DS1957

Cryptographic Java-Powered iButton™

www.iButton.com

SECURITY FEATURES

- Single-chip, physically secure coprocessor for non-secure host
- Arithmetic accelerator executes 1024-bit public key cryptography in less than 1 second
- Unresettable True Time Clock self-imposes expiration dates and date/time stamping
- Durable stainless steel case clearly shows visual evidence of physical tampering
- 134 kbytes NV SRAM zeroes itself in response to tampering or cooling below -50°C
- 64 kbytes of ROM stores unalterable validated firmware as Software ICs
- Unique, factory-lasered 64-bit registration number (8-bit family code + 48-bit serial number + 8-bit CRC tester) assures absolute traceability because no two parts are alike
- Multi-drop controller supports multiple iButtons on the same signal line for n-factor security
- Signal path to the chip is limited to the lid of the iButton to enhance security
- Communication protocol includes 16-bit CRCs to insure error free packet transfers to the buffer memory even with intermittent connection
- The iButton can be accessed while affixed to a wearable accessory to lower the chance of being lost or stolen
- Only an authorized service provider can install a transaction file which programs the iButton for a specific application
- The transaction file contains scripts that call on tested pre-fabricated cryptographic functions
- Each file can be locked after installation so that additional transaction files for new services may be installed without interference

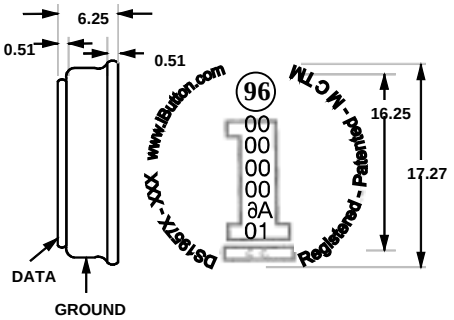
GENERAL FEATURES

- Overdrive mode boosts communication speed from 16.3 kbits to 142 kbits per second
- Operating temperature range from -20°C to +70°C
- Over 5 years of data retention
- Timing accuracy better than ± 4 minutes/month at 25°C
- Button shape is insensitive to angular orientation and self-aligning with the Dot receptor
- The iButton is easily affixed with self-stick adhesive backing, latched by its flange, or locked with a ring pressed onto its rim
- Mechanical specification and 1-Wire® communication follow the iButton Book Of Standards

SUPPORT

- Script tool kit speeds the development of applications such as micro cash meters or stored value purses
- Adapters for LPT port (parallel), COM Port (serial), ETHERNET (10-base-T), or USB connectivity

F5 MICROCAN



All dimensions are shown in millimeters.

ISSUING AND ACTIVATION INFORMATION

See www.ibutton.com

EXAMPLES OF ACCESSORIES

DS1410E	Parallel Port Button Holder
DS1402D-DB8	iButton Dot Receptor
DS9093F	Snap-In Fob
DS9096P	Self-Stick Adhesive Pad
DS9093RA	Mounting Lock Ring
DS9101	Multi-Purpose Clip

CRYPTO iButton DESCRIPTION

The DS1957 Crypto iButton is a secure coprocessor residing in a stainless steel MicroCan with 1-Wire interface. The device is addressed by matching its individual 64-bit factory-lasered registration number. The 64-bit number consists of an 8-bit family code, a unique 48-bit serial number, and an 8-bit cyclic redundancy check. Data is transferred serially via the 1-Wire communication protocol which requires only a single data lead and a ground return for communication and power. Typical applications include authentication of the person at the computer, secure transmission of E-mail, electronic notary service, electronic cash dispenser, electronic secure monetary transactions, software authorization and usage metering, postal metering service, electronic signature, micro-cash metering and physically secure coprocessor array for servers.

OVERVIEW

The block diagram in Figure 1 shows the relationships between the major control and memory sections of the DS1957. The DS1957 has four main components: 1) 64-bit lasered ROM, 2) I/O and status registers, 3) microcomputer with 64 kbyte firmware and 134 kbyte nonvolatile data memory, and 4) arithmetic accelerator for modular arithmetic. The device derives its power for I/O communication entirely 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 of this "parasite" power source during the low times of the 1-Wire line until it returns high to replenish the parasite (capacitor) supply. During program execution, the 1-Wire line must be pulled high to 5 volt via a low-impedance transistor to provide the energy for the microcomputer and the accelerator to operate. After a time period that was agreed between bus master and Crypto iButton, the DS1957 stops executing the program and waits for the bus master to initiate another processing cycle and so on, until data processing is finished. The timing is based on the True Time Clock of the DS1957 and the Real Time Clock of the bus master.

The hierarchical structure of the 1-Wire protocol is shown in Figure 2. The bus master must first provide one of the six ROM Function Commands, 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, 5) Overdrive-Skip ROM or 6) Overdrive-Match ROM. Upon completion of an Overdrive ROM command byte executed at regular speed, the device will enter the Overdrive mode where all subsequent communication occurs at a higher speed. These commands operate on the 64-bit lasered ROM portion of each device and can singulate 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. The protocol required for these ROM Function Commands is described in Figure 6. After a ROM function command is successfully executed, the data transfer and control functions that operate the microcomputer inside the DS1957 become accessible and the bus master may issue any one of the nine commands specific to the DS1957. The protocol for these commands is described in Figure 4. Unless otherwise stated, all data is read and written least significant bit first.

64-BIT LASERED ROM

Each DS1957 contains a unique ROM code that is 64 bits long. The first eight bits are a 1-Wire family code. The next 48 bits are a unique serial number. The last eight bits are a CRC of the first 56 bits. (See Figure 3). The 64-bit ROM and ROM Function Control section allow the DS1957 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 operate the microcomputer and accelerator of the DS1957 are not accessible until the ROM function protocol has been satisfied. This protocol is described in the ROM functions flow chart (Figure 6). The 1-Wire bus master must first provide one of six ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, 5) Overdrive-Skip ROM or 6) Overdrive-Match ROM. After a ROM function sequence has been successfully executed, the bus master may then provide any one of the commands specific to the DS1957 (Figure 4).

The 1-Wire CRC of the lasered ROM is generated using the polynomial $X^8 + X^5 + X^4 + 1$. Additional information about the Dallas Semiconductor 1-Wire Cyclic Redundancy Check is available in the "Book of DS19xx iButton Standards." The shift register acting as the CRC accumulator is initialized to zero. Then starting with the least significant bit of the family code, one bit at a time is shifted in. After the eighth 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 zeroes.

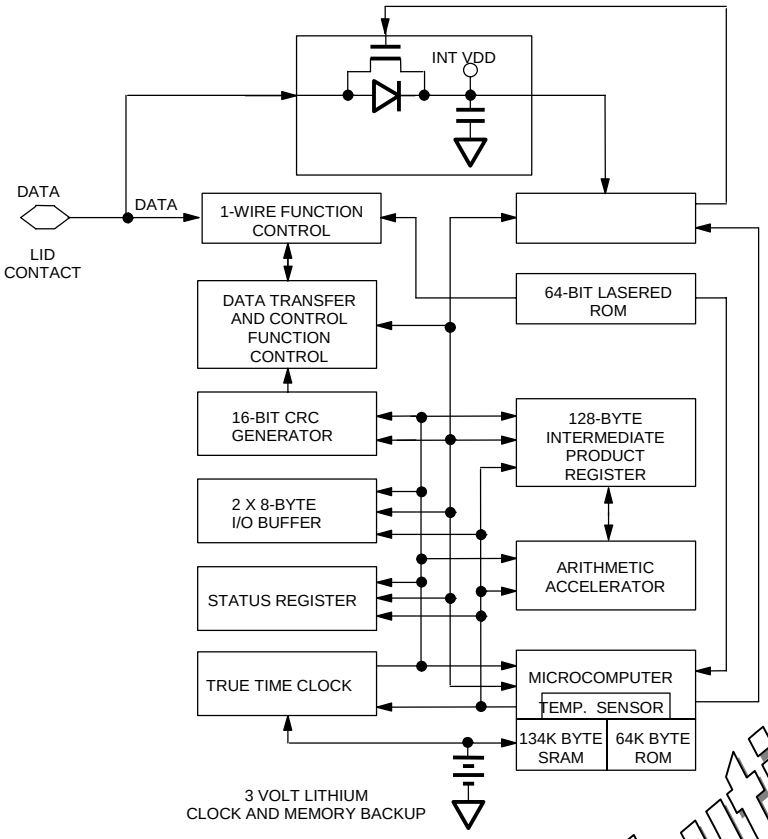
DATA AND CONTROL REGISTERS

From the application software developers perspective, the DS1957 looks like two registers and a buffer and control logic that are accessed through a 1-Wire bus (Figure 1). After the Intermediate Product Register (IPR) and the I/O buffer have been loaded with data and control information, the microcomputer will be started, perform the requested tasks and place the result into the IPR from where it can be read by the bus master. The data processing inside the DS1957 and the structure of the data packets that tell the microcomputer what tasks to perform is governed by the ROM firmware. A detailed description of the firmware functions and data packet formats is found in the "Crypto iButton Firmware Reference Manual."

