

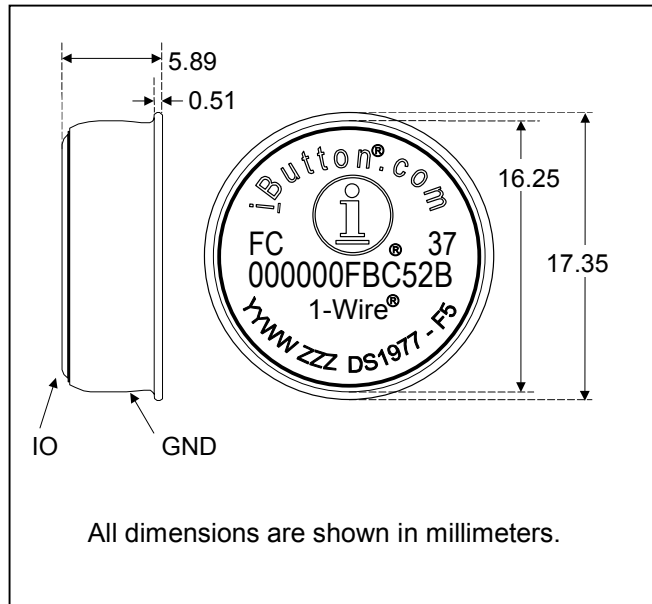
### iButton DESCRIPTION

The DS1977 is a 32kB EEPROM in a rugged, iButton® enclosure. Access to the memory can be password-protected with different passwords for read-only and full access. Data is transferred serially through the 1-Wire® protocol, which requires only a single data lead and a ground return. Every DS1977 is factory lasered with a guaranteed unique 64-bit registration number that allows for absolute traceability. The durable stainless-steel iButton package is highly resistant to environmental hazards such as dirt, moisture, and shock. Accessories permit the DS1977 iButton to be mounted on almost any object, including containers, pallets, and bags.

### APPLICATIONS

Maintenance/Inspection Data Storage  
 Medical Data Carrier  
 Health Data Carrier  
 Audit Data Storage and Carrier

### F5 MicroCAN



*iButton, 1-Wire, and MicroCAN are registered trademarks of Dallas Semiconductor.*

### SPECIAL FEATURES

- 32kB EEPROM Organized as Pages of 64 Bytes Each
- Optional Password Protection with Different 64-Bit Passwords for Read and Full Access
- Communicates to Host with a Single Digital Signal at Up to 15.3kbps at Standard Speed or Up to 125kbps in Overdrive Mode Using 1-Wire Protocol
- Operating Range: 2.8V to 5.25V, -40°C to +85°C
- Minimum 100k Write Cycles Endurance
- 15kV Built-in ESD Protection

### COMMON iButton FEATURES

- Unique Factory-Lasered 64-Bit Registration Number Assures Error-Free Device Selection and Absolute Traceability Because No Two Parts are Alike
- Built-In Multidrop Controller for 1-Wire Net
- Chip-Based Data Carrier Stores Digital Identification and Information, Armored in a Durable Stainless-Steel Case
- Data can be Accessed While Affixed to Object
- Button Shape is Self-Aligning with Cup-Shaped Probes
- Easily Affixed with Self-Stick Adhesive Backing, Latched by its Flange, or Locked with a Ring Pressed onto its Rim
- Presence Detector Acknowledges