

# DALLAS SEMICONDUCTOR

## DS1820 1-Wire™ Digital Thermometer

### FEATURES

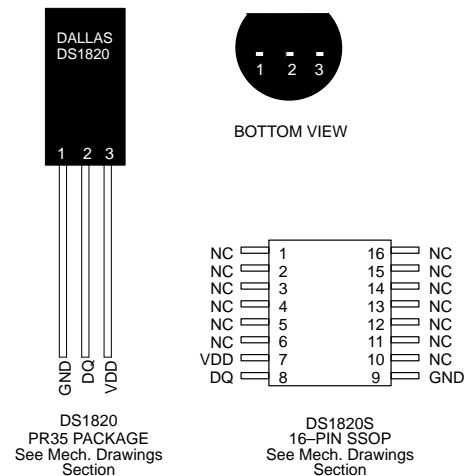
- Unique 1-Wire™ interface requires only one port pin for communication
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line
- Zero standby power required
- Measures temperatures from  $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  in  $0.5^{\circ}\text{C}$  increments. Fahrenheit equivalent is  $-67^{\circ}\text{F}$  to  $+257^{\circ}\text{F}$  in  $0.9^{\circ}\text{F}$  increments
- Temperature is read as a 9-bit digital value.
- Converts temperature to digital word in 200 ms (typ.)
- User-definable, nonvolatile temperature alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

### DESCRIPTION

The DS1820 Digital Thermometer provides 9-bit temperature readings which indicate the temperature of the device.

Information is sent to/from the DS1820 over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor to a DS1820. Power for reading, writing, and performing temperature conversions can be derived from the data line itself with no need for an external power source.

### PIN ASSIGNMENT



### PIN DESCRIPTION

GND	–	Ground
DQ	–	Data In/Out
V <sub>DD</sub>	–	Optional V <sub>DD</sub>
NC	–	No Connect

Because each DS1820 contains a unique silicon serial number, multiple DS1820s can exist on the same 1-Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings, equipment or machinery, and in process monitoring and control.

**DETAILED PIN DESCRIPTION**

PIN 16-PIN SSOP	PIN PR35	SYMBOL	DESCRIPTION
9	1	GND	<b>Ground.</b>
8	2	DQ	<b>Data Input/Output pin.</b> For 1-Wire operation: Open drain. (See "Parasite Power" section.)
7	3	V <sub>DD</sub>	<b>Optional V<sub>DD</sub> pin.</b> See "Parasite Power" section for details of connection.

DS1820S (16-pin SSOP): All pins not specified in this table are not to be connected.

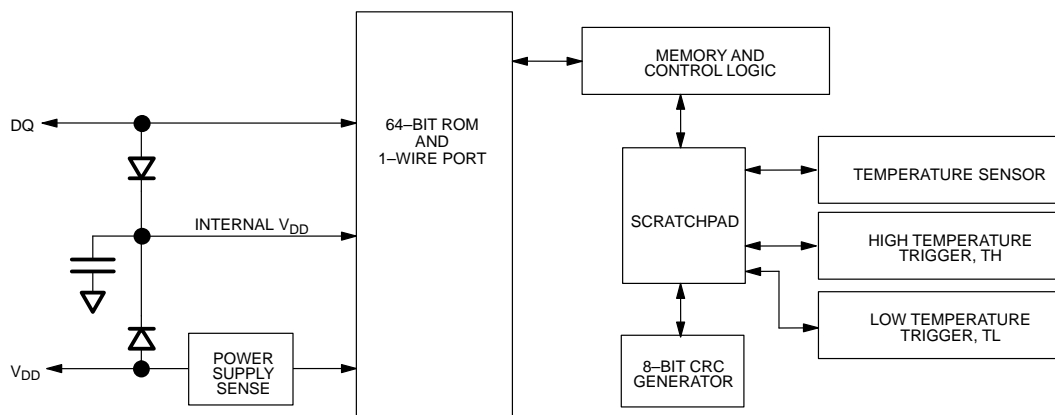
**OVERVIEW**

The block diagram of Figure 1 shows the major components of the DS1820. The DS1820 has three main data components: 1) 64-bit lasered ROM, 2) temperature sensor, and 3) nonvolatile temperature alarm triggers TH and TL. The device derives its power from the 1-Wire communication line by storing energy on an internal capacitor during periods of time when the signal line is high and continues to operate off this power source during the low times of the 1-Wire line until it returns high to replenish the parasite (capacitor) supply. As an alternative, the DS1820 may also be powered from an external 5 volts supply.

Communication to the DS1820 is via a 1-Wire port. With the 1-Wire port, the memory and control functions will not be available before the ROM function protocol has been established. The master must first provide one of five ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, or 5) Alarm Search. These commands operate on the 64-bit lasered ROM portion of each device and can single out

a specific device if many are present on the 1-Wire line as well as indicate to the Bus Master how many and what types of devices are present. After a ROM function sequence has been successfully executed, the memory and control functions are accessible and the master may then provide any one of the six memory and control function commands.

One control function command instructs the DS1820 to perform a temperature measurement. The result of this measurement will be placed in the DS1820's scratchpad memory, and may be read by issuing a memory function command which reads the contents of the scratchpad memory. The temperature alarm triggers TH and TL consist of one byte EEPROM each. If the alarm search command is not applied to the DS1820, these registers may be used as general purpose user memory. Writing TH and TL is done using a memory function command. Read access to these registers is through the scratchpad. All data is read and written least significant bit first.

**DS1820 BLOCK DIAGRAM** Figure 1

### PARASITE POWER

The block diagram (Figure 1) shows the parasite powered circuitry. This circuitry “steals” power whenever the I/O or  $V_{DD}$  pins are high. I/O will provide sufficient power as long as the specified timing and voltage requirements are met (see the section titled “1-Wire Bus System”). The advantages of parasite power are two-fold: 1) by parasiting off this pin, no local power source is needed for remote sensing of temperature, and 2) the ROM may be read in absence of normal power.

In order for the DS1820 to be able to perform accurate temperature conversions, sufficient power must be provided over the I/O line when a temperature conversion is taking place. Since the operating current of the DS1820 is up to 1 mA, the I/O line will not have sufficient drive due to the 5K pull-up resistor. This problem is particularly acute if several DS1820's are on the same I/O and attempting to convert simultaneously.

There are two ways to assure that the DS1820 has sufficient supply current during its active conversion cycle. The first is to provide a strong pull-up on the I/O line whenever temperature conversions or copies to the  $E^2$  memory are taking place. This may be accomplished by using a MOSFET to pull the I/O line directly to the power supply as shown in Figure 2. The I/O line must be switched over to the strong pull-up within 10  $\mu$ s maximum after issuing any protocol that involves copying to the  $E^2$  memory or initiates temperature conversions. When using the parasite power mode, the  $V_{DD}$  pin must be tied to ground.

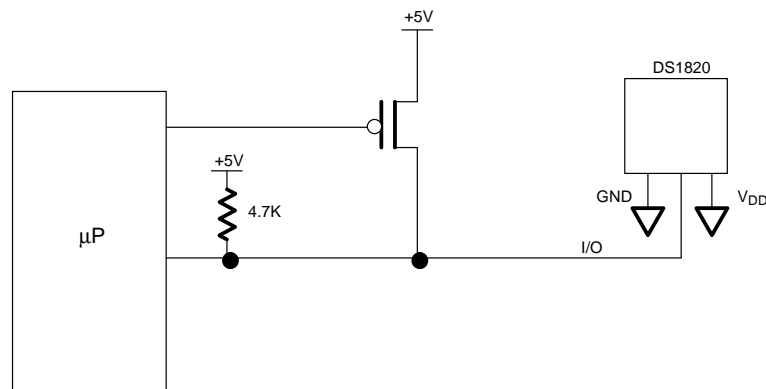
Another method of supplying current to the DS1820 is through the use of an external power supply tied to the

$V_{DD}$  pin, as shown in Figure 3. The advantage to this is that the strong pull-up is not required on the I/O line, and the bus master need not be tied up holding that line high during temperature conversions. This allows other data traffic on the 1-Wire bus during the conversion time. In addition, any number of DS1820's may be placed on the 1-Wire bus, and if they all use external power, they may all simultaneously perform temperature conversions by issuing the Skip ROM command and then issuing the Convert T command. Note that as long as the external power supply is active, the GND pin may not be floating.