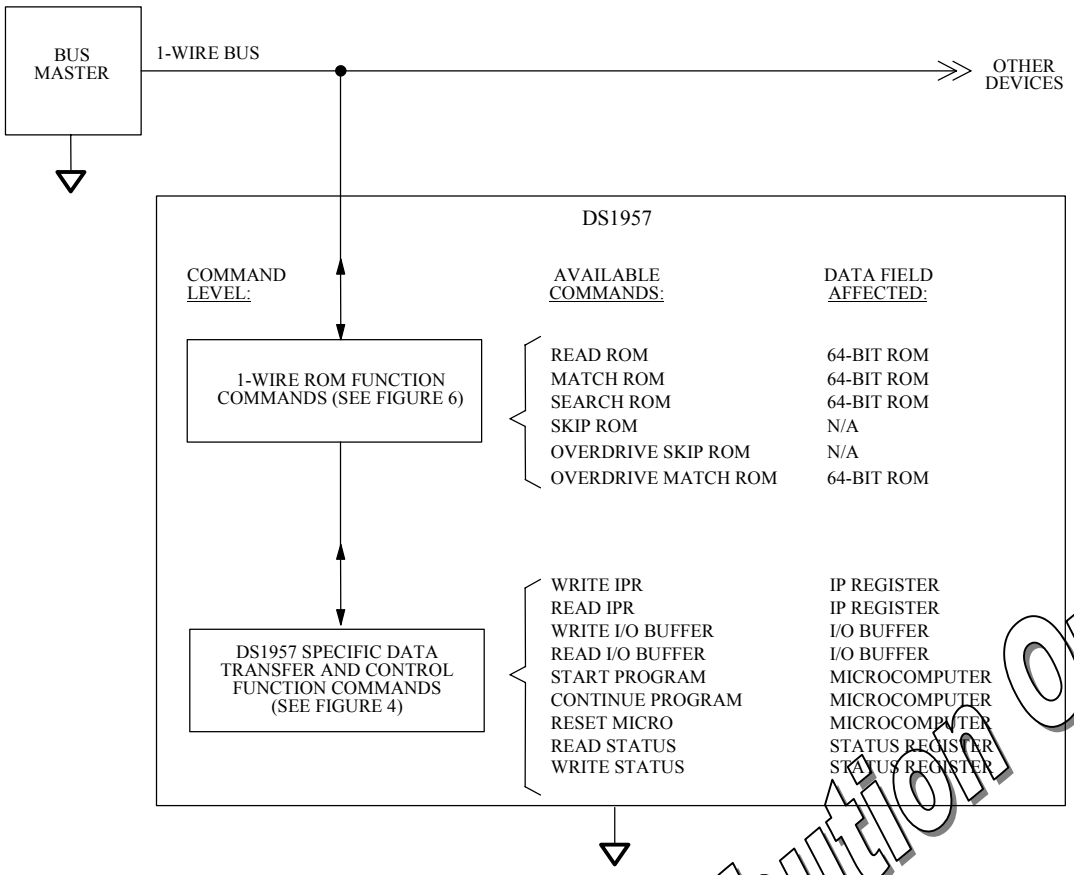
The IPR is normally used to transfer command codes and message data to the microcomputer in blocks of up to 128 bytes and to receive results back from the microcomputer. The messages themselves may extend over several blocks. The message transfer management, which includes block number, block length, bytes remaining, CRC and checksum, is controlled by means of the I/O buffer. During extensive mathematical operations, the microcomputer will pause after a predefined amount of time and allow the bus master to read progress information from the status register. The bus master will then signal the microcomputer to resume computation until the next pause occurs, and so on, until the task is finished.

The duration of the computation time slices is controlled by the True Time Clock. Other functions of the True Time Clock are date/time stamping of events and imposing expiration dates. The arithmetic accelerator is optimized for exponentiation, multiplication and squaring. Such operations are typically required for encryption/decryption of data packets that use the large number theory of public key cryptography.

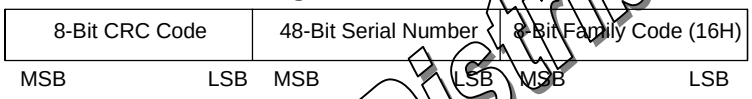
DS1957 FUNCTIONAL BLOCK DIAGRAM Figure 1



HIERARCHICAL STRUCTURE FOR 1-WIRE PROTOCOL Figure 2



64-BIT LASERED ROM Figure 3



DATA TRANSFER AND CONTROL FUNCTION COMMANDS

The "Data Transfer And Control Function Flow Chart" (Figure 4) describes the protocols necessary for accessing the various registers within the DS1957 and to control its microcomputer. The Data Transfer and Control Function Control section interprets the commands issued by the bus master and creates the correct control signals within the device. The protocol issued by the bus master always starts with a command byte which determines if the device is to be read or written or if the microcomputer is to be activated or reset. Except for the Intermediate Product Register (IPR) and reading the Status Register, communication with the DS1957 requires not only issuing the correct command sequence but also providing a command-dependent 16-bit release sequence at the appropriate time. To access the IPR or the I/O buffer for read or write, the bus master has to specify the number of bytes to be transferred. Address information is not required since accessing the IPR and the I/O buffer always begin at address 00. Unless the bus master reads back from the IPR before the microcomputer has processed the data, all data transferred to the DS1957 and received back by the bus master are sent least significant bit first.

WRITE INTERMEDIATE PRODUCT REGISTER (IPR) [0FH]

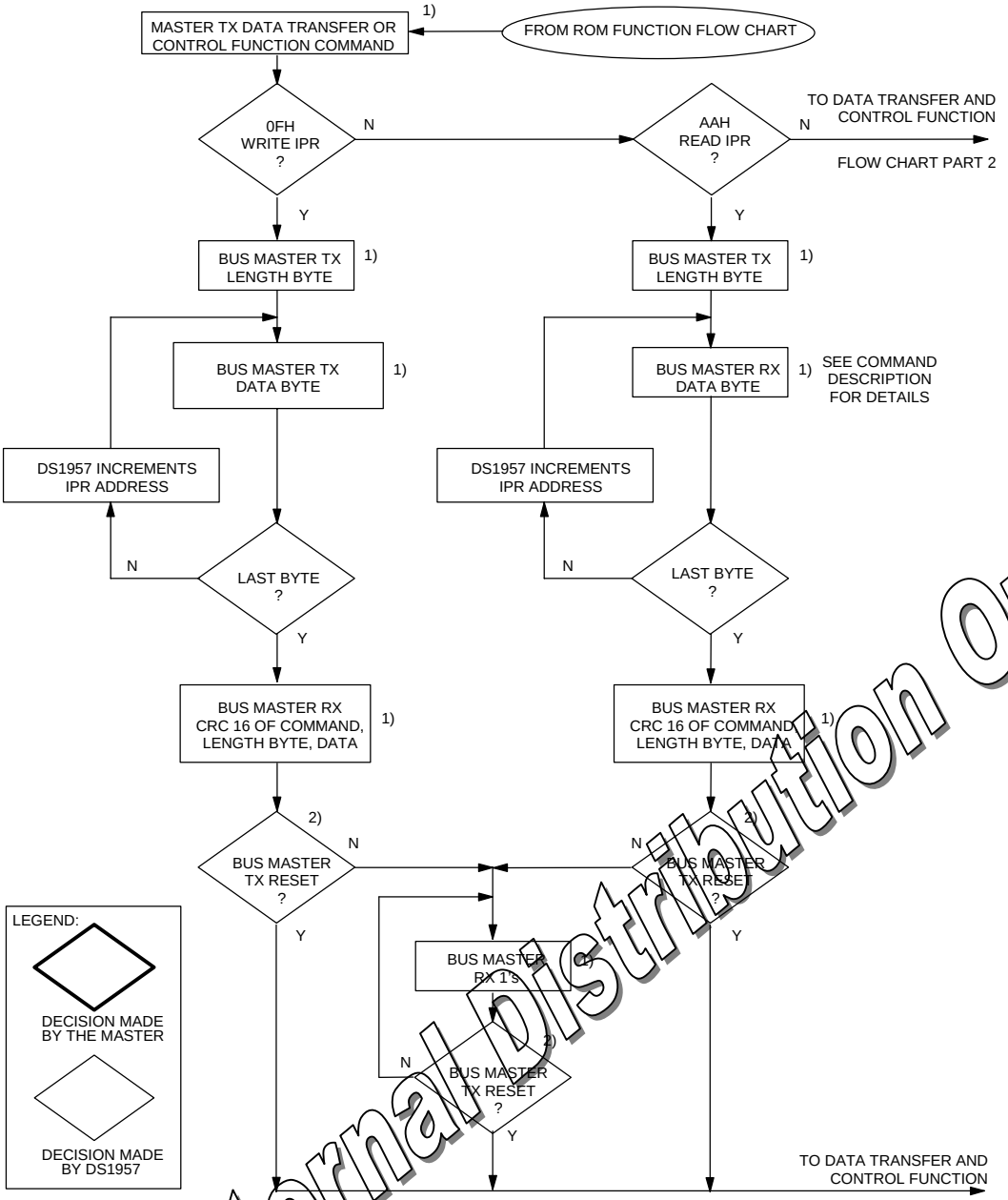
The intermediate product register is used to transfer firmware command codes and data to the microcomputer inside the DS1957 and to read back the results of computations. After the command byte, the bus master transmits the length byte indicating the number of data bytes to be written to the IPR. The length byte may be in the range of 01H to 80H (1 to 128 decimal). The length byte is followed by the data bytes to be written. After as many data bytes as specified by the length byte have been transmitted by the bus master, the DS1957 responds with an inverted CRC16 of the command code, length byte and all data bytes in their original sequence. This ends the Write IPR command and the DS1957 waits for a reset pulse to be sent by the bus master. The CRC allows the bus master to easily check the correctness of the communication without the need for reading back all the data for verification. Writing to the IPR erases as many bytes of the old data as are newly written. Reading back 128 bytes with the READ IPR command sequence will reproduce the new bytes first followed by the remainder of the previous contents.

READ INTERMEDIATE PRODUCT REGISTER (IPR) [AAH]

The intermediate product register is typically read to receive the results of a computation after the microcomputer inside the DS1957 has processed the input data. However, it is also possible to read back the data before it has been processed. The flow chart in both cases is the same, but the data is received in different order.

