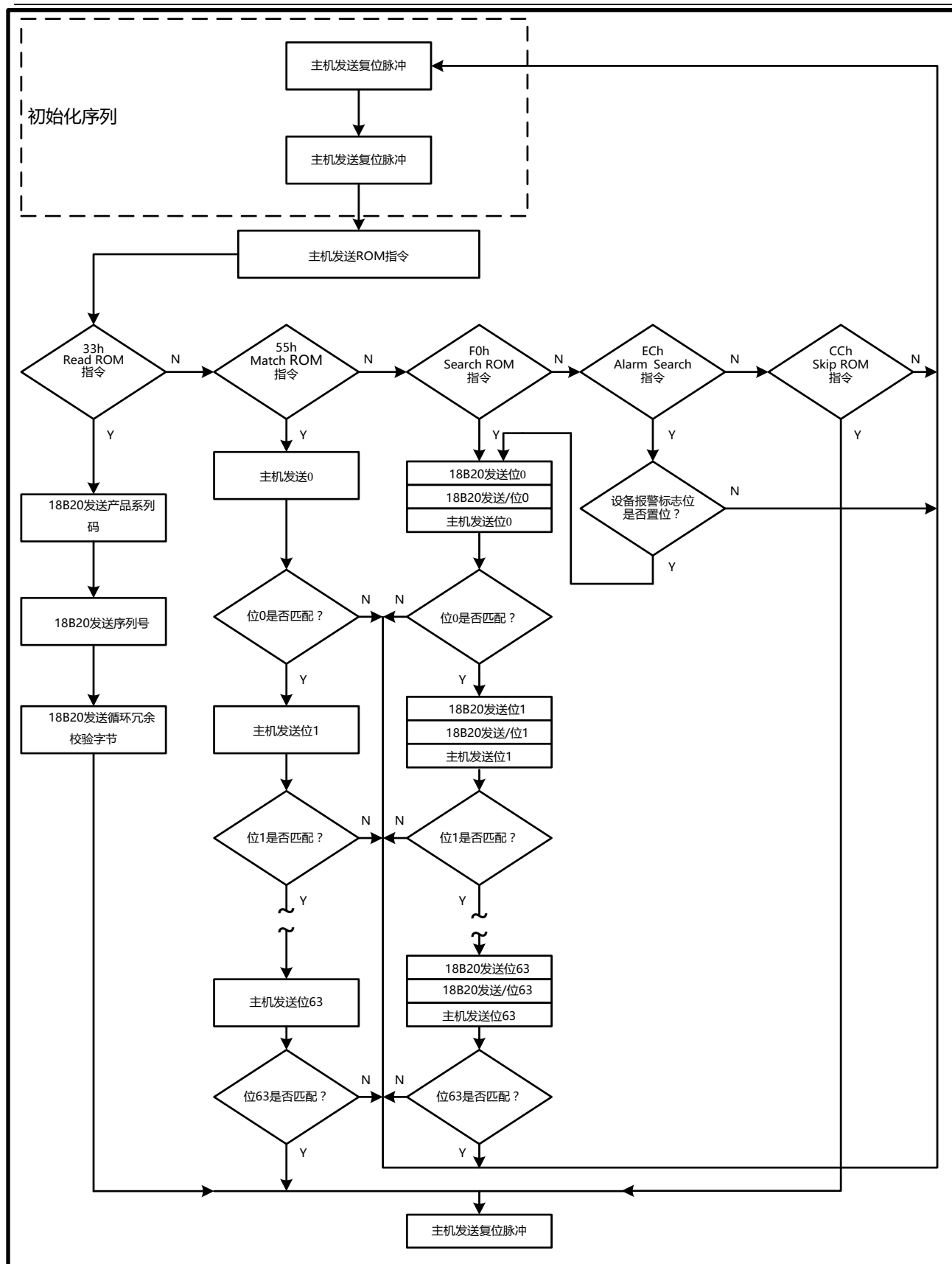


图 11 ROM 指令流程图

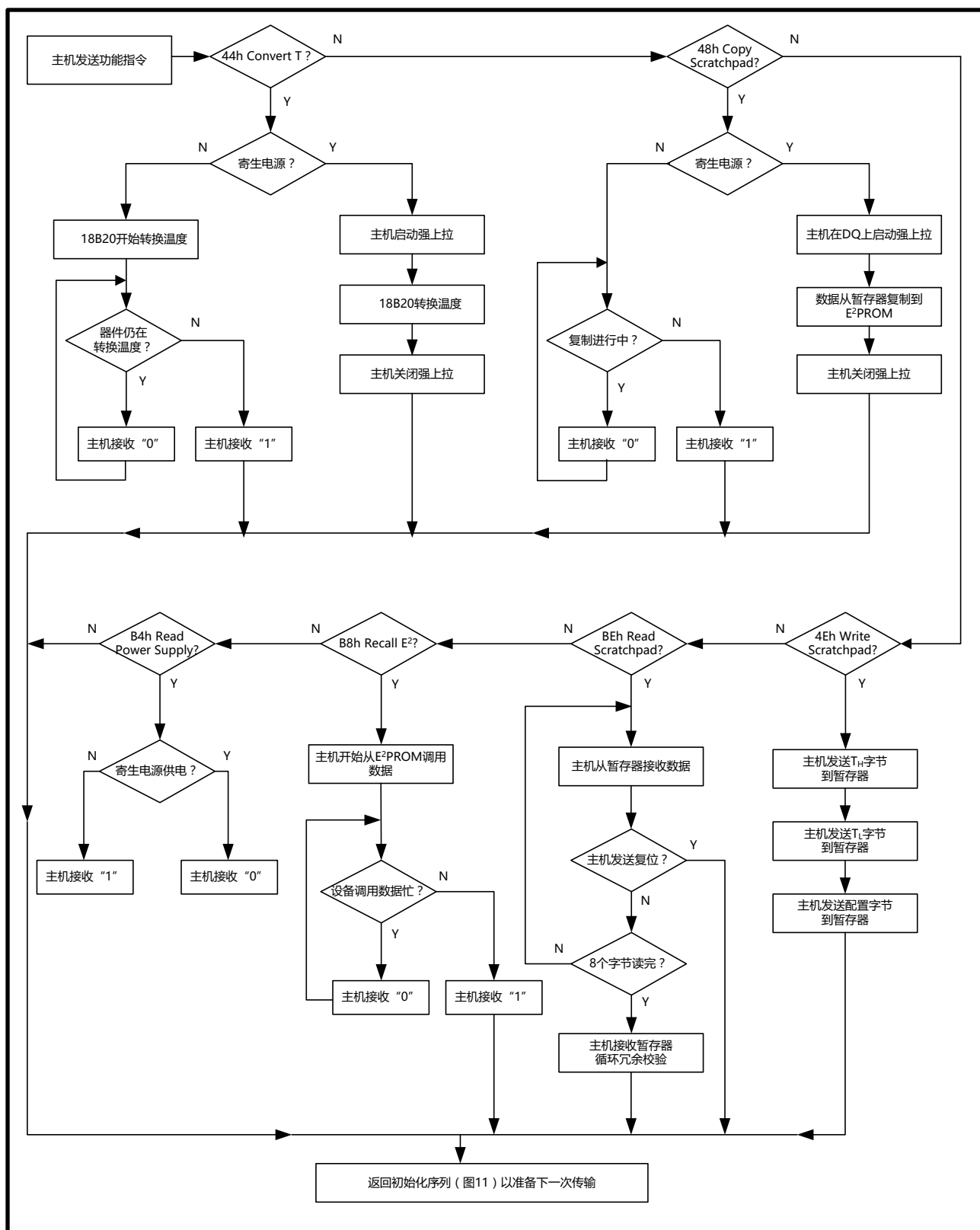


图 12 18B20 功能指令流程图

单总线信令

18B20 使用严格的单总线通讯协议以确保数据完整性。该协议定义了几种信令类型：复位脉冲，存在脉冲，写 0，写 1，读 0，读 1。除了存在脉冲之外，所有信令都由总线主机发起。

初始化——复位与存在脉冲

所有与 18B20 的通讯都始于初始化序列，这个序列包含主机发出的一个复位脉冲，跟随一个 18B20 发出的存在脉冲。图 13 解释了这一序列。当 18B20 发送存在脉冲以响应复位脉冲，其向主机表明它挂在总线上，并且已经准备好运行。在初始化序列过程中，主机通过将单总线拉低至少 $480\mu\text{s}$ 来发出复位脉冲。总线主机随后释放总线进入接收模式。当总线被释放后， $5\text{k}\Omega$ 上拉电阻会把总线拉高。当 18B20 检测到这个上升沿，它等待 $15\mu\text{s}$ 到 $60\mu\text{s}$ 然后通过把单总线拉低 $60\mu\text{s}$ 到 $240\mu\text{s}$ 来发出存在脉冲。