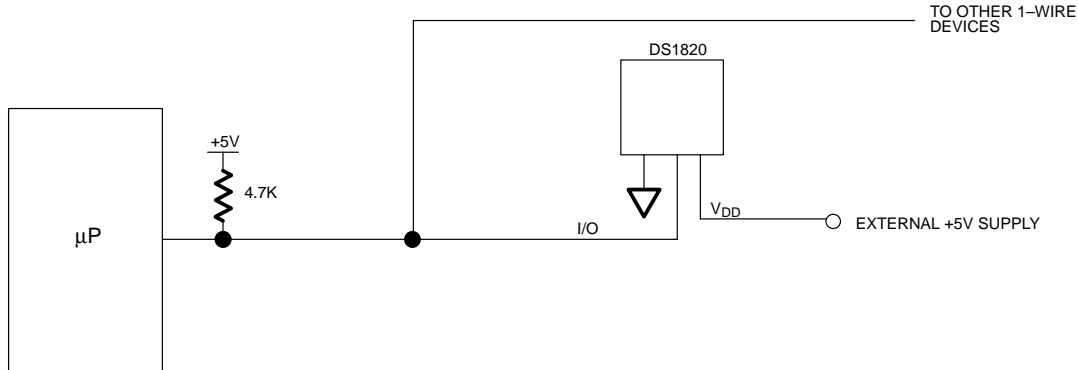
The use of parasite power is not recommended above 100°C, since it may not be able to sustain communications given the higher leakage currents the DS1820 exhibits at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that  $V_{DD}$  be applied to the DS1820.

For situations where the bus master does not know whether the DS1820's on the bus are parasite powered or supplied with external  $V_{DD}$ , a provision is made in the DS1820 to signal the power supply scheme used. The bus master can determine if any DS1820's are on the bus which require the strong pull-up by sending a Skip ROM protocol, then issuing the read power supply command. After this command is issued, the master then issues read time slots. The DS1820 will send back “0” on the 1-Wire bus if it is parasite powered; it will send back a “1” if it is powered from the  $V_{DD}$  pin. If the master receives a “0”, it knows that it must supply the strong pull-up on the I/O line during temperature conversions. See “Memory Command Functions” section for more detail on this command protocol.

### STRONG PULL-UP FOR SUPPLYING DS1820 DURING TEMPERATURE CONVERSION Figure 2



### USING $V_{DD}$ TO SUPPLY TEMPERATURE CONVERSION CURRENT Figure 3



#### OPERATION – MEASURING TEMPERATURE

The DS1820 measures temperature through the use of an on-board proprietary temperature measurement technique. A block diagram of the temperature measurement circuitry is shown in Figure 4.

The DS1820 measures temperature by counting the number of clock cycles that an oscillator with a low temperature coefficient goes through during a gate period determined by a high temperature coefficient oscillator. The counter is preset with a base count that corresponds to  $-55^{\circ}\text{C}$ . If the counter reaches zero before the gate period is over, the temperature register, which is also preset to the  $-55^{\circ}\text{C}$  value, is incremented, indicating that the temperature is higher than  $-55^{\circ}\text{C}$ .

At the same time, the counter is then preset with a value determined by the slope accumulator circuitry. This circuitry is needed to compensate for the parabolic behavior of the oscillators over temperature. The counter is then clocked again until it reaches zero. If the gate period is still not finished, then this process repeats.

The slope accumulator is used to compensate for the non-linear behavior of the oscillators over temperature, yielding a high resolution temperature measurement. This is done by changing the number of counts necessary for the counter to go through for each incremental degree in temperature. To obtain the desired resolution, therefore, both the value of the counter and the number of counts per degree C (the value of the slope accumulator) at a given temperature must be known.

Internally, this calculation is done inside the DS1820 to provide  $0.5^{\circ}\text{C}$  resolution. The temperature reading is

provided in a 16-bit, sign-extended two's complement reading. Table 1 describes the exact relationship of output data to measured temperature. The data is transmitted serially over the 1-Wire interface. The DS1820 can measure temperature over the range of  $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  in  $0.5^{\circ}\text{C}$  increments. For Fahrenheit usage, a lookup table or conversion factor must be used.

Note that temperature is represented in the DS1820 in terms of a  $1/2^{\circ}\text{C}$  LSB, yielding the following 9-bit format:

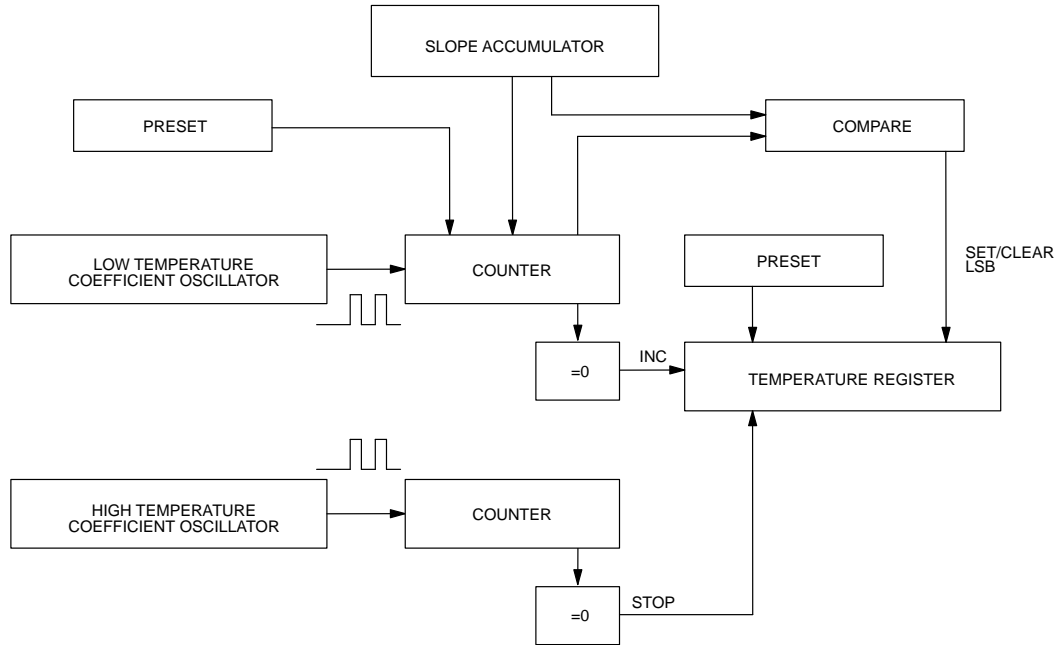
MSB								LSB
1	1	1	0	0	1	1	1	0

=  $-25^{\circ}\text{C}$

The most significant (sign) bit is duplicated into all of the bits in the upper MSB of the two-byte temperature register in memory. This "sign-extension" yields the 16-bit temperature readings as shown in Table 1.

Higher resolutions may be obtained by the following procedure. First, read the temperature, and truncate the  $0.5^{\circ}\text{C}$  bit (the LSB) from the read value. This value is TEMP\_READ. The value left in the counter may then be read. This value is the count remaining (COUNT\_REMAIN) after the gate period has ceased. The last value needed is the number of counts per degree C (COUNT\_PER\_C) at that temperature. The actual temperature may be then be calculated by the user using the following:

$$\text{TEMPERATURE} = \text{TEMP\_READ} - 0.25 + \frac{(\text{COUNT\_PER\_C} - \text{COUNT\_REMAIN})}{\text{COUNT\_PER\_C}}$$

**TEMPERATURE MEASURING CIRCUITRY** Figure 4

**TEMPERATURE/DATA RELATIONSHIPS** Table 1

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	00000000 11111010	00FA
+25°C	00000000 00110010	0032h
+1/2°C	00000000 00000001	0001h
+0°C	00000000 00000000	0000h
-1/2°C	11111111 11111111	FFFFh
-25°C	11111111 11001110	FFCEh
-55°C	11111111 10010010	FF92h

**OPERATION – ALARM SIGNALING**

After the DS1820 has performed a temperature conversion, the temperature value is compared to the trigger values stored in TH and TL. Since these registers are 8-bit only, the 0.5°C bit is ignored for comparison. The most significant bit of TH or TL directly corresponds to the sign bit of the 16-bit temperature register. If the result of a temperature measurement is higher than TH or lower than TL, an alarm flag inside the device is set.

This flag is updated with every temperature measurement. As long as the alarm flag is set, the DS1820 will respond to the alarm search command. This allows many DS1820s to be connected in parallel doing simultaneous temperature measurements. If somewhere the temperature exceeds the limits, the alarming device(s) can be identified and read immediately without having to read non-alarming devices.