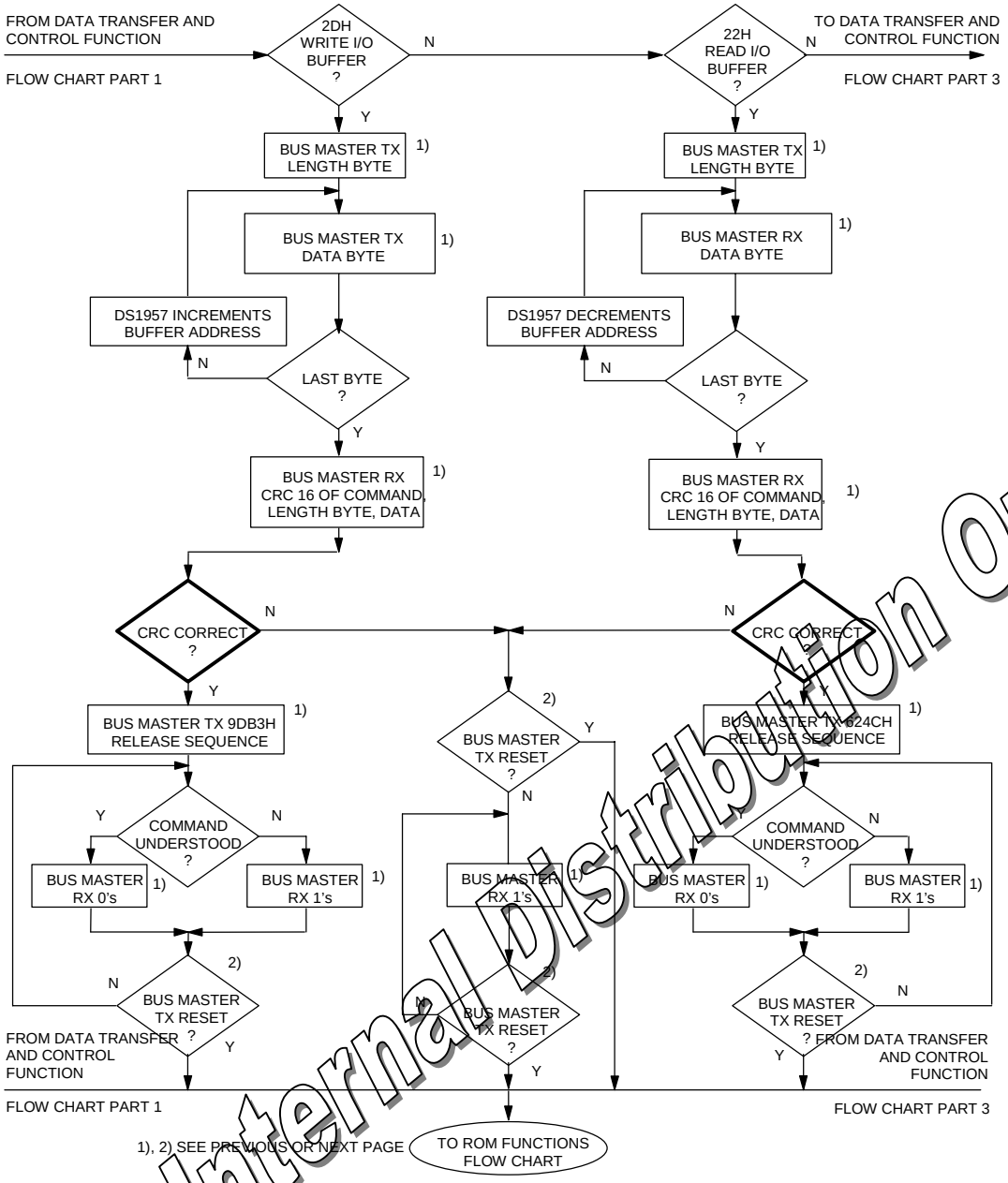
After the microcomputer inside the DS1957 has placed the results of a computation into the IPR, the bus master first reads the I/O buffer (see Read I/O Buffer command) to learn the length of the resulting data in the IPR. In the next step the bus master accesses the DS1957 again, transmits the Read IPR command byte followed by the length byte indicating the number of data bytes to be read from the IPR. Now the bus master reads as many data bytes as specified by the length byte. After this, the DS1957 responds with an inverted CRC16 of the command code, length byte and all data bytes in their original sequence. If the CRC calculated by the bus master does not match the value transmitted by the DS1957, the data transfer was not successful. Due to the hardware design of the IPR (circular buffer) the best way to recover the data is to repeat the computation. If the CRC value matches and the master wants to restore the status of the IPR for a second read of the same data, it has to access the DS1957 again and issue another READ IPR command sequence. If now the length byte is 128 minus the previously used length value, and the master reads as many bytes as the new length byte specifies plus the CRC16, the original status of the IPR is restored. If this second access cycle is omitted or a transmission error occurs before the bus master sends the reset pulse to finish the command, the data placed into the IPR by the microcomputer is lost and the computation needs to be repeated. The second access cycle can be avoided if one always reads all 128 bytes of the IPR.

DATA TRANSFER AND CONTROL FUNCTION FLOW CHART Figure 4

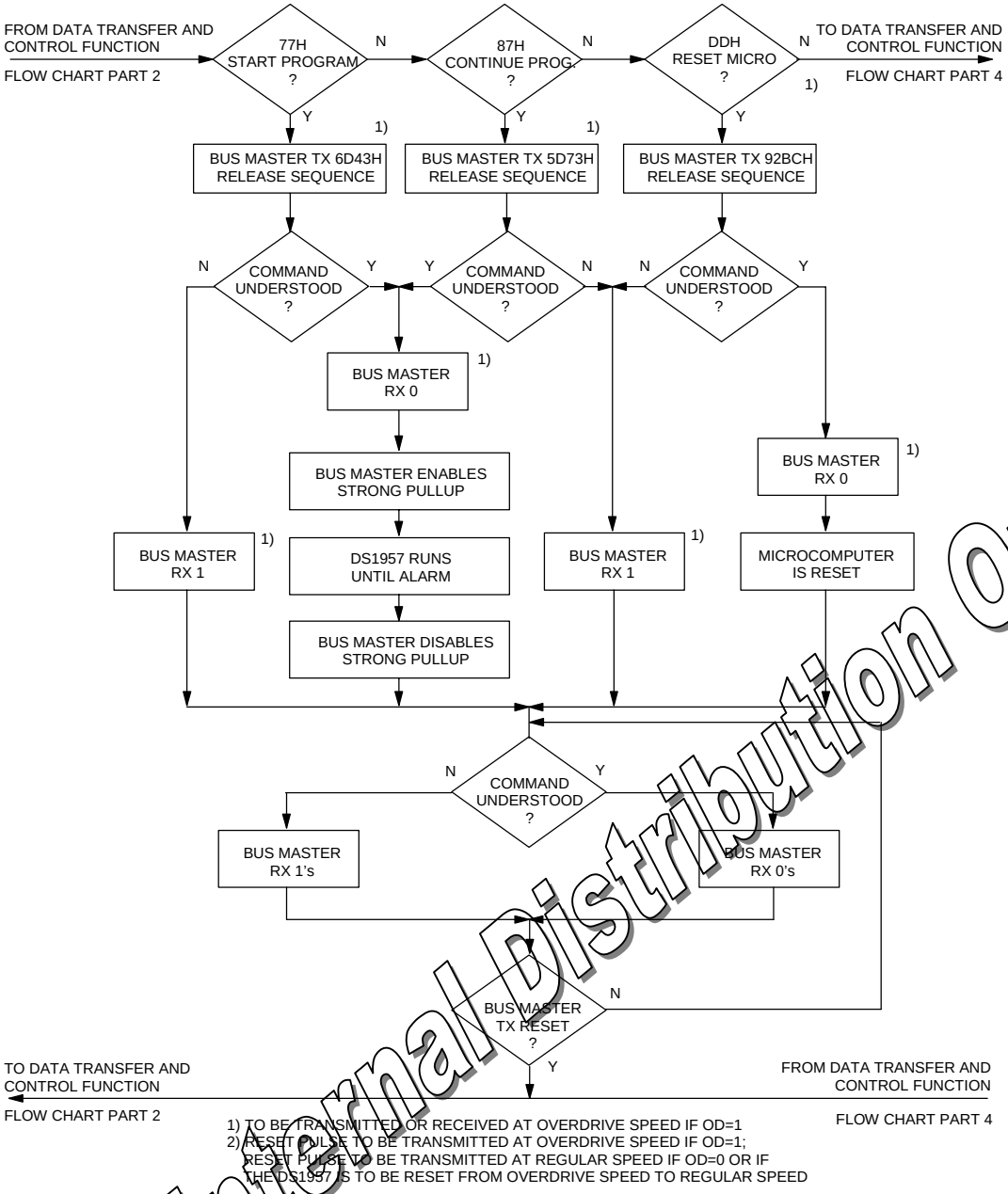


1) TO BE TRANSMITTED OR RECEIVED AT OVERDRIVE SPEED IF OD=1
2) RESET PULSE TO BE TRANSMITTED AT OVERDRIVE SPEED IF OD=1; RESET PULSE TO BE TRANSMITTED AT REGULAR SPEED IF OD=0 OR IF THE DS1957 IS TO BE RESET FROM OVERDRIVE SPEED TO REGULAR SPEED