读/写时隙

总线主机在写时隙写数据到 18B20，在读时隙从 18B20 读数据。每个时隙在单总线上传输一个数据位。

写时隙

有两种写时隙：“写 1”时隙和“写 0”时隙。总线主机通过写 1 时隙把一个逻辑 1 写入 18B20，通过写 0 时隙把一个逻辑 0 写入 18B20。所有写时隙必须持续最少 $60\mu\text{s}$ ，并且两个写时隙之间至少有 $1\mu\text{s}$ 的恢复时间。两种写时隙都是通过主机把

单总线拉低来发起（见图 14）。

要产生写 1 时隙，把单总线拉低之后，总线主机必须在 $15\mu\text{s}$ 内释放单总线。总线被释放后， $5\text{k}\Omega$ 上拉电阻会把总线拉高。要产生一个写 0 时隙，把单总线拉低之后，总线主机必须在整个时隙期间持续保持总线低（至少 $60\mu\text{s}$ ）。

18B20 在主机发起写时隙后，会在至少 $15\mu\text{s}$ 到 $60\mu\text{s}$ 的时间窗口内采样单总线。如果在这个采样时间窗口总线为高，一个 1 就被写入 18B20。如果总线是低，一个 0 会被写入 18B20。

读时隙

18B20 只能在主机发布读时隙期间可以传送数据到主机。所有主机在发布 Read Scratchpad [BEh] 或 Read Power Supply [B4h] 指令后，必须立即产生读时隙，这样 18B20 才能提供所要求的数据。另外，主机可以在发布 Convert T[44h] 或 Recall E2 [B8h] 指令后产生读时隙以了解运行状态。这部分机理在 [18B20 功能指令](#) 章节有详细解释。所有读时隙必须持续至少 $60\mu\text{s}$ ，并且两个写时隙之间恢复时间不少于 $1\mu\text{s}$ 。读时隙的产生是通过主机拉低单总线至少 $1\mu\text{s}$ 然后释放总线来实现（见图 14）。主机发起读时隙之后，18B20 会开始在总线上传输 1 或 0。18B20 通过保持总线高发送 1 并通过拉低总线发送 0。当传输 0 的时候，

18B20 会在时隙结束时释放总线，之后总线会被上拉电阻拉回高空闲状态。18B20 的输出数据在启动时隙的下降沿后 $15\mu\text{s}$ 之内有效。所以，主机必须在时隙启动之后 $15\mu\text{s}$ 之内释放总线并采样总线状态。图 15 说明了在一个读时隙内 t_{INIT} ， t_{RC} 和 t_{SAMPLE} 的总和必须少于 $15\mu\text{s}$ 。图 16 显示了系统的时间裕度可以通过以下方法最大化：保持 t_{INT} 和 t_{RC} 越短越好，以及把主机采样时间放到读时隙 $15\mu\text{s}$ 周期的末尾。