### 64-BIT LASERED ROM

Each DS1820 contains a unique ROM code that is 64-bits long. The first eight bits are a 1-Wire family code (DS1820 code is 10h). The next 48 bits are a unique serial number. The last eight bits are a CRC of the first 56 bits. (See Figure 5.) The 64-bit ROM and ROM Function Control section allow the DS1820 to operate as a 1-Wire device and follow the 1-Wire protocol detailed in the section "1-Wire Bus System". The functions required to control sections of the DS1820 are not accessible until the ROM function protocol has been satisfied. This protocol is described in the ROM function protocol flowchart (Figure 6). The 1-Wire bus master must first provide one of five ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, or 5) Alarm Search. After a ROM functions sequence has been successfully executed, the functions specific to the DS1820 are accessible and the bus master may then provide one of the six memory and control function commands.

### CRC GENERATION

The DS1820 has an 8-bit CRC stored in the most significant byte of the 64-bit ROM. The bus master can compute a CRC value from the first 56-bits of the 64-bit ROM and compare it to the value stored within the DS1820 to determine if the ROM data has been received error-free by the bus master. The equivalent polynomial function of this CRC is:

$$\text{CRC} = X^8 + X^5 + X^4 + 1$$

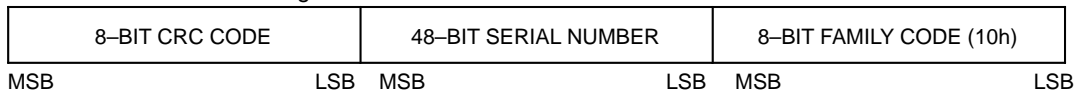
The DS1820 also generates an 8-bit CRC value using the same polynomial function shown above and pro-

vides this value to the bus master to validate the transfer of data bytes. In each case where a CRC is used for data transfer validation, the bus master must calculate a CRC value using the polynomial function given above and compare the calculated value to either the 8-bit CRC value stored in the 64-bit ROM portion of the DS1820 (for ROM reads) or the 8-bit CRC value computed within the DS1820 (which is read as a ninth byte when the scratchpad is read). The comparison of CRC values and decision to continue with an operation are determined entirely by the bus master. There is no circuitry inside the DS1820 that prevents a command sequence from proceeding if the CRC stored in or calculated by the DS1820 does not match the value generated by the bus master.

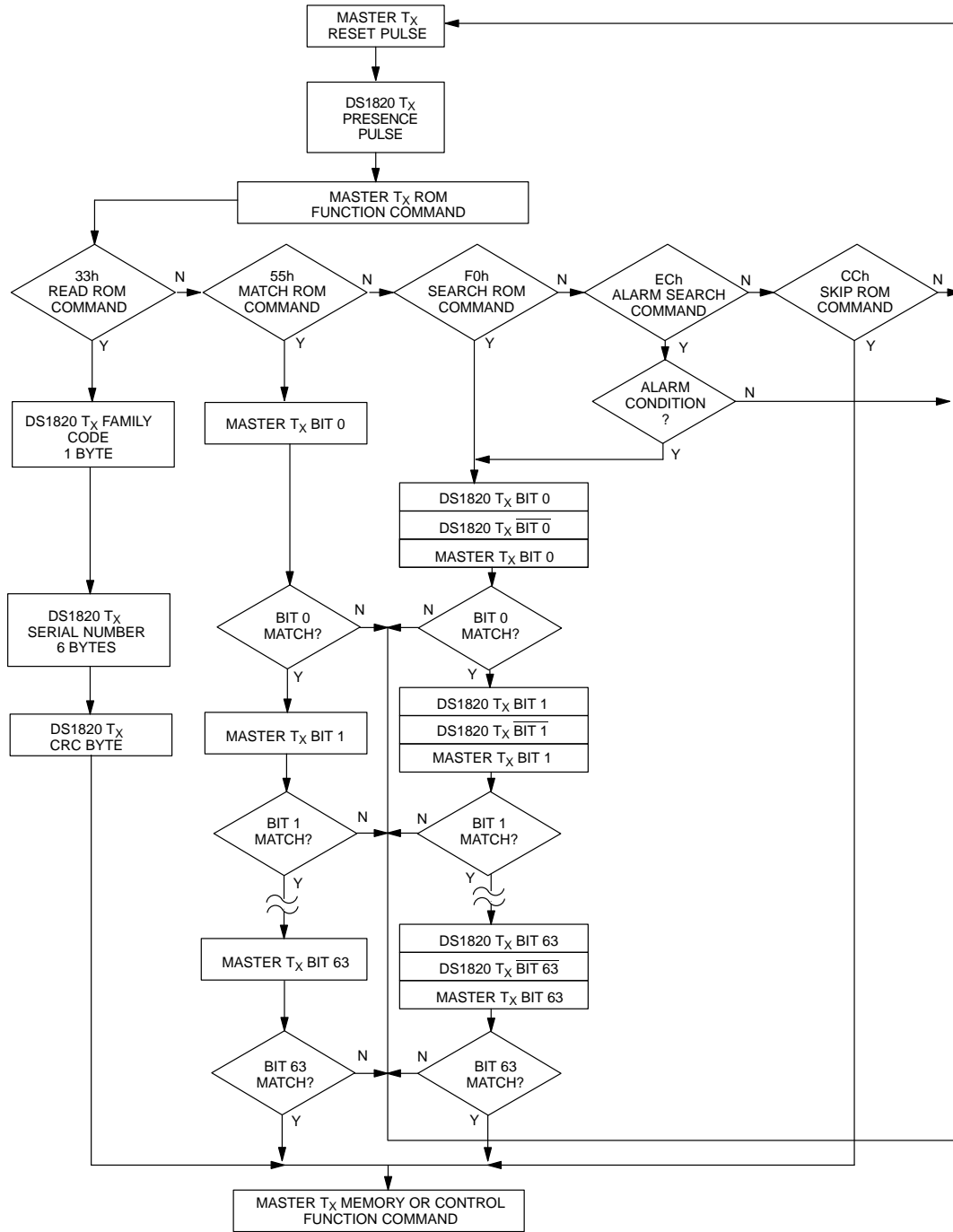
The 1-Wire CRC can be generated using a polynomial generator consisting of a shift register and XOR gates as shown in Figure 7. Additional information about the Dallas 1-Wire Cyclic Redundancy Check is available in Application Note 27 entitled "Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor Touch Memory Products".

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.

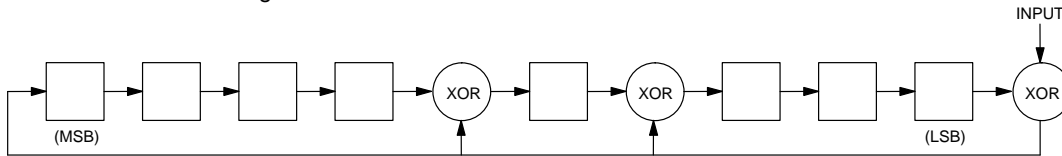
64-BIT LASERED ROM Figure 5



ROM FUNCTIONS FLOW CHART Figure 6



**1-WIRE CRC CODE** Figure 7



**MEMORY**

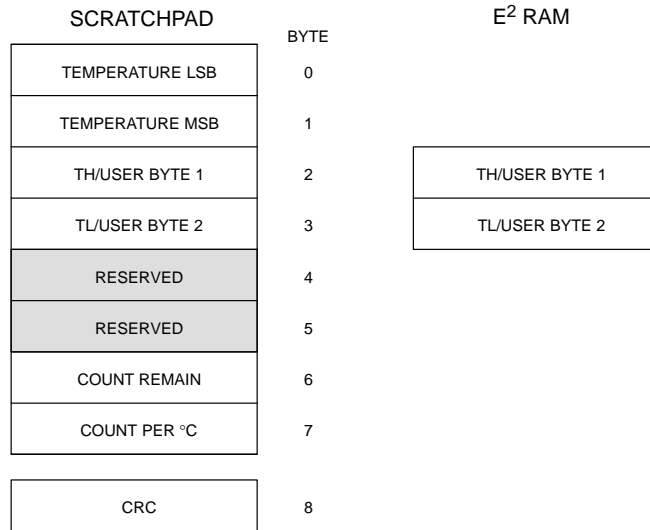
The DS1820's memory is organized as shown in Figure 8. The memory consists of a scratchpad RAM and a nonvolatile, electrically erasable (E<sup>2</sup>) RAM, which stores the high and low temperature triggers TH and TL. The scratchpad helps insure data integrity when communicating over the 1-Wire bus. Data is first written to the scratchpad where it can be read back. After the data has been verified, a copy scratchpad command will transfer the data to the nonvolatile (E<sup>2</sup>) RAM. This process insures data integrity when modifying the memory.

The scratchpad is organized as eight bytes of memory. The first two bytes contain the measured temperature

information. The third and fourth bytes are volatile copies of TH and TL and are refreshed with every power-on reset. The next two bytes are not used; upon reading back, however, they will appear as all logic 1's. The seventh and eighth bytes are count registers, which may be used in obtaining higher temperature resolution (see "Operation—measuring Temperature" section).