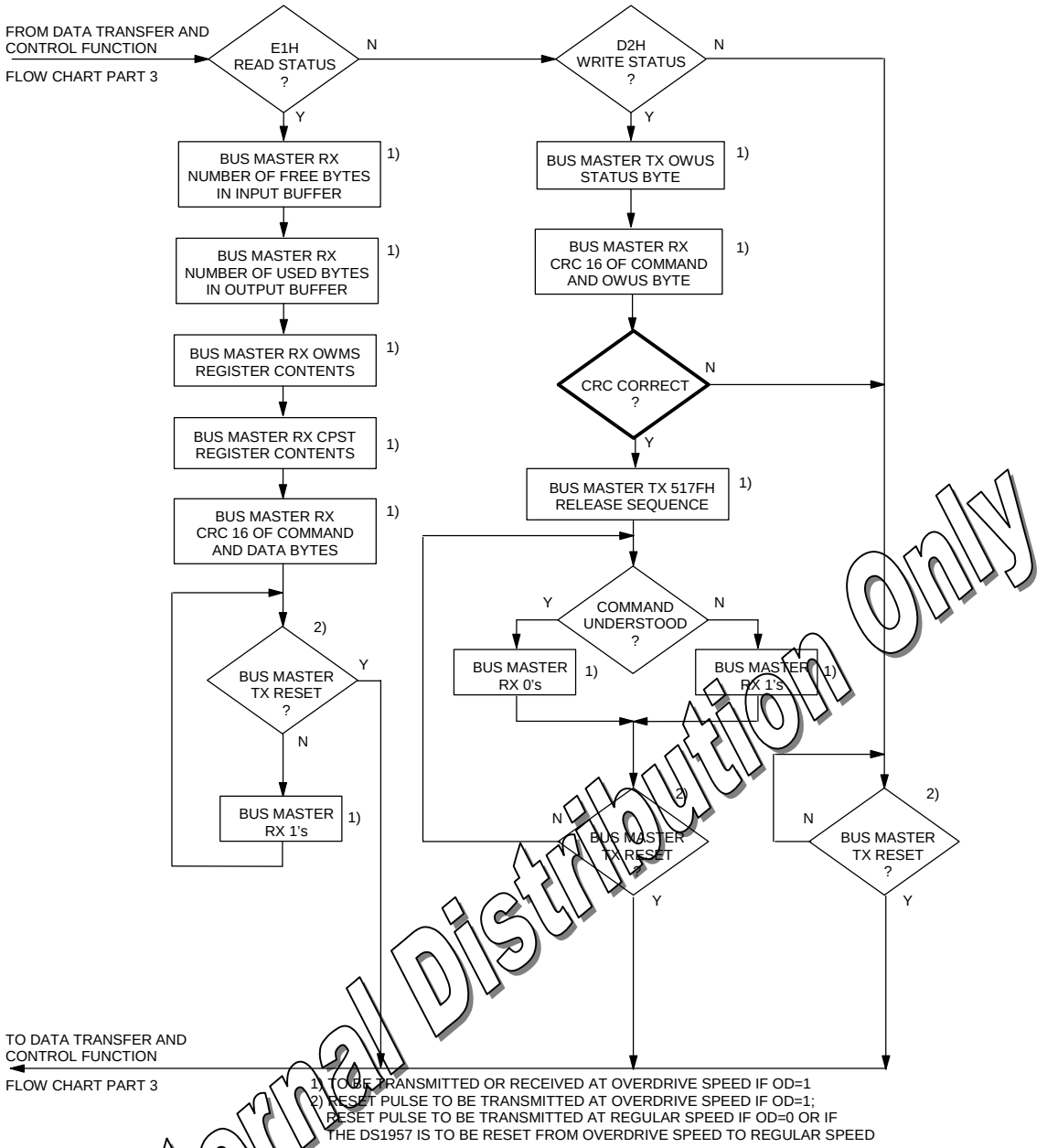
DATA TRANSFER AND CONTROL FUNCTION FLOW CHART Figure 4 (cont'd)



DATA TRANSFER AND CONTROL FUNCTION FLOW CHART Figure 4 (cont'd)



DATA TRANSFER AND CONTROL FUNCTION FLOW CHART Figure 4 (cont'd)



When reading data from the IPR that has not been processed by the microcomputer inside the DS1957, the bus master already knows the actual value of the length byte. To read back data for optional verification, the bus master transmits the Read IPR command byte followed by the length byte and reads as many data bytes as specified by the length byte. After this, the DS1957 responds with an inverted CRC16 of the command code, length byte and all data bytes. The data bytes are received in reversed order, (i.e., the most significant bit of the last byte written to the IPR is received first). After the last byte, the second last follows, and so on. The last byte written is the first to be received (LIFO, last in, first out). The calculation of the CRC is based on the bytes as they are received by the bus master. If the CRC calculated by the bus master does not match the value transmitted by the DS1957, the data transfer was not successful and the IPR needs to be rewritten. If the CRC value matches, the master should access the DS1957 again and issue another READ IPR command sequence. If now the length byte is 128 minus the previously used length value, and the master reads as many bytes as the new length byte specifies plus the CRC16, the original status of the IPR is restored. If this second access cycle is omitted or a transmission error occurs before the bus master sends the reset pulse to finish the command, the data previously written to the IPR is lost and needs to be rewritten. The second access cycle can be avoided if one always reads all 128 bytes of the IPR.

WRITE I/O BUFFER [2DH]

The I/O buffer is used under firmware control to manage the data that is exchanged between bus master and microcomputer through the Intermediate Product Register (IPR). The I/O buffer consists of two separate 8-byte memory sections, one for sending data to (input buffer) and one for receiving data from the DS1957 (output buffer). Both, the write and read access to the I/O buffer require the specification of a length byte. The number of free bytes in the input buffer and valid (used) bytes in the output buffer is available to the bus master by reading the status of the DS1957 using the Read Status command. Details on the format of the messages exchanged through the I/O buffer are specified in the "Crypto iButton Firmware Reference Manual."

After the command code for the Write I/O Buffer command, the bus master transmits the length byte followed by the data bytes to be written. The length byte must not exceed the number of free bytes in the input buffer. After as many data bytes as specified by the length byte have been transmitted, the bus master receives the inverted CRC16 of the command code, length byte and data bytes transmitted. The bus master now calculates the CRC and compares it with the CRC value received from the DS1957. If the CRC values do not match, the data transfer was not successful and the Write I/O Buffer command sequence needs to be repeated. If the CRC values match, the bus master has to transmit the release sequence 9DB3H to signal the microcomputer inside the DS1957 the arrival of new data. As a confirmation, the bus master will receive all 0s if the release sequence was accepted by the DS1957. If the bus master receives 1s instead, the release sequence was not accepted and the command sequence needs to be repeated.

READ I/O BUFFER [22H]

The Read I/O Buffer command flow follows the same syntax as the Write I/O Buffer command. After the command code for the Read I/O Buffer command the bus master transmits the length byte and starts reading. After as many data bytes as specified by the length byte have been received, the bus master receives the inverted CRC16 of the command code, length byte and the data bytes. The bus master now calculates the CRC and compares it with the CRC value received from the DS1957. If the CRC values do not match, the data transfer was not successful and the Read I/O Buffer command sequence needs to be repeated. If the CRC values match, the bus master has to transmit the release sequence 624CH to signal to the microcomputer inside the DS1957 that the data has been read successfully. As a confirmation, the bus master will receive all 0s if the release sequence was accepted by the DS1957. If the bus master receives 1s instead, the release sequence was not accepted and the command sequence needs to be repeated.

START PROGRAM [77H]

The DS1957 Crypto iButton includes a microcomputer that performs logical and mathematical functions on data it receives from the bus master. During 1-Wire communication, this microcomputer is in a stop mode where it consumes virtually no power (see DC Characteristics). After the IPR and I/O buffer have been loaded with command, data and data management information, the microcomputer is started and power is provided through the 1-Wire line. The computations inside the DS1957 are done in time slices, which are controlled by an internal timer. The duration of the time slices is negotiated between the bus master and DS1957 before the program execution starts.

The Start Program command is used to tell the firmware of the DS1957 that the previous command is to be cancelled (if it was not finished) and that internal registers that might still contain data from the previous command are to be cleared for a fresh start. After this clean-up the microcomputer will work on the input data, interpret the firmware command and begin the requested type of data processing until it is stopped by the timer that controls the time slices. After the command code for Start Program the bus master has to transmit the release sequence 6D43H to confirm that it wants to start the microcomputer. Now the bus master has to read back a single bit from the DS1957. If the bus master reads a 1, the command code and/or release sequence have not been accepted and have to be repeated. If the command code and release sequence have been accepted by the DS1957, the bus master will read a 0. Immediately after this, the bus master has to activate a strong pull-up to 5 volt to provide power. After the pre-negotiated time slice for data processing is over, the microcomputer inside the DS1957 will stop. By the use of its own timer the bus master knows when this occurs, disables the strong pull-up and sends a reset pulse. The bus master will get information on the progress of the data processing by reading the status registers of the DS1957.

CONTINUE PROGRAM [87H]

The Continue Program command is used to let the microcomputer inside the DS1957 resume the execution of the current program for another time slice. After the command code for Continue Program the bus master has to transmit the release sequence 5D73H to confirm that it wants to activate the microcomputer again. Now the bus master has to read back a single bit from the DS1957. If the bus master reads a 1, the command code and/or release sequence have not been accepted and have to be repeated. If the command code and release sequence have been accepted by the DS1957, the bus master will read a 0. Immediately after this, the bus master has to activate a strong pull-up to 5 volt to provide power. After the time slice for data processing is over, the microcomputer inside the DS1957 will stop. By the use of its own timer the bus master knows when this occurs, disables the strong pull-up and sends a reset pulse. The bus master will get information on the progress of the data processing by reading the status registers of the DS1957.