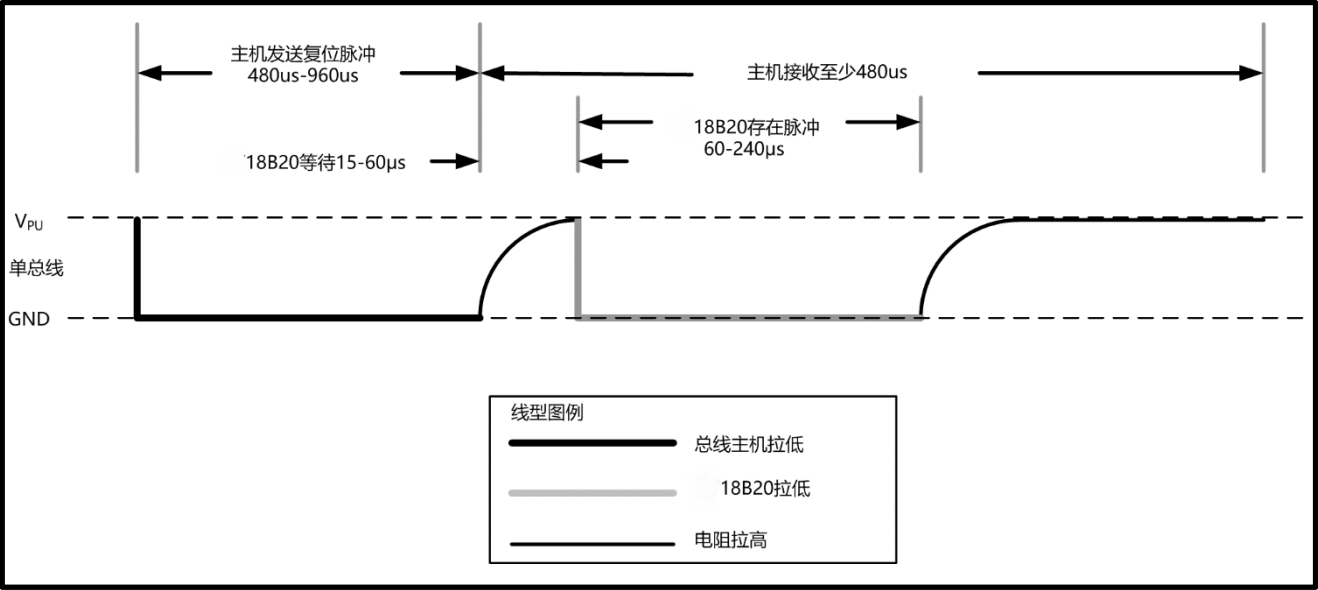


图 13 初始化时序

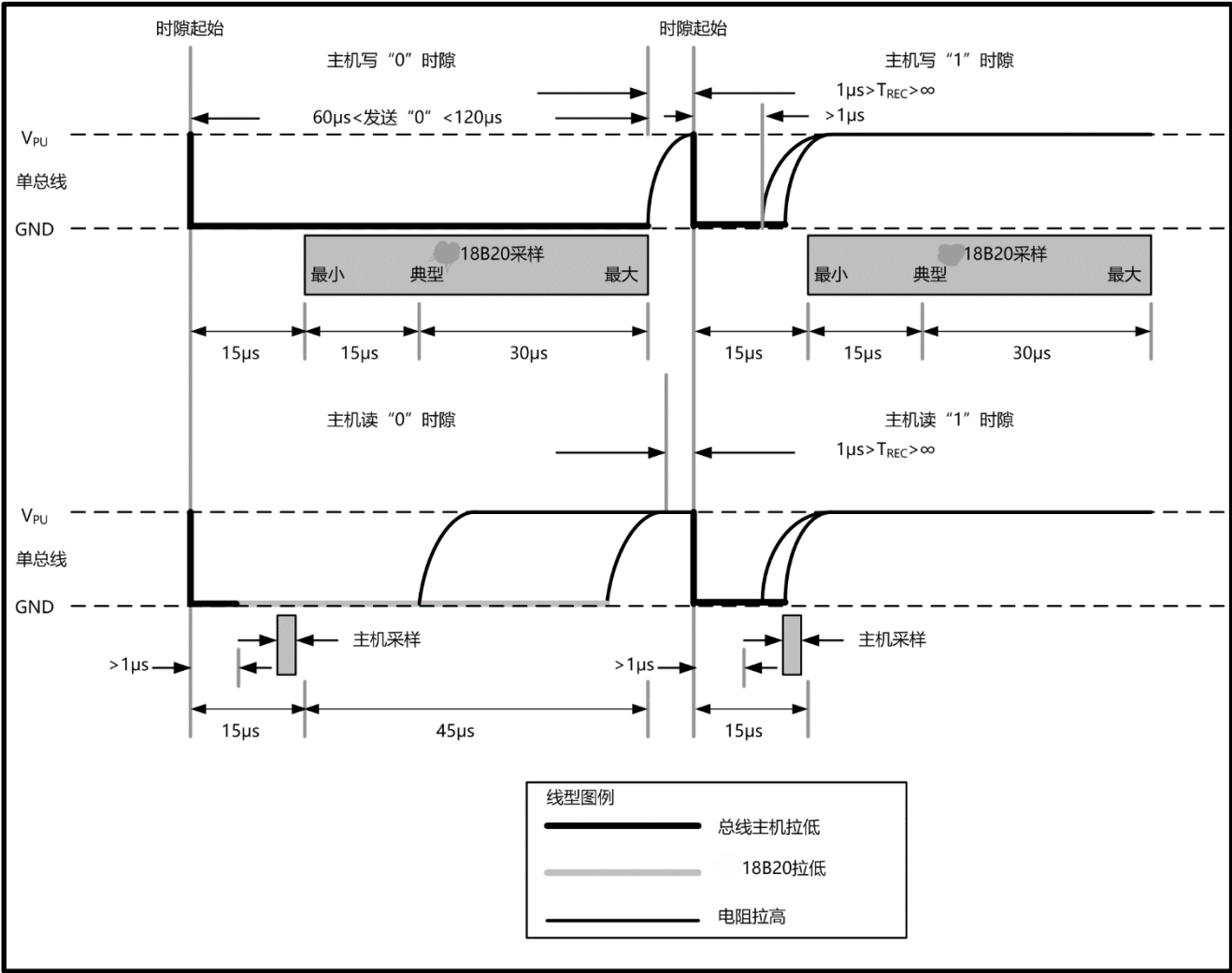


图 14 读/写时隙时序

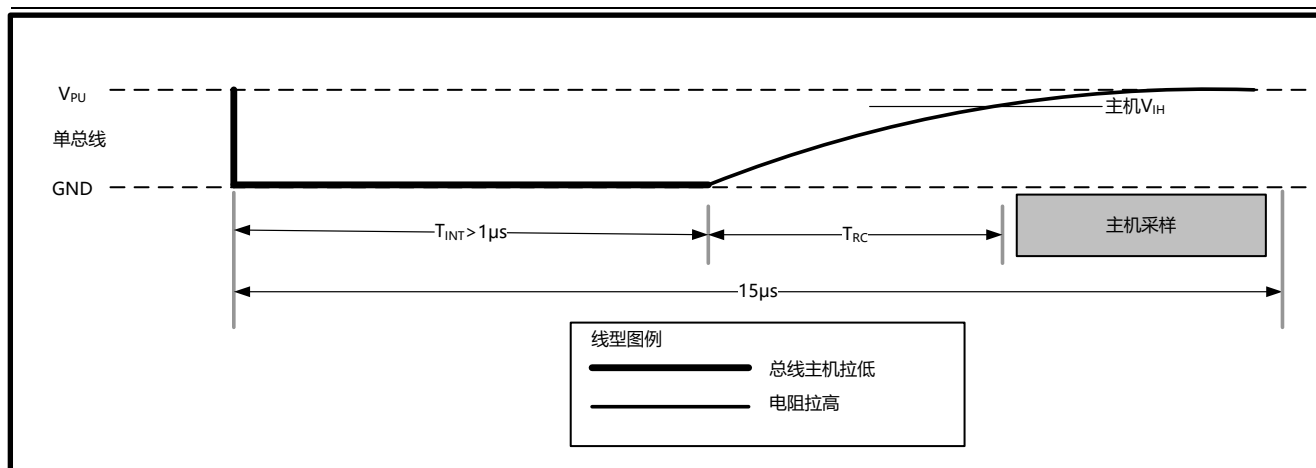


图 15 详细主机读 1 时序

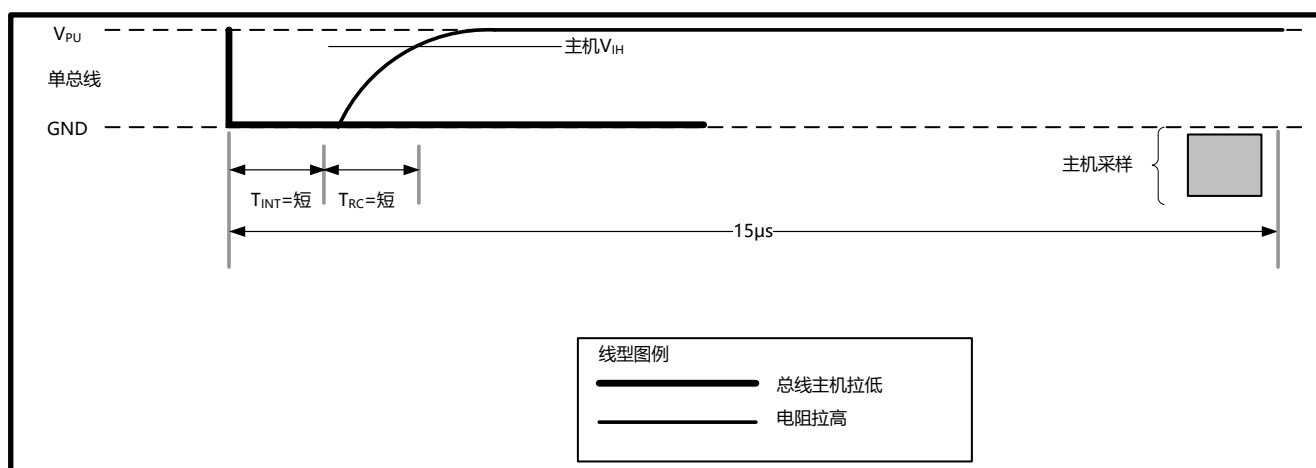


图 16 推荐的主机读 1 时序

使用寄生电源。总线主机对某个特定的 18B20 启动温度转换之后读取其缓存器然后重新计算循环冗余校验来验证数据。

18B20 运行示例 1

在这个实例中总线上有多个 18B20 并且在

主机模式	数据（最低位在前）	指令
发送	复位	主机发布复位脉冲
接收	存在	18B20 通过存在脉冲响应
发送	55h	主机发送 Match ROM 指令
发送	64 位 ROM 编码	主机发送 18B20 的 ROM 编码
发送	44h	主机发布 Convert T 指令