There is a ninth byte which may be read with a Read Scratchpad command. This byte contains a cyclic redundancy check (CRC) byte which is the CRC over all of the eight previous bytes. This CRC is implemented in the fashion described in the section titled "CRC Generation".

**DS1820 MEMORY MAP** Figure 8





## 1-WIRE BUS SYSTEM

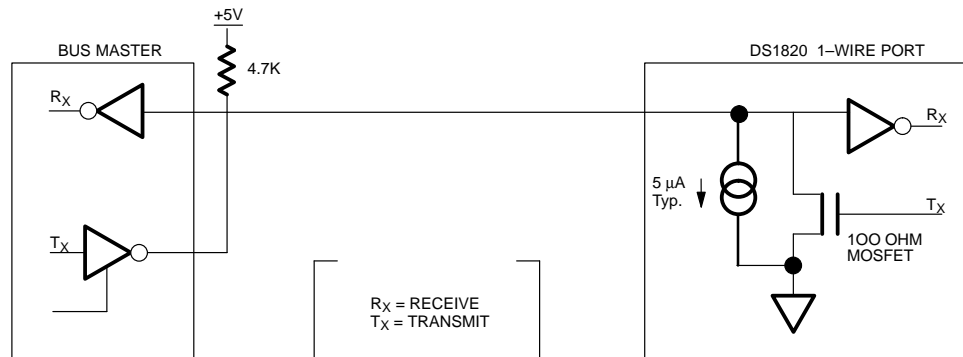
The 1-Wire bus is a system which has a single bus master and one or more slaves. The DS1820 behaves as a slave. The discussion of this bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing).

## 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 DS1820 (I/O pin) is open drain with an internal circuit equivalent to that shown in Figure 9. A multidrop bus consists of a 1-Wire bus with multiple slaves attached. The 1-Wire bus requires a pullup resistor of approximately 5K $\Omega$ .

## HARDWARE CONFIGURATION Figure 9



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. Infinite recovery time can occur between bits so long as the 1-Wire bus is in the inactive (high) state during the recovery period. If this does not occur and the bus is left low for more than 480  $\mu$ s, all components on the bus will be reset.

## TRANSACTION SEQUENCE

The protocol for accessing the DS1820 via the 1-Wire port is as follows:

- Initialization
- ROM Function Command
- Memory 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 DS1820 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 five ROM function commands. All ROM function commands are 8-bits long. A list of these commands follows (refer to flowchart in Figure 6):

**Read ROM [33h]**

This command allows the bus master to read the DS1820's 8-bit family code, unique 48-bit serial number, and 8-bit CRC. This command can only be used if there is a single DS1820 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).

**Match ROM [55h]**

The match ROM command, followed by a 64-bit ROM sequence, allows the bus master to address a specific DS1820 on a multidrop bus. Only the DS1820 that exactly matches the 64-bit ROM sequence will respond to the following memory function command. All slaves that do not match the 64-bit ROM sequence will wait for a reset pulse. This command can be used with a single or multiple devices on the bus.

**Skip ROM [CCh]**

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

**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 ROM codes. The search ROM command allows the bus master to use a process of elimination to identify the 64-bit ROM codes of all slave devices on the bus.

**Alarm Search [ECh]**

The flowchart of this command is identical to the Search ROM command. However, the DS1820 will respond to this command only if an alarm condition has been encountered at the last temperature measurement. An alarm condition is defined as a temperature higher than TH or lower than TL. The alarm condition remains set as long as the DS1820 is powered up, or until another temperature measurement reveals a non-alarming value. For alarming, the trigger values stored in EEPROM are taken into account. If an alarm condition exists and the TH or TL settings are changed, another temperature

conversion should be done to validate any alarm conditions.

**Example of a ROM Search**

The ROM search 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 simple, 3-step routine on each bit of the ROM. After one complete pass, the bus master knows the contents of the ROM in one device. The remaining number of devices and their ROM codes may be identified by additional passes.

The following example of the ROM search process assumes four different devices are connected to the same 1-Wire bus. The ROM data of the four devices is as shown:

```
ROM1  00110101...
ROM2  10101010...
ROM3  11110101...
ROM4  00010001...
```

The search process is as follows:

1. The bus master begins the initialization sequence by issuing a reset pulse. The slave devices respond by issuing simultaneous presence pulses.
2. The bus master will then issue the Search ROM command on the 1-Wire bus.
3. The bus master reads a bit from the 1-Wire bus. Each device will respond by placing the value of the first bit of their respective ROM data onto the 1-Wire bus. ROM1 and ROM4 will place a 0 onto the 1-Wire bus, i.e., pull it low. ROM2 and ROM3 will place a 1 onto the 1-Wire bus by allowing the line to stay high. The result is the logical AND of all devices on the line, therefore the bus master sees a 0. The bus master reads another bit. Since the Search ROM data command is being executed, all of the devices on the 1-Wire bus respond to this second read by placing the complement of the first bit of their respective ROM data onto the 1-Wire bus. ROM1 and ROM4 will place a 1 onto the 1-Wire, allowing the line to stay high. ROM2 and ROM3 will place a 0 onto the 1-Wire, thus it will be pulled low. The bus master again observes a 0 for the complement of the first ROM data bit. The bus master has determined that there are some devices on the 1-Wire bus that have a 0 in the first position and others that have a 1.

The data obtained from the two reads of the 3–step routine have the following interpretations:

- 00 There are still devices attached which have conflicting bits in this position.
  - 01 All devices still coupled have a 0–bit in this bit position.
  - 10 All devices still coupled have a 1–bit in this bit position.
  - 11 There are no devices attached to the 1–Wire bus.
4. The bus master writes a 0. This deselects ROM2 and ROM3 for the remainder of this search pass, leaving only ROM1 and ROM4 connected to the 1–Wire bus.
  5. The bus master performs two more reads and receives a 0–bit followed by a 1–bit. This indicates that all devices still coupled to the bus have 0's as their second ROM data bit.
  6. The bus master then writes a 0 to keep both ROM1 and ROM4 coupled.
  7. The bus master executes two reads and receives two 0–bits. This indicates that both 1–bits and 0–bits exist as the third bit of the ROM data of the attached devices.
  8. The bus master writes a 0–bit. This deselects ROM1 leaving ROM4 as the only device still connected.
  9. The bus master reads the remainder of the ROM bits for ROM4 and continues to access the part if desired. This completes the first pass and uniquely identifies one part on the 1–Wire bus.
  10. The bus master starts a new ROM search sequence by repeating steps 1 through 7.
  11. The bus master writes a 1–bit. This decouples ROM4, leaving only ROM1 still coupled.
  12. The bus master reads the remainder of the ROM bits for ROM1 and communicates to the underlying logic if desired. This completes the second ROM search pass, in which another of the ROMs was found.
  13. The bus master starts a new ROM search by repeating steps 1 through 3.
  14. The bus master writes a 1–bit. This deselects ROM1 and ROM4 for the remainder of this search pass, leaving only ROM2 and ROM3 coupled to the system.

15. The bus master executes two read time slots and receives two zeros.

16. The bus master writes a 0–bit. This decouples ROM3, and leaving only ROM2.
17. The bus master reads the remainder of the ROM bits for ROM2 and communicates to the underlying logic if desired. This completes the third ROM search pass, in which another of the ROMs was found.
18. The bus master starts a new ROM search by repeating steps 13 through 15.
19. The bus master writes a 1–bit. This decouples ROM2, leaving only ROM3.
20. The bus master reads the remainder of the ROM bits for ROM3 and communicates to the underlying logic if desired. This completes the fourth ROM search pass, in which another of the ROMs was found.

**Note the following:**

The bus master learns the unique ID number (ROM data pattern) of one 1–Wire device on each ROM Search operation. The time required to derive the part's unique ROM code is:

$$960 \mu\text{s} + (8 + 3 \times 64) 61 \mu\text{s} = 13.16 \text{ ms}$$

The bus master is therefore capable of identifying 75 different 1–Wire devices per second.

**I/O SIGNALING**

The DS1820 requires strict protocols to insure data integrity. The protocol consists of several types of signaling on one line: reset pulse, presence pulse, write 0, write 1, read 0, and read 1. All of these signals, with the exception of the presence pulse, are initiated by the bus master.

The initialization sequence required to begin any communication with the DS1820 is shown in Figure 11. A reset pulse followed by a presence pulse indicates the DS1820 is ready to send or receive data given the correct ROM command and memory function command.