RESET MICRO [DDH]

The Reset Micro command is used if the bus master was not able to successfully begin a new data processing sequence using the Start Program command. In contrast to Start Program and Continue Program, the Reset Micro command keeps the microcomputer halted and performs a true hardware reset cycle. This explains why the strong pull-up to 5 volt is not required. After the command code for Reset Micro the bus master has to transmit the release sequence 92BCH to confirm that it wants to reset the microcomputer. Now the bus master has to read back a single bit from the DS1957. If the bus master reads a 1, the command code and/or release sequence have not been accepted and have to be repeated.
If the command code and release sequence have been accepted by the DS1957, the bus master will read a 0. Immediately after this, a reset signal is applied to the microcomputer by its supervisory and control logic. The Reset Micro not only cancels the current firmware command but also clears all internal registers.

READ STATUS [E1H]

In addition to the Intermediate Product Register (IPR) and the I/O buffer for data exchange and exchange management respectively, the DS1957 provides several bytes of status information. These status bytes manage the use of the I/O buffer, signal the availability of resources and the progress of numerical calculations, pass bit-level information to control the program flow within the microcomputer and return bit-level responses. The bus master can read a total of four bytes and can write one byte of status information. The status register names are OWMS (1-Wire Micro Status), CPST (Accelerator Status) and OWUS (1-Wire UART Status).

After the command code for Read Status the bus master will read a total of six bytes. The first byte tells the number of free bytes in the input section of the I/O buffer. This number may be anywhere from 00H to 08H with the 00H standing for a full and the 08H for an empty buffer. The next byte tells the number of used (valid) bytes in the output section of the I/O buffer. This number ranges from 00H to 08H with the 00H standing for an empty and the 08H for a full buffer. The next two bytes are the 1-Wire Micro Status (OWMS) and the Accelerator Status (CPST). The remaining two bytes of the sequence are the inverted CRC16 over the Read Status command code and the four bytes of status information.

Both the OWMS and CPST are read-only for the bus master. The lower six bits of the OWMS signal various conditions and bit-level responses from the microcomputer inside the DS1957. The details on the use of these bits are explained in the "Crypto iButton Firmware Reference Manual." Bit 6 (BPOR) signals power failures. Bit 7 (IOST) indicates the availability of the Intermediate Product Register for reading/writing data through the 1-Wire bus. Depending on the status of the program, the IPR may temporarily be locked for exclusive use by the arithmetic accelerator.

OWMS REGISTER

7	6	5	4	3	2	1	0
IOST	BPOR	STAT5	STAT4	STAT3	STAT2	STAT1	STAT0

6	BPOR	0 = normal operation 1 = indicates power failure
7	IOST	0 = bus master may access the I/O buffer as well as the IPR 1 = bus master can only access the I/O buffer

The read-only CPST register directly interfaces to the accelerator's control logic. The upper seven bits of the CPST register represent the seven most significant bits of the accelerator's 11 bit exponentiation counter. Bit 0 of CPST is the logical OR of all bits of this counter. At the beginning of a numerical calculation, the exponentiation counter starts at a high number and decrements. The lower the number in the CPST register, the further the calculation has progressed. The calculation is finished when CPST reads 00H.

WRITE STATUS [D2H]

The 1-Wire UART Status (OWUS) is the only status byte that can be written by the bus master. It resembles the counterpart of the OWMS register. The upper four bits of the OWUS are not assigned to any function. The lower bits, STAT0 to STAT3, are used to pass bit-level information to control the program flow within the microcomputer. The details on the use of these bits are explained in the "Crypto iButton Firmware Reference Manual."

OWMS REGISTER

7	6	5	4	3	2	1	0
X	X	X	X	STAT3	STAT2	STAT1	STAT0

After the command code for Write Status the bus master transmits the new value of the OWUS byte. Now the bus master receives the inverted CRC16 of command code and the OWUS byte transmitted.

The bus master now calculates the CRC and compares it with the CRC value received from the DS1957. If the CRC values do not match, the data transfer was not successful and the Write Status command sequence needs to be repeated. If the CRC values match, the bus master has to transmit the release sequence 517FH to signal the microcomputer inside the DS1957 the arrival of new data. As a confirmation, the bus master will receive all 0s if the release sequence was accepted by the DS1957. If the bus master receives 1s instead, the release sequence was not accepted and the command sequence needs to be repeated.

1-WIRE BUS SYSTEM

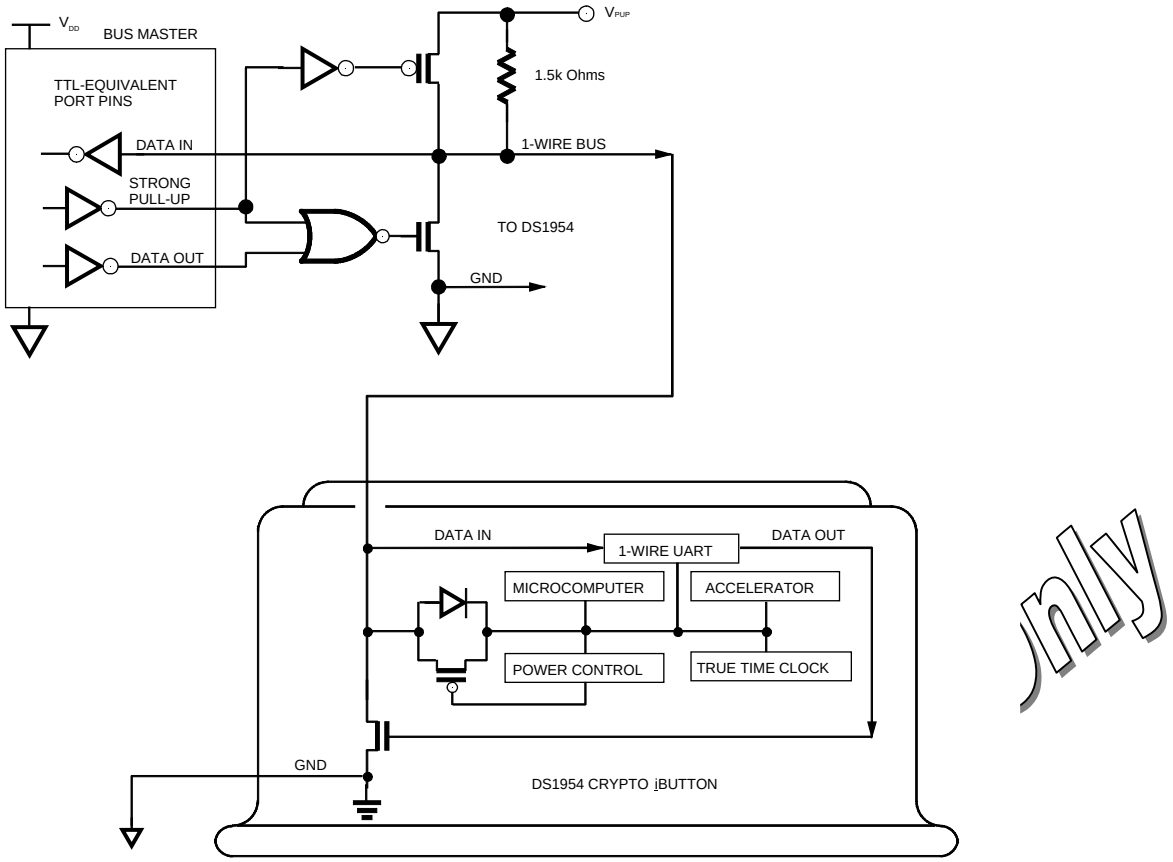
The 1-Wire bus is a system which has a single bus master and one or more slaves. In all instances, the DS1957 is a slave device. The bus master is typically a microcontroller or an IBM-compatible PC. The discussion of this bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal type and timing). A 1-Wire protocol defines bus transactions in terms of the bus state during specified time slots that are initiated on the falling edge of sync pulses from the bus master. For a more detailed protocol description, refer to Chapter 4 of the "Book of DS19xx iButton Standards."

HARDWARE CONFIGURATION

The 1-Wire bus has only a single line by definition; it is important that each device on the bus be able to drive it at the appropriate time. To facilitate this, each device attached to the 1-Wire bus must have an open drain connection or 3-state outputs. The DS1957 is an open drain part. The bus master requires a pull-up resistor at the master end of the bus. Both, the bus master equivalent circuit and the equivalent circuit of the DS1957's 1-Wire interface are shown in Figure 5. The value of the pull-up resistor should be approximately 1.5 kΩ for short line lengths.

A multidrop bus consists of a 1-Wire bus with multiple slaves attached. At regular speed the 1-Wire bus has a maximum data rate of 16.3 kbits per second. The speed can be boosted to 142 kbits per second by activating the Overdrive mode. The idle state for the 1-Wire bus is high. If, for any reason, a transaction needs to be suspended, the bus MUST be left in the idle state if the transaction is to resume. If this does not occur and the bus is left low for more than 16 µs (overdrive speed) or more than 120 µs (regular speed), one or more of the devices on the bus may be reset.

HARDWARE CONFIGURATION Figure 5



TRANSACTION SEQUENCE

The sequence for accessing the DS1957 via the 1-Wire port is as follows:

· Initialization
· ROM Function Command
· Data Transfer or Control Function Command
· Read/Write Register/Buffer or Control

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 a presence pulse(s) transmitted by the slave(s).

The presence pulse lets the bus master know that the DS1957 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 six ROM function commands. All ROM function commands are eight 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 DS1957's 8-bit family code, unique 48-bit serial number, and 8-bit CRC. This command can be used only if there is a single DS1957 on the bus. If more than one slave is present on the bus, a data collision will occur when all slaves try to transmit at the same time (open drain will produce a wired-AND result). The resultant family code and 48-bit serial number will usually result in a mismatch of the CRC.

MATCH ROM [55H]

The match ROM command, followed by a 64-bit ROM sequence, allows the bus master to address a specific DS1957 on a multidrop bus. Only the DS1957 that exactly matches the 64-bit ROM sequence will respond to the subsequent 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 pull-downs 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. 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. See Chapter 5 of the "Book of DS19xx iButton Standards" for a comprehensive discussion of a ROM search, including an actual example.