发送	DQ 线由强上拉保持高	主机在转换期间（ t_{CONV} ）对 DQ 实施强上拉
发送	复位	主机发布复位脉冲
接收	存在	18B 20 通过存在脉冲响应
发送	55h	主机发布 Match ROM 指令
发送	64 位 ROM 编码	主机发送 18B20 的 ROM 编码
发送	BEh	主机发布 Read Scratchpad 指令
接收	9 个数据字节	主机读包含循环冗余校验在内整个暂存器。然后主机重算暂存器的前 8 个字节数据的循环冗余校验并与读取的循环冗余校验（字节 9）进行比较。如果匹配，主机继续；否则，重复整个读操作

18B20 运行示例 2

用寄生电源。主机写入 TH, TL 以及配置数据到 18B20 的暂存器中,然后读暂存器并且重新计算循环冗余校验以验证数据。之后主机把暂存器的内容复制到 E2PROM。

在这个示例里总线上只有一个 18B20 并且使

主机模式	数据 (最低位在前)	指令
发送	复位	主机发布复位脉冲
接收	存在	18B20 通过存在脉冲响应
发送	CCh	主机发布 Skip ROM 指令
发送	4Eh	主机发布 Write Scratchpad 指令
发送	3 个数据字节	主机发送 3 个数据字节到暂存器 (TH, TL 和配置)
发送	复位	主机发布复位脉冲
接收	存在	18B20 通过存在脉冲响应
发送	CCh	主机发布 Skip ROM 指令
发送	BEh	主机发布 Read Scratchpad 指令
接收	9 个数据字节	主机读包含循环冗余校验在内整个暂存器。然后主机重算暂存器的前 8 个字节数据的循环冗余校验并与读取的循环冗余校验 (字节 9) 进行比较。如果匹配, 主机继续; 否则, 重复整个读操作
发送	复位	主机发布复位脉冲
接收	存在	18B20 通过存在脉冲响应
发送	CCh	主机发布 Skip ROM 指令
发送	48h	主机发布 Copy Scratchpad 指令
发送	DQ 线由强上拉保持高	主机在复制操作期间对 DQ 实施至少 10ms 强上拉

电气特性

绝对最大额定值

任意引脚到地电压值	-0.5V to +6.0V
工作温度范围	-55°C to +125°C
存储温度范围	-55°C to +125°C
焊接温度	参考 IPC/JEDEC J-STD-020 规范