The bus master transmits (TX) a reset pulse (a low signal for a minimum of 480  $\mu\text{s}$ ). The bus master then releases the line and goes into a receive mode (RX). The 1–Wire bus is pulled to a high state via the 5K pull–up resistor. After detecting the rising edge on the

I/O pin, the DS1820 waits 15–60  $\mu$ s and then transmits the presence pulse (a low signal for 60–240  $\mu$ s).

### **MEMORY COMMAND FUNCTIONS**

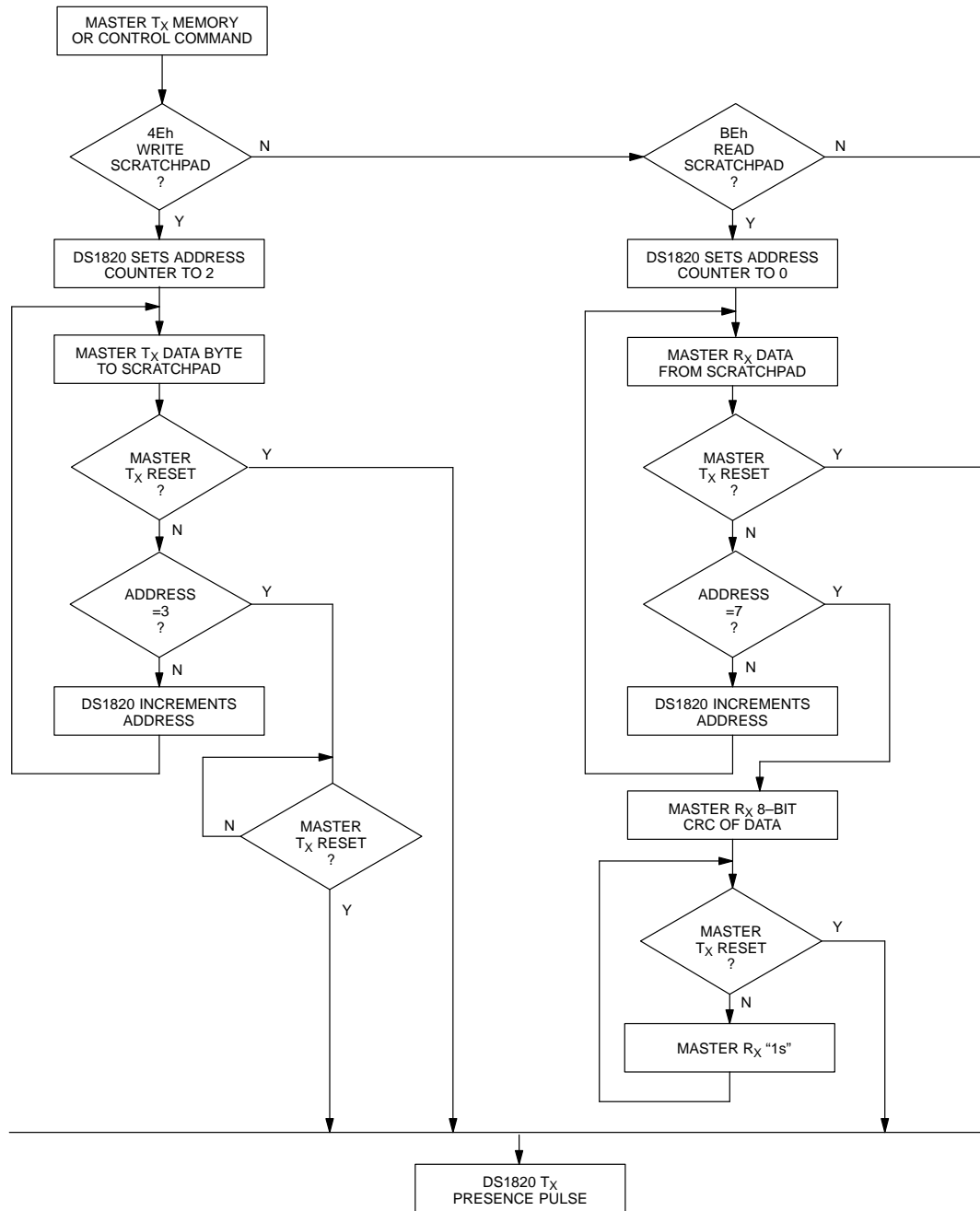
The following command protocols are summarized in Table 2, and by the flowchart of Figure 10.

### **Write Scratchpad [4Eh]**

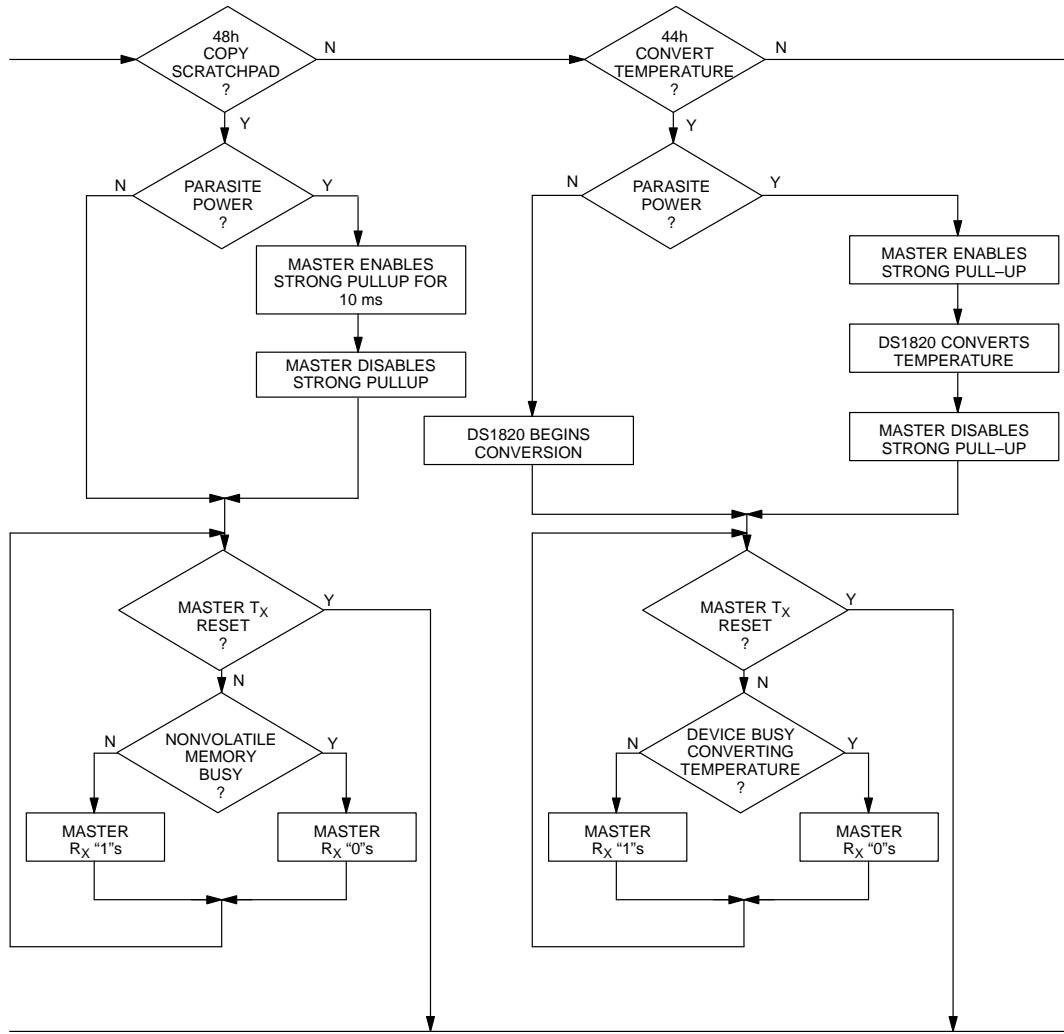
This command writes to the scratchpad of the DS1820, starting at address 2. The next two bytes written will be saved in scratchpad memory, at address locations 2 and 3. Writing may be terminated at any point by issuing a reset.