OVERDRIVE SKIP ROM [3CH]

On a single-drop bus this command can save time by allowing the bus master to access the memory functions without providing the 64-bit ROM code. Unlike the normal Skip ROM command the Overdrive Skip ROM sets the DS1957 in the Overdrive Mode (OD=1). All communication following this command has to occur at Overdrive Speed until a reset pulse of minimum 480 µs duration resets all devices on the bus to regular speed (OD=0).

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

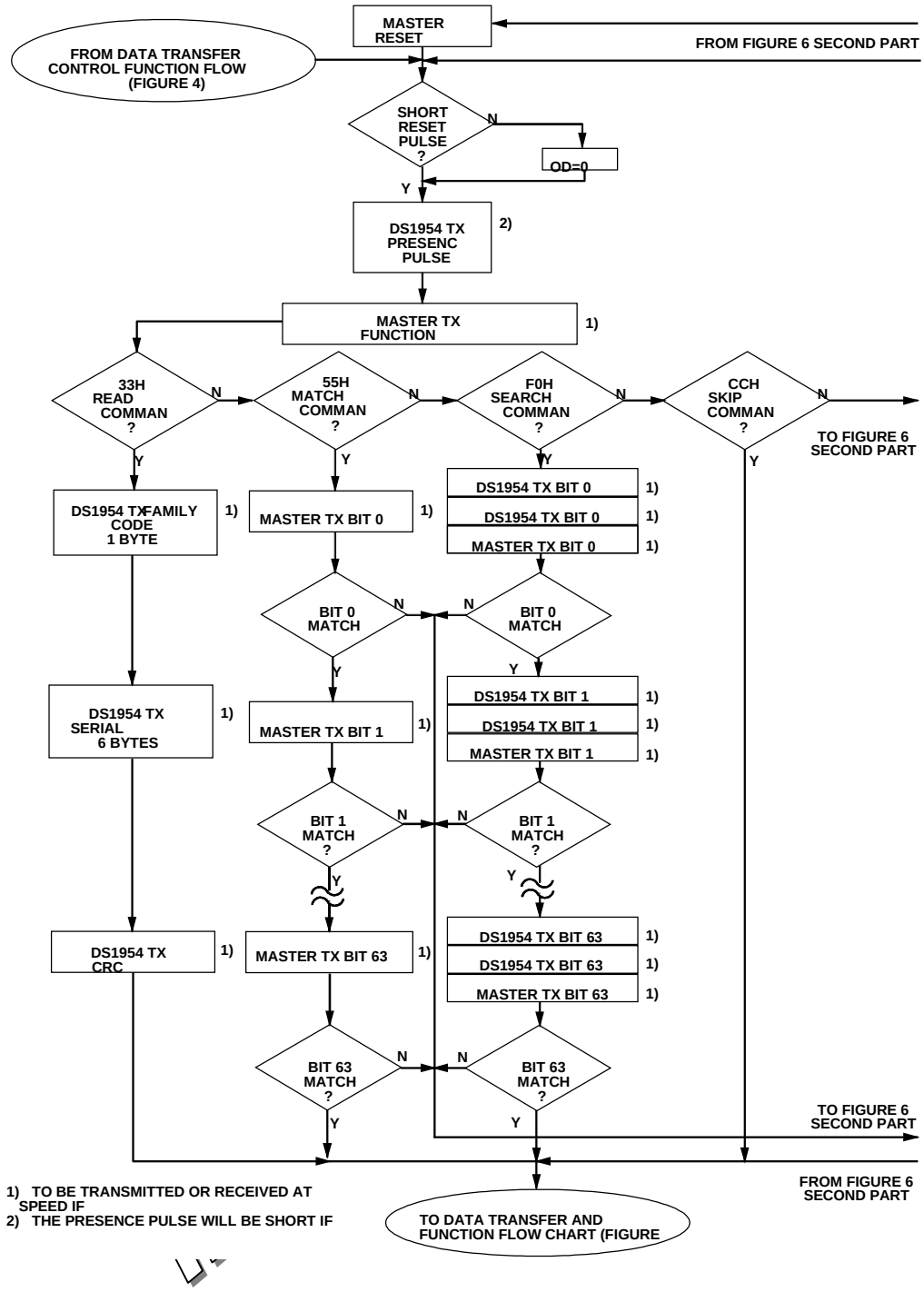
OVERDRIVE MATCH ROM [69H]

The Overdrive Match ROM command, followed by a 64-bit ROM sequence transmitted at Overdrive Speed, allows the bus master to address a specific DS1957 on a multidrop bus and to simultaneously set it in Overdrive Mode. Only the DS1957 that exactly matches the 64-bit ROM sequence will respond to the subsequent memory function command. Slaves already in Overdrive mode from a previous Overdrive Skip or Match command will remain in Overdrive mode. All other slaves that do not match the 64-bit ROM sequence or do not support Overdrive will return to or remain at regular speed and wait for a reset pulse of minimum 480 µs duration. The Overdrive Match ROM command can be used with a single or multiple devices on the bus.

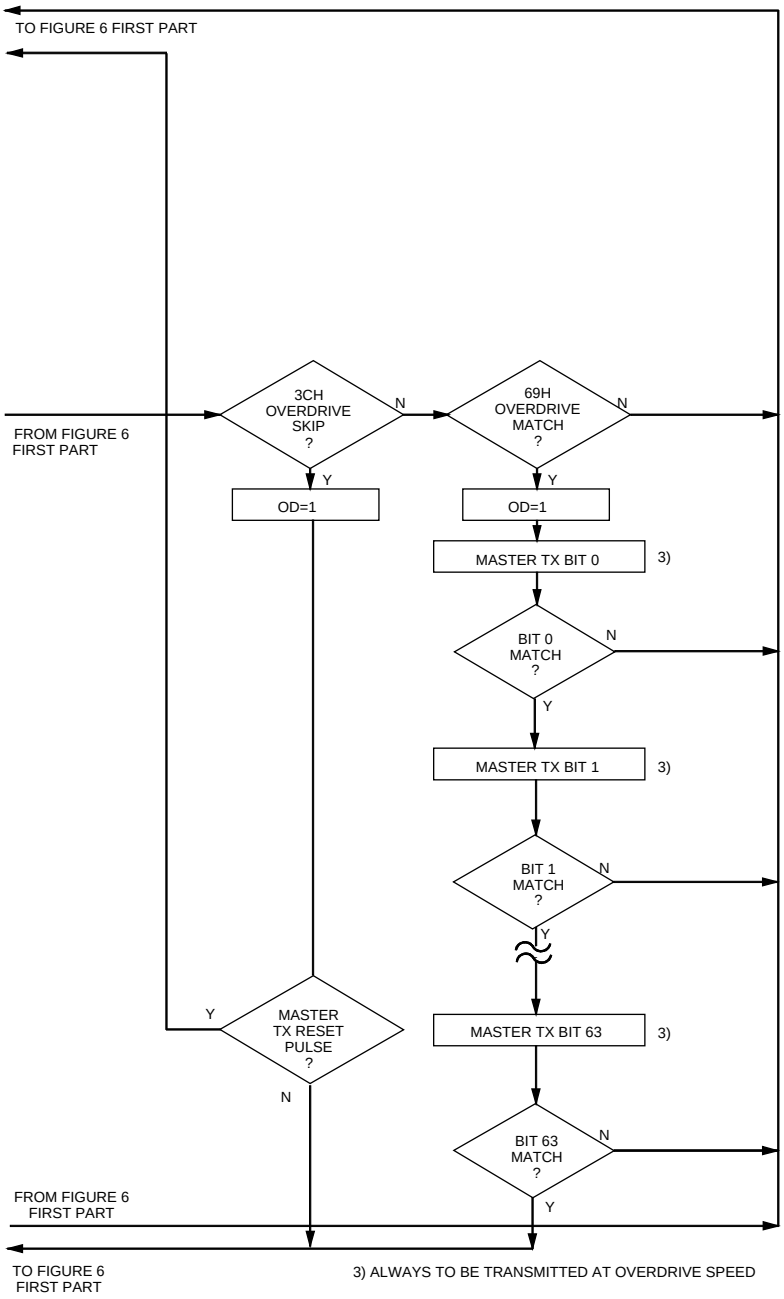
1-WIRE SIGNALING

The DS1957 requires strict protocols to insure data integrity. The protocol consists of five types of signaling on one line: Reset Sequence with Reset Pulse and Presence Pulse, Write 0, Write 1, Read Data and Strong Pull-up. All these signals except presence pulse are initiated by the bus master. The DS1957 can communicate at two different speeds, regular speed and Overdrive Speed. If not explicitly set into the Overdrive Mode, the DS1957 will communicate at regular speed. While in Overdrive Mode the fast timing applies to all communication-related wave forms.

ROM FUNCTIONS FLOW CHART Figure 6



ROM FUNCTIONS FLOW CHART Figure 6 (cont'd)



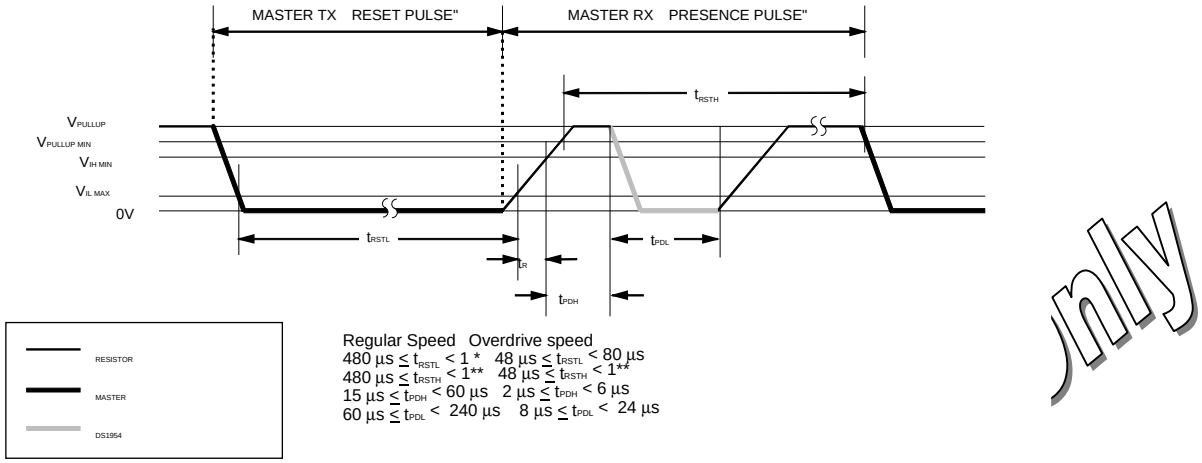
The initialization sequence required to begin any communication with the DS1957 is shown in Figure 7. A Reset Pulse followed by a Presence Pulse indicates the DS1957 is ready to accept a ROM command. The bus master transmits (TX) a reset pulse ($t_{RSTL,}$ minimum 480 µs at regular speed, 48 µs at Overdrive Speed). The bus master then releases the line and goes into receive mode (RX). The 1-Wire bus is pulled to a high state via the pull-up resistor. After detecting the rising edge on the data pin, the DS1957 waits (t_{PDH}, 15-60 µs at regular speed, 2-6 µs at overdrive speed) and then transmits the presence pulse (t_{PDL}, 60-240 µs at regular speed, 8-24 µs at Overdrive Speed).

A Reset Pulse of 480 µs or longer will exit the Overdrive Mode returning the device to regular speed. If the DS1957 is in Overdrive Mode and the Reset Pulse is no longer than 80 µs the device will remain in Overdrive Mode.

READ/WRITE TIME SLOTS

The definitions of write and read time slots are illustrated in Figure 8. All time slots are initiated by the master driving the data line low. The falling edge of the data line synchronizes the DS1957 to the master by triggering a delay circuit in the DS1957. During write time slots, the delay circuit determines when the DS1957 will sample the data line. For a read data time slot, if a "0" is to be transmitted, the delay circuit determines how long the DS1957 will hold the data line low overriding the 1 generated by the master. If the data bit is a "1", the DS1957 will leave the read data time slot unchanged.

INITIALIZATION PROCEDURE "RESET AND PRESENCE PULSES" Figure 7

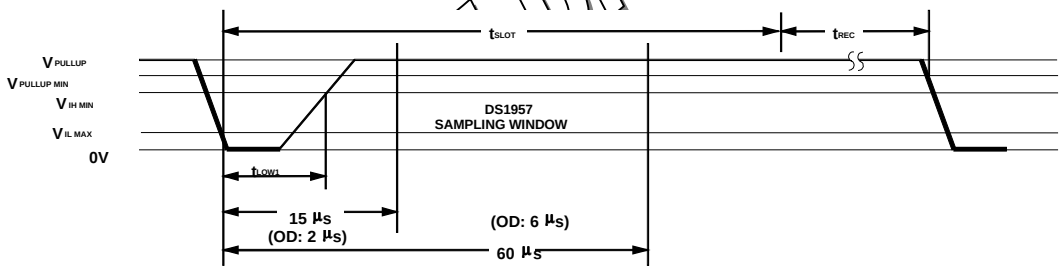


Regular Speed Overdrive speed
480 µs ≤ t_{RSTL} < 1 * 48 µs ≤ t_{RSTL} < 80 µs
480 µs ≤ t_{RSTH} < 1** 48 µs ≤ t_{RSTH} < 1**
15 µs ≤ t_{PDH} < 60 µs 2 µs ≤ t_{PDH} < 6 µs
60 µs ≤ t_{PDL} < 240 µs 8 µs ≤ t_{PDL} < 24 µs

* IN ORDER NOT TO MASK INTERRUPT SIGNALLING BY OTHER DEVICES ON THE 1-WIRE BUS, tRSTL + tR SHOULD ALWAYS BE LESS THAN 960 µs
** INCLUDES RECOVERY TIME

READ/WRITE TIMING DIAGRAM Figure 8

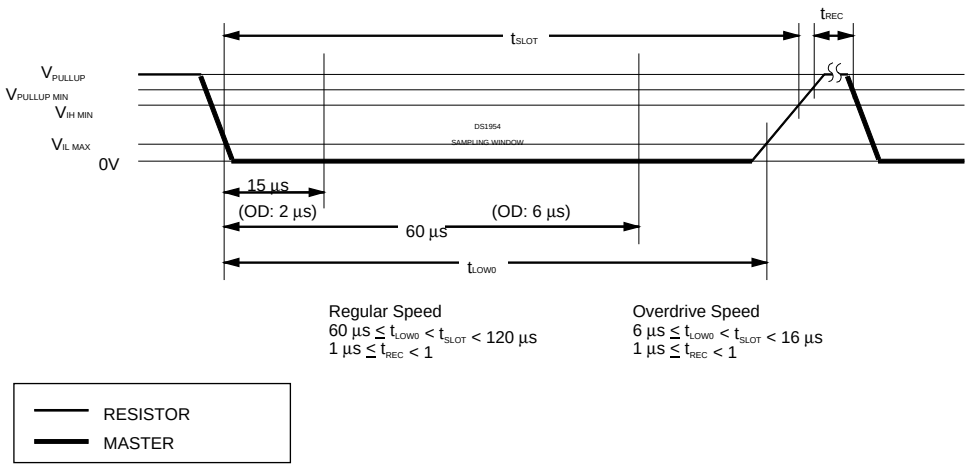
WRITE-ONE TIME SLOT



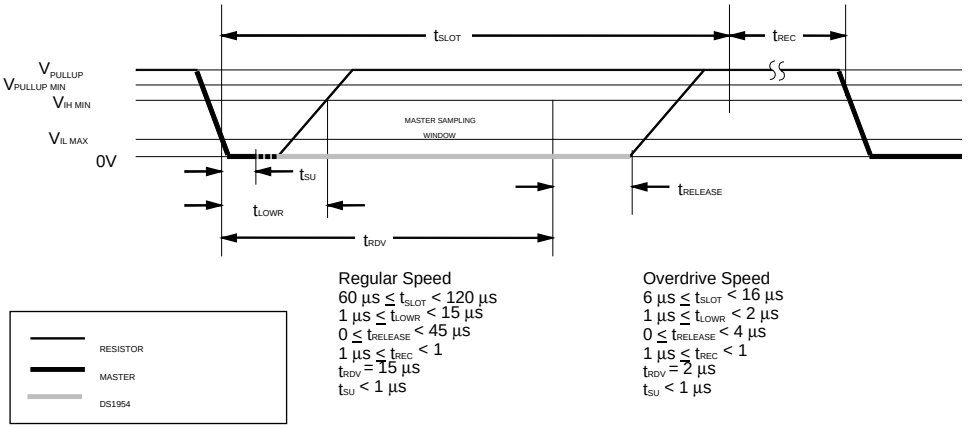
Regular Speed
60 µs ≤ t_{SLOT} < 120 µs
1 µs ≤ t_{LOW1} < 15 µs
1 µs ≤ t_{REC} < 1

Overdrive Speed
6 µs ≤ t_{SLOT} < 16 µs
1 µs ≤ t_{LOW1} < 2 µs
1 µs ≤ t_{REC} < 1

WRITE-ZERO TIME SLOT



READ-DATA TIME SLOT



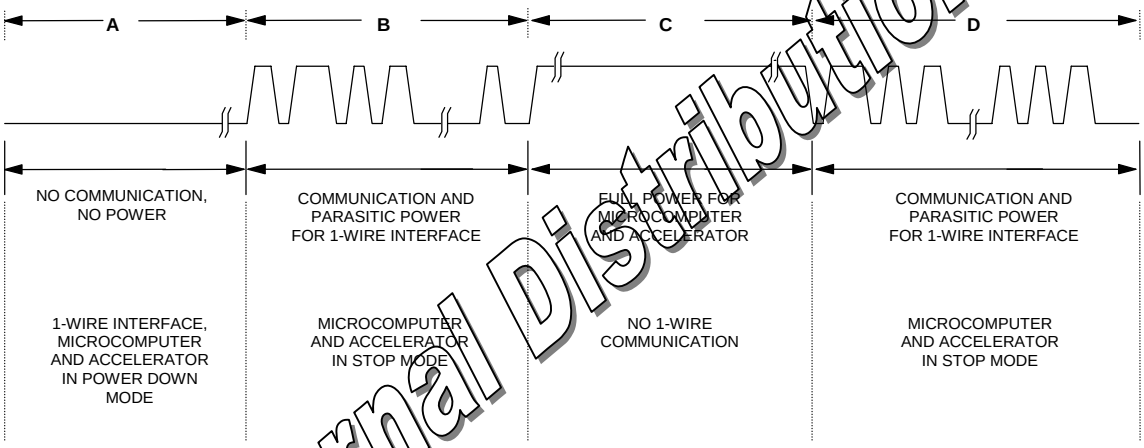
POWER TRANSFER

The DS1957 Crypto iButton communicates and gets power through the 1-Wire data line. Typically the DS1957 will see four different power situations on the 1-Wire bus: A) no power at all; B) parasitic supply for initial and repeated communication; C) full power for (repeated) times of computation; and D) parasitic supply for final communication prior to removal from the 1-Wire bus (Figure 9). During communication, the microcomputer and accelerator inside the DS1957 remain in a STOP mode (time period B and D). Once data has been transferred, the microcomputer and accelerator are enabled to process the data (period C). After the data processing has been completed the microcomputer and accelerator return to a low power STOP mode and the resulting data is then extracted through the 1-Wire interface by the bus master (period D). The amount of data and the required calculations associated with the data will govern the number of periods (B & C) that the master and the DS1957 will use to transfer and process data. The time associated with period C depends on the assigned task. When the microcomputer is loading data through the 1-Wire interface, the time will be short. When the microcomputer and accelerator are being used to perform calculations the time for period C will be significantly longer. During 1-Wire communication, the DS1957 gets the power through the 1-Wire pull-up resistor of 1.5 kΩ (see Figure 5). A diode plus a capacitor on chip form a parasitic power supply

which is sufficient to operate the 1-Wire interface and to maintain the STOP mode of the microcomputer and accelerator. Once the initial data is loaded through the 1-Wire interface, the bus master will send either a Start Program or Continue Program command to the DS1957. Either of these commands will turn on the internal P-Channel bypass to the diode in the parasitic power supply and enable the system power controls to clear the STOP mode once the internal supply voltage rises above a certain trip point. In addition to sending the Start Program or Continue Program command, the bus master also has to activate the P-Channel transistor in its 1-Wire interface to bypass the 1-Wire pull-up resistor (see Data Transfer and Control Function Flow Charts and command descriptions for details). Once the STOP mode is cleared, the microcomputer and/or accelerator will begin to operate (period C). At a predetermined time, the microcomputer and accelerator will both re-enter the STOP mode and wait for the bus master to start communication.