这些仅为极限参数下，对于器件在此极限条件或高于此极限条件的环境中的功能运行，本规格书并不适用。长期暴露于此极限环境会影响器件的可靠性。

直流电气特性

-55°C到+125°C； $V_{DD}=3.0V$ 到 $5.5V$

参数	符号	条件	最小	典型	最大	单位	备注
供电电压	V_{DD}	本地供电	+3.0		+5.5	V	1
上拉电压	V_{PU}	寄生电源	+3.0		+5.5	V	1,2
		本地电源	+3.0		V_{DD}		
温度误差	t_{ERR}	-10°C 到 +85°C			± 0.5	°C	3
		-55°C 到 +125°C			± 2		
输入逻辑低	V_{IL}					V	1,4,5
输入逻辑高	V_{IH}	本地电源	+2.2			V	1,6
		寄生电源	+3.0				
灌电流	I_L	$V_{I/O}=0.4V$	4.0			mA	1
待机电流	I_{DDs}			750	1000	nA	7,8
有功电流	I_{DD}	$V_{DD}=5V$		1	1.5	mA	9
DQ 输入电流	I_{DQ}			5		μA	10
漂移				± 0.2		°C	11

备注：

- 1) 所有电压以地为参考。
- 2) 上拉供电电压规格假设上拉器件为理想器件，因而上拉的高电平等于 V_{PU} 。为了满足 18B20 的 V_{IH} 规范，实际的强上拉供电轨必须考虑到开启时上拉电阻的电压降裕度，所以 $V_{PU-ACTUAL} = V_{PU_IDEAL} + V_{TRANSISTOR}$ 。
- 3) 参考图 17 的典型特性曲线。
- 4) 逻辑低规范条件为 4mA 灌电流。
- 5) 为了确保低电压寄生电源供电的存在脉冲， V_{ILMAX} 可能需要减小至 0.5V。
- 6) 逻辑高规范条件为 1mA 源电流。
- 7) 待机电流规范条件最高到 +70°C。+125°C 条件下的待机电流典型值为 3 μA 。
- 8) 为了最小化 I_{DDs} ，DQ 应该在以下范围内： $GND \leq DQ \leq GND + 0.3V$ 或 $V_{DD} - 0.3V \leq DQ \leq V_{DD}$ 。
- 9) 有功电流指有效的温度转换或 E^2PROM 写操作期间的供电电流。
- 10) DQ 线是高（高阻态）
- 11) 漂移数据基于 1000 小时压力测试，条件为 +125°C， $V_{DD}=5.5V$ 。

交流电气特性-非易失性存储器

-55°C到+125°C ; $V_{DD}=3.0V$ 到 5.5V

参数	符号	条件	最低	典型	最大	单位
非易失存储写周期	t_{WR}			2	10	ms
E ² PROM 写次数	N_{EEWR}	-55°C 到 +55°C	50000			次
E ² PROM 数据保留	t_{EEDR}	-55°C 到 +55°C	10			年

交流电气特性

-55°C到+125°C ; $V_{DD}=3.0V$ 到 5.5V

参数	符号	条件	最小	典型	最大	单位	备注
温度转化时间	t_{CONV}	9 位分辨率			93.75	ms	1
		10 位分辨率			187.5		
		11 位分辨率			375		
		12 位分辨率			750		
强上拉时间	t_{SPON}	从 ConvT T 指令发布起			10	μs	
时隙长度	t_{SLOT}		60		120	μs	1
恢复时间	t_{REC}		1			μs	1
写 0 低电平时间	t_{LOW0}		60		120	μs	1
写 1 低电平时间	t_{LOW1}		1		15	μs	1
读数据有效时间	t_{RDV}				15	μs	1
复位高电平时间	t_{RSTH}		480			μs	1
复位低电平时间	t_{RSTL}		480			μs	1,2
存在检测高电平时间	t_{PDHIGH}		15		60	μs	1
存在检测低电平时间	t_{PDLOW}		60		240	μs	1
电容	$C_{IN/OUT}$				25	pF	