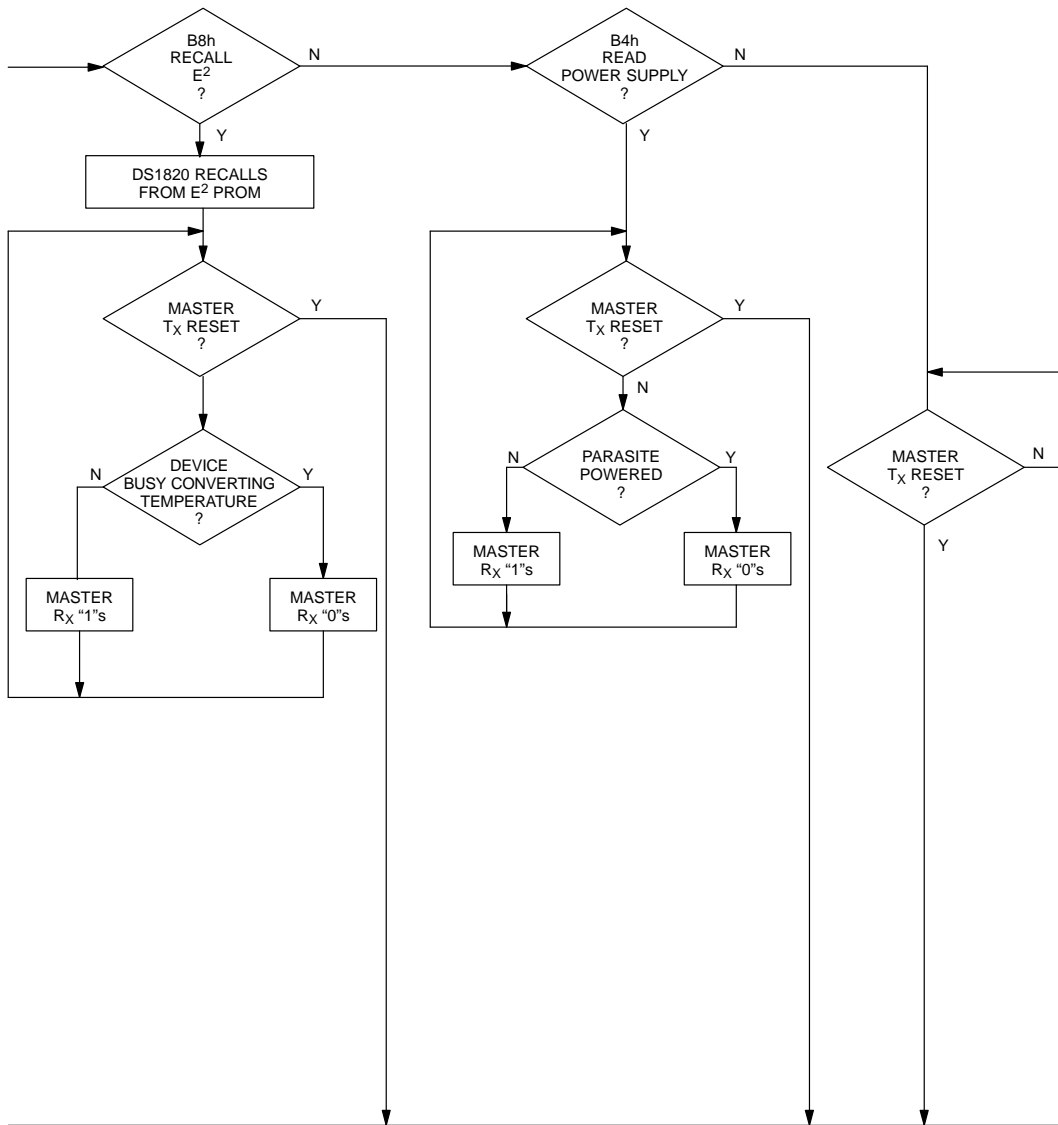
MEMORY FUNCTIONS FLOW CHART Figure 10

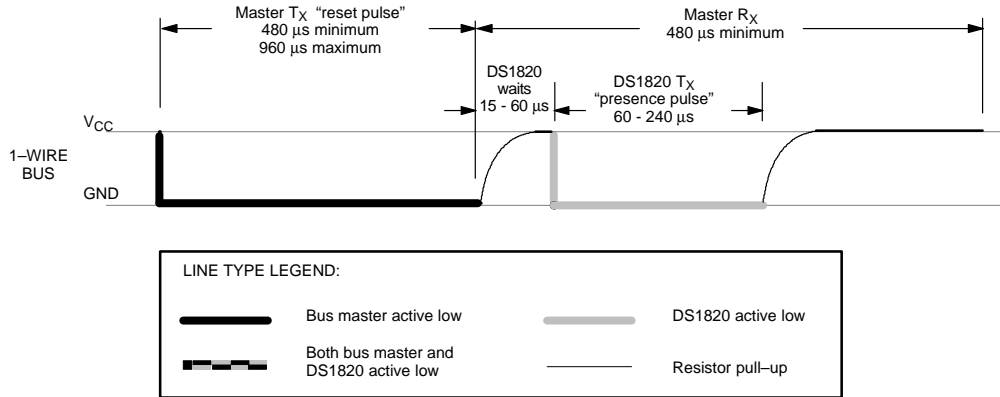


MEMORY FUNCTIONS FLOW CHART Figure 10 (cont'd)



MEMORY FUNCTIONS FLOW CHART Figure 10 (cont'd)



**INITIALIZATION PROCEDURE “RESET AND PRESENCE PULSES”** Figure 11**DS1820 COMMAND SET** Table 2

INSTRUCTION	DESCRIPTION	PROTOCOL	1-WIRE BUS AFTER ISSUING PROTOCOL	NOTES
<b>TEMPERATURE CONVERSION COMMANDS</b>				
Convert T	Initiates temperature conversion.	44h	<read temperature busy status>	1
<b>MEMORY COMMANDS</b>				
Read Scratchpad	Reads bytes from scratchpad and reads CRC byte.	BEh	<read data up to 9 bytes>	
Write Scratchpad	Writes bytes into scratchpad at addresses 2 and 3 (TH and TL temperature triggers).	4Eh	<write data into 2 bytes at addr. 2 and addr. 3>	
Copy Scratchpad	Copies scratchpad into nonvolatile memory (addresses 2 and 3 only).	48h	<read copy status>	2
Recall E <sup>2</sup>	Recalls values stored in nonvolatile memory into scratchpad (temperature triggers).	B8h	<read temperature busy status>	
Read Power Supply	Signals the mode of DS1820 power supply to the master.	B4h	<read supply status>	

**NOTES:**

1. Temperature conversion takes up to 500 ms. After receiving the Convert T protocol, if the part does not receive power from the V<sub>DD</sub> pin, the I/O line for the DS1820 must be held high for at least 500 ms to provide power during the conversion process. As such, no other activity may take place on the 1-Wire bus for at least this period after a Convert T command has been issued.
2. After receiving the Copy Scratchpad protocol, if the part does not receive power from the V<sub>DD</sub> pin, the I/O line for the DS1820 must be held high for at least 10 ms to provide power during the copy process. As such, no other activity may take place on the 1-Wire bus for at least this period after a Copy Scratchpad command has been issued.



**Read Scratchpad [BEh]**

This command reads the contents of the scratchpad. Reading will commence at byte 0, and will continue through the scratchpad until the 9th (byte=8, CRC) byte is read. If not all locations are to be read, the master may issue a reset to terminate reading at any time.

**Copy Scratchpad [48h]**

This command copies the scratchpad into the E<sup>2</sup> memory of the DS1820, storing the temperature trigger bytes in nonvolatile memory. If the bus master issues read time slots following this command, the DS1820 will output "0" on the bus as long as it is busy copying the scratchpad to E<sup>2</sup>; it will return a "1" when the copy process is complete. If parasite powered, the bus master has to enable a strong pull-up for at least 10 ms immediately after issuing this command.

**Convert T [44h]**

This command begins a temperature conversion. No further data is required. The temperature conversion will be performed and then the DS1820 will remain idle. If the bus master issues read time slots following this command, the DS1820 will output "0" on the bus as long as it is busy making a temperature conversion; it will return a "1" when the temperature conversion is complete. If parasite powered, the bus master has to enable a strong pullup for 500 ms immediately after issuing this command.

**Recall E2 [B8h]**

This command recalls the temperature trigger values stored in E<sup>2</sup> to the scratchpad. This recall operation happens automatically upon power-up to the DS1820 as well, so valid data is available in the scratchpad as soon as the device has power applied. With every read data time slot issued after this command has been sent, the device will output its temperature converter busy flag "0"=busy, "1"=ready.

**Read Power Supply [B4h]**

With every read data time slot issued after this command has been sent to the DS1820, the device will signal its power mode: "0"=parasite power, "1"=external power supply provided.

**READ/WRITE TIME SLOTS**

DS1820 data is read and written through the use of time slots to manipulate bits and a command word to specify the transaction.

**Write Time Slots**

A write time slot is initiated when the host pulls the data line from a high logic level to a low logic level. There are two types of write time slots: Write One time slots and Write Zero time slots. All write time slots must be a minimum of 60  $\mu$ s in duration with a minimum of a one  $\mu$ s recovery time between individual write cycles.

The DS1820 samples the I/O line in a window of 15  $\mu$ s to 60  $\mu$ s after the I/O line falls. If the line is high, a Write One occurs. If the line is low, a Write Zero occurs (see Figure 12).

For the host to generate a Write One time slot, the data line must be pulled to a logic low level and then released, allowing the data line to pull up to a high level within 15  $\mu$ s after the start of the write time slot.

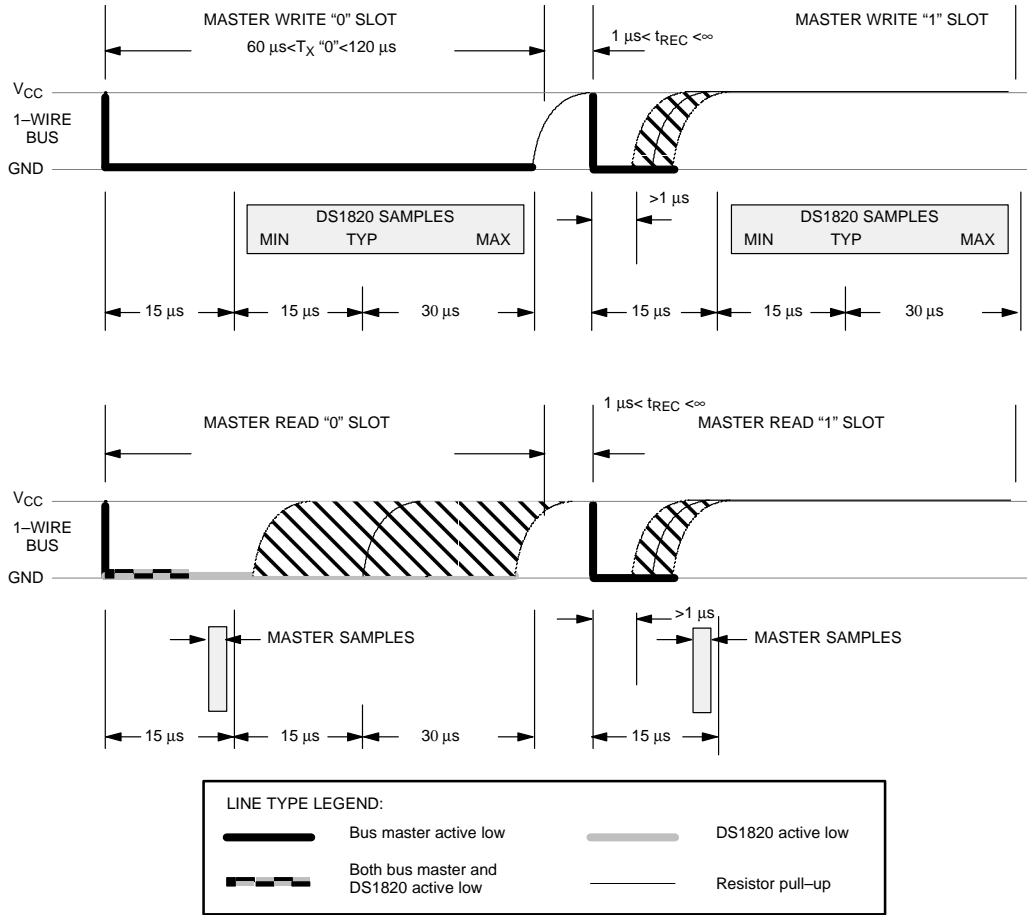
For the host to generate a Write Zero time slot, the data line must be pulled to a logic low level and remain low for 60  $\mu$ s.

**Read Time Slots**

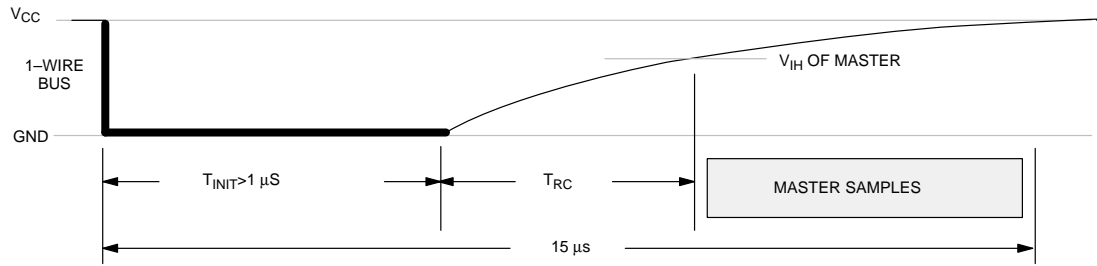
The host generates read time slots when data is to be read from the DS1820. A read time slot is initiated when the host pulls the data line from a logic high level to logic low level. The data line must remain at a low logic level for a minimum of one  $\mu$ s; output data from the DS1820 is valid for 15  $\mu$ s after the falling edge of the read time slot. The host therefore must stop driving the I/O pin low in order to read its state 15  $\mu$ s from the start of the read slot (see Figure 12). By the end of the read time slot, the I/O pin will pull back high via the external pull-up resistor. All read time slots must be a minimum of 60  $\mu$ s in duration with a minimum of a one  $\mu$ s recovery time between individual read slots.

Figure 13 shows that the sum of  $T_{INIT}$ ,  $T_{RC}$ , and  $T_{SAMPLE}$  must be less than 15  $\mu$ s. Figure 14 shows that system timing margin is maximized by keeping  $T_{INIT}$  and  $T_{RC}$  as small as possible and by locating the master sample time towards the end of the 15  $\mu$ s period.

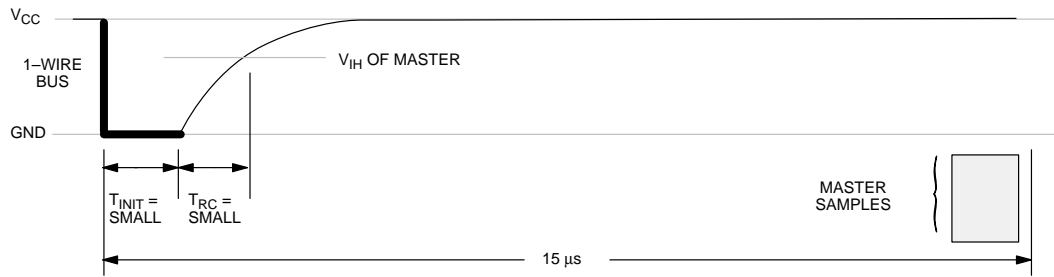
**READ/WRITE TIMING DIAGRAM** Figure 12



**DETAILED MASTER READ "1" TIMING Figure 13**



**RECOMMENDED MASTER READ "1" TIMING Figure 14**



LINE TYPE LEGEND:			
	Bus master active low		DS1820 active low
	Both bus master and DS1820 active low		Resistor pull-up

**Related Application Notes**

The following Application Notes can be applied to the DS1820. These notes can be obtained from the Dallas

Semiconductor "Application Note Book", via our website at <http://www.dalsemi.com/>, or through our faxback service at (214) 450-0441.

Application Note 27: "Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor Touch Memory Product"

Application Note 55: "Extending the Contact Range of Touch Memories"

Application Note 74: "Reading and Writing Touch Memories via Serial Interfaces"

Application Note 104: "Minimalist Temperature Control Demo"

Application Note 105: "High Resolution Temperature Measurement with Dallas Direct-to-Direct Temperature Sensors"

Application Note 106: "Complex MicroLANs"

Application Note 108: "MicroLAN – In the Long Run"

Sample 1-Wire subroutines that can be used in conjunction with AN74 can be downloaded from the website or our Anonymous FTP Site.



**MEMORY FUNCTION EXAMPLE Table 3**

Example: Bus Master initiates temperature conversion, then reads temperature (parasite power assumed).

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Reset pulse (480–960 $\mu$ s).
RX	Presence	Presence pulse.
TX	55h	Issue "Match ROM" command.
TX	<64-bit ROM code>	Issue address for DS1820.
TX	44h	Issue "Convert T" command.
TX	<I/O LINE HIGH>	I/O line is held high for at least 500 ms by bus master to allow conversion to complete.
TX	Reset	Reset pulse.
RX	Presence	Presence pulse.
TX	55h	Issue "Match ROM" command.
TX	<64-bit ROM code>	Issue address for DS1820.
TX	BEh	Issue "Read Scratchpad" command.
RX	<9 data bytes>	Read entire scratchpad plus CRC; the master now recalculates the CRC of the eight data bytes received from the scratchpad, compares the CRC calculated and the CRC read. If they match, the master continues; if not, this read operation is repeated.
TX	Reset	Reset Pulse.
RX	Presence	Presence pulse, done.

**MEMORY FUNCTION EXAMPLE** Table 4

Example: Bus Master writes memory (parasite power and only one DS1820 assumed).

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Reset pulse.
RX	Presence	Presence pulse.
TX	CCh	Skip ROM command.
TX	4Eh	Write Scratchpad command.
TX	<2 data bytes>	Writes two bytes to scratchpad (TH and TL).
TX	Reset	Reset pulse.
RX	Presence	Presence pulse.
TX	CCh	Skip ROM command.
TX	BEh	Read Scratchpad command.
RX	<9 data bytes>	Read entire scratchpad plus CRC. The master now recalculates the CRC of the eight data bytes received from the scratchpad, compares the CRC and the two other bytes read back from the scratchpad. If data match, the master continues; if not, repeat the sequence.
TX	Reset	Reset pulse.
RX	Presence	Presence pulse.
TX	CCh	Skip ROM command.
TX	48h	Copy Scratchpad command; after issuing this command, the master must wait 6 ms for copy operation to complete.
TX	Reset	Reset pulse.
RX	Presence	Presence pulse, done.

**MEMORY FUNCTION EXAMPLE** Table 5

Example: Temperature conversion and interpolation (external power supply and only one DS1820 assumed).

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Reset pulse.
TR	Presence	Presence pulse.
TX	CCh	Skip ROM command.
TX	44h	Convert T command.
RX	<1 data byte>	Read busy flag eight times. The master continues reading one byte (or bit) after another until the data is FFh (all bits 1).
TX	Reset	Reset pulse.
RX	Presence	Presence pulse.
TX	CCh	Skip ROM command.
TX	BEh	Read Scratchpad command.
RX	<9 data bytes>	Read entire scratchpad plus CRC. The master now recalculates the CRC of the eight data bytes received from the scratchpad and compares both CRCs. If the CRCs match, the data is valid. The master saves the temperature value and stores the contents of the count register and count per °C register as COUNT_REMAIN and COUNT_PER_C, respectively.
TX	Reset	Reset pulse.
RX	Presence	Presence pulse, done.
–	–	CPU calculates temperature as described in the data sheet for higher resolution.

**ABSOLUTE MAXIMUM RATINGS\***

Voltage on Any Pin Relative to Ground	-0.5V to +7.0V
Operating Temperature	-55°C to +125°C
Storage Temperature	-55°C to +125°C
Soldering Temperature	260°C for 10 seconds

\* 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.

**RECOMMENDED DC OPERATING CONDITIONS**

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	$V_{DD}$	I/O Functions	2.8	5.0	5.5		1, 2
		$\pm 1/2^\circ\text{C}$ Accurate Temperature Conversions	4.3		5.5	V	
Data Pin	I/O		-0.5		+5.5	V	2
Logic 1	$V_{IH}$		2.0		$V_{CC}+0.3$	V	2, 3
Logic 0	$V_{IL}$		-0.3		+0.8	V	2, 4

**DC ELECTRICAL CHARACTERISTICS**(-55°C to +125°C;  $V_{DD}=3.6\text{V}$  to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Thermometer Error	$t_{ERR}$	-0°C to +70°C			$\pm 1/2$	°C	1, 9, 10
		-55°C to 0°C and +70°C to +125°C			See Typical Curve		
Input Logic High	$V_{IH}$		2.2		5.5	V	2, 3
Input Logic Low	$V_{IL}$		-0.3		+0.8	V	2, 4
Sink Current	$I_L$	$V_{I/O}=0.4\text{V}$	-4.0			mA	2
Standby Current	$I_Q$			200	350	nA	8
Active Current	$I_{DD}$			1	1.5	mA	5, 6
Input Load Current	$I_L$			5		$\mu\text{A}$	7

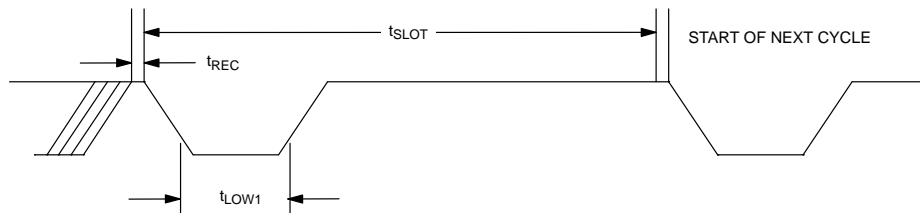


**AC ELECTRICAL CHARACTERISTICS:**(-55°C to +125°C;  $V_{DD}=3.6V$  to 5.5V)

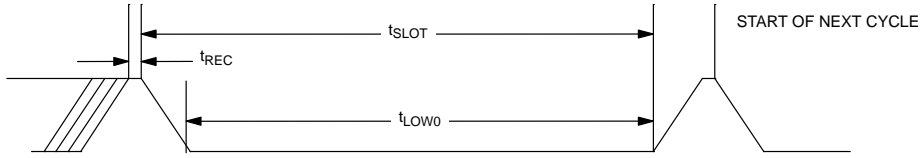
PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Temperature Conversion Time	$t_{CONV}$		200	500	ms	
Time Slot	$t_{SLOT}$	60		120	$\mu s$	
Recovery Time	$t_{REC}$	1			$\mu s$	
Write 0 Low Time	$t_{LOW0}$	60		120	$\mu s$	
Write 1 Low Time	$t_{LOW1}$	1		15	$\mu s$	
Read Data Valid	$t_{RDV}$			15	$\mu s$	
Reset Time High	$t_{RSTH}$	480			$\mu s$	
Reset Time Low	$t_{RSTL}$	480		4800	$\mu s$	
Presence Detect High	$t_{PDHIGH}$	15		60	$\mu s$	
Presence Detect Low	$t_{PDLOW}$	60		240	$\mu s$	
Capacitance	$C_{IN/OUT}$			25	pF	

**NOTES:**

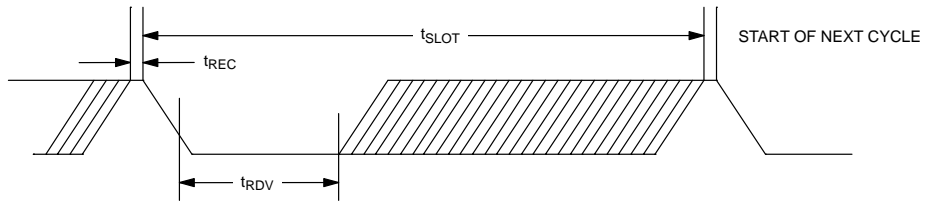
- Temperature conversion will work with  $\pm 2^\circ C$  accuracy down to  $V_{DD} = 3.4$  volts.
- All voltages are referenced to ground.
- Logic one voltages are specified at a source current of 1 mA.
- Logic zero voltages are specified at a sink current of 4 mA.
- $I_{DD}$  specified with  $V_{CC}$  at 5.0 volts.
- Active current refers to either temperature conversion or writing to the  $E^2$  memory. Writing to  $E^2$  memory consumes approximately 200  $\mu A$  for up to 10 ms.
- Input load is to ground.
- Standby current specified up to 70°C. Standby current typically is 5  $\mu A$  at 125°C.
- See Typical Curve for specification limits outside the 0°C to 70°C range. Thermometer error reflects sensor accuracy as tested during calibration.
- Typical accuracy curve valid for  $4.3V \leq V_{DD} \leq 5.5V$ .

**1-WIRE WRITE ONE TIME SLOT**

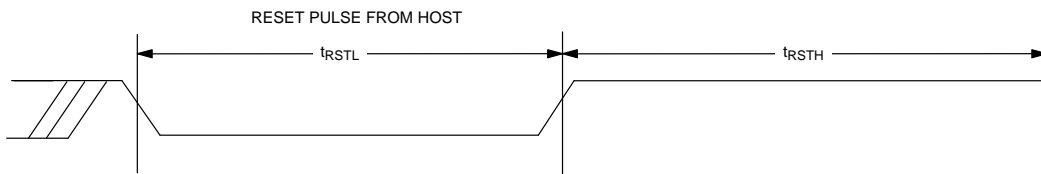
**1-WIRE WRITE ZERO TIME SLOT**



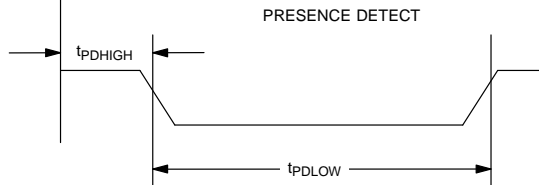
**1-WIRE READ ZERO TIME SLOT**



**1-WIRE RESET PULSE**



**1-WIRE PRESENCE DETECT**



**TYPICAL PERFORMANCE CURVE**

DS1820 DIGITAL TERMOMETER AND THERMOSTAT  
TEMPERATURE READING ERROR