During the time where power is supplied through the 1-Wire bus, it is impossible for the bus master and the DS1957 to communicate. To inform each other about the progress of a computation, for example, both the Master and the DS1957 have to agree on an accurate value of the time for computations. This value is mutually agreed on during the initial data exchange and is based on the synchronization of the True Time Clock in the DS1957 and the Real Time Clock in the bus master. By synchronizing to crystal based clocks in the bus master and DS1957, it is possible to guarantee that the DS1957 will place the microcomputer and accelerator into a STOP mode before the bus master begins the next data communication.

POWER SITUATIONS ON THE 1-WIRE BUS Figure 9



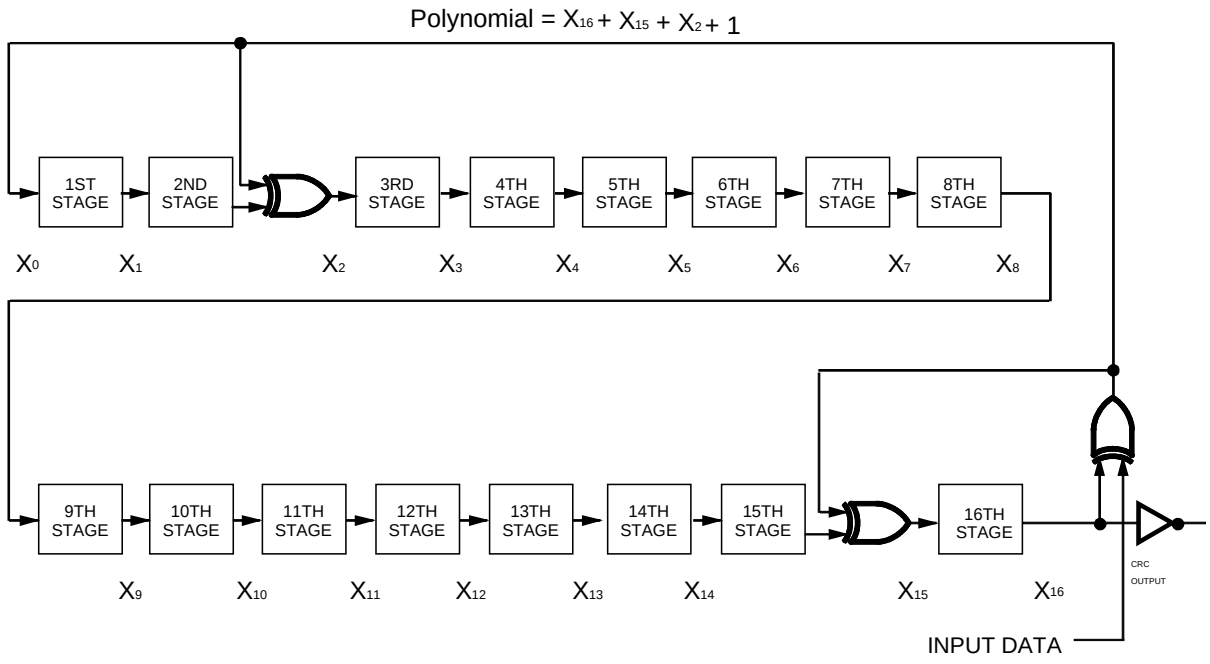
CRC GENERATION

With the DS1957 there are two different types of CRCs (Cyclic Redundancy Checks). One CRC is an 8-bit type and is 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 DS1957 to determine if the ROM data has been received error-free by the bus master. The equivalent polynomial function of this CRC is: $X^8 + X^5 + X^4 + 1$. This 8-bit CRC is received in the true (non-inverted) form when reading the ROM of the DS1957. It is computed at the factory and lasered into the ROM.

The other CRC is a 16-bit type, generated according to the standardized CRC16-polynomial function $x^{16} + x^{15} + x^2 + 1$. This CRC is used for error detection when reading/writing the Intermediate Product Register (IPR), I/O Buffer and Status Memory for fast verification of data integrity. It is the same type of CRC as is used with NV RAM based iButtons for error detection within the iButton Extended File Structure. In contrast to the 8-bit CRC, the 16-bit CRC is always returned or sent in the complemented (inverted) form. A CRC generator inside the DS1957 chip (Figure 10) will calculate a new 16-bit CRC as shown in the command flow chart of Figure 4. The bus master compares the CRC value read from the device to the one it calculates from the data and decides whether to continue with an operation or to start all over again.

In every case, the 16-bit CRC value is the result of shifting the command byte into the cleared CRC generator, followed by the control and/or data bytes as shown in the Data Transfer and Control Function Flow Chart. For more details on generating CRC values including example implementations in both hardware and software, see the "Book of DS19xx iButton Standards."

CRC-16 HARDWARE DESCRIPTION AND POLYNOMIAL Figure 10



PHYSICAL SPECIFICATIONS

Size	See mechanical drawing
Weight	3.3 grams
Humidity	90% RH at 50°C
Altitude	10,000 feet
Expected Service Life	5 years at 25°C

ABSOLUTE MAXIMUM RATINGS*

Voltage on DATA to Ground	-0.5V to +6.0V
Operating Temperature	-20°C to +70°C
Storage Temperature	-20°C to +70°C

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

DC ELECTRICAL CHARACTERISTICS

(V_{PUP}=4.0V to 5.5V; -20°C to +70°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Logic 1	V_{IH}	2.2			V	1
Logic 0	V_{IL}	-0.3		+0.6	V	1
Output Logic Low @4 mA	V_{OL}			0.4	V	1
Output Logic High	V_{OH}		V_{PUP}	5.5	V	1, 2
Input Load Current Idle	I_{LI}		100		µA	3
Input Load Current Computing	I_{LC}		20		mA	3

CAPACITANCE

(t_A = 25°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
I/O (1-Wire)	$C_{IN/OUT}$			150	nF	6

AC ELECTRICAL CHARACTERISTICS REGULAR SPEED

(V_{PUP}=4.0V to 5.5V; -20°C to +70°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Time Slot	t_{SLOT}	60		120	µs	
Write 1 Low Time	t_{LOW1}	1		15	µs	
Write 0 Low Time	t_{LOW0}	60		120	µs	
Read Low Time	t_{LOWR}	1		15	µs	
Read Data Valid	t_{RDV}		exactly 15		µs	
Release Time	$t_{RELEASE}$	0	15	45	µs	
Read Data Setup	t_{SU}			1	µs	5
Recovery Time	t_{REC}	1			µs	
Reset Time High	t_{RSTH}	480			µs	4
Reset Time Low	t_{RSTL}	480			µs	7
Presence Detect High	t_{PDHIGH}	15		60	µs	
Presence Detect Low	t_{PDLOW}	60		240	µs	

AC ELECTRICAL CHARACTERISTICS
OVERDRIVE SPEED

(V_{PUP}=4.0V to 5.5V; -20°C to +70°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Time Slot	t_{SLOT}	6		16	µs	
Write 1 Low Time	t_{LOW1}	1		2	µs	
Write 0 Low Time	t_{LOW0}	6		16	µs	
Read Low Time	t_{LOWR}	1		2	µs	
Read Data Valid	t_{RDV}		exactly 2		µs	
Release Time	$t_{RELEASE}$	0	1.5	4	µs	
Read Data Setup	t_{SU}			1	µs	5
Recovery Time	t_{REC}	1			µs	
Reset Time High	t_{RSTH}	48			µs	4
Reset Time Low	t_{RSTL}	48		80	µs	
Presence Detect High	t_{PDHIGH}	2		6	µs	
Presence Detect Low	t_{PDLOW}	8		24	µs	

NOTES:

1. All voltages are referenced to ground.
2. V_{PUP} = external pull-up voltage.
3. Input load is to ground.
4. An additional reset or communication sequence cannot begin until the reset high time has expired.
5. Read data setup time refers to the time the host must pull the 1-Wire bus low to read a bit. Data is guaranteed to be valid within 1 µs of this falling edge.
6. Capacitance on the data pin could be 150 nF when power is first applied. If a 1.5 kΩ resistor is used to pull up the data line to V_{CC}, 500 µs after power has been applied, the parasite capacitance will not affect normal communications.
7. The reset low time (t_{RSTL}) should be restricted to a maximum of 960 µs, to allow interrupt signaling, otherwise, it could mask or conceal interrupt pulses.