备注：

- 1) 参考图 18 的时序图。
- 2) 在寄生电源模式下，如果 $t_{RSTL} > 960 \mu s$ ，可能发生上电复位。

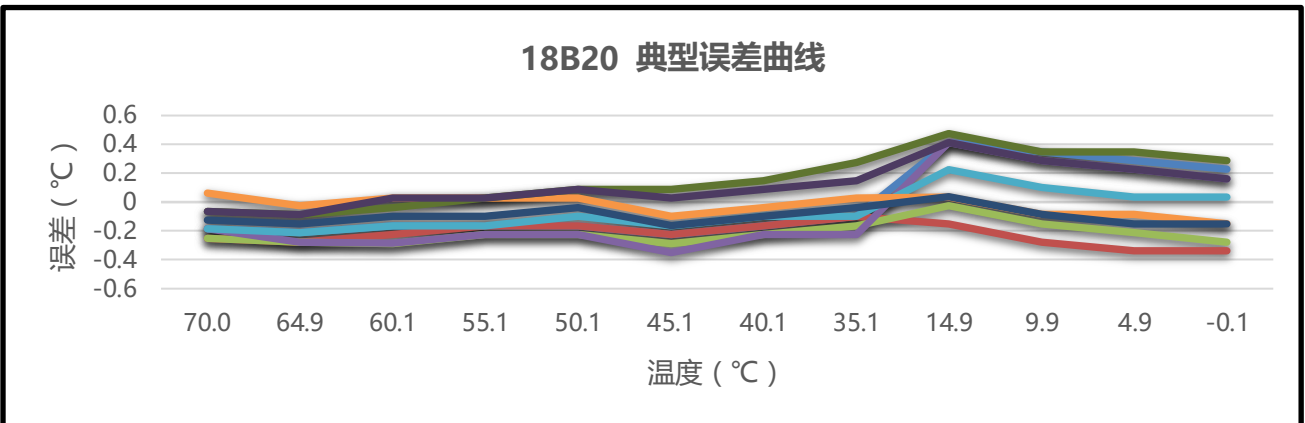
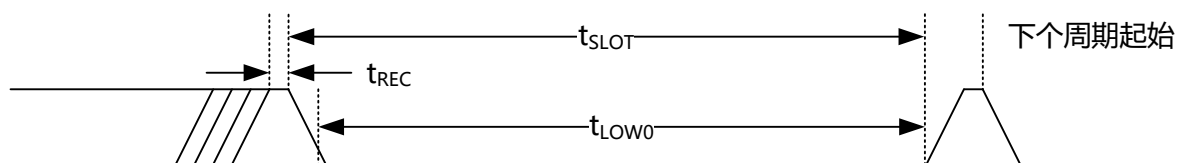
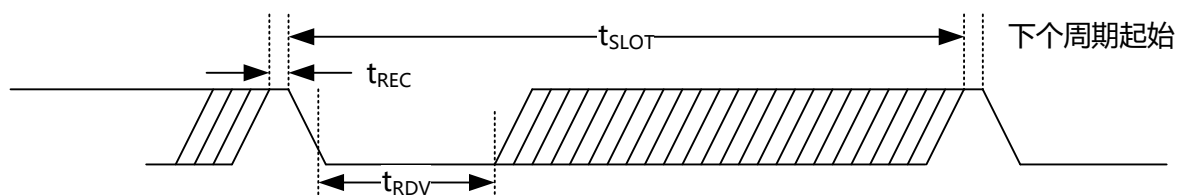


图 17 18B20 典型误差曲线

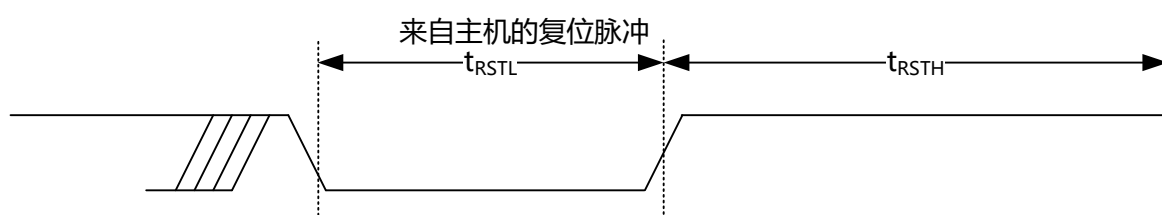
单总线写0时序



单总线读0时序



单总线复位脉冲



单总线存在检测

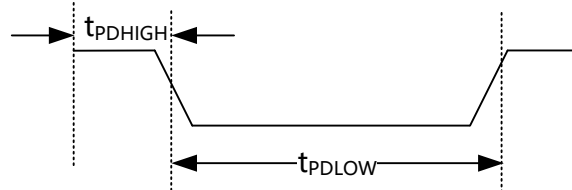


图 18 时序图

版本号	描述	编者	时间
2.0	整体功能介绍	HXY	2016.12.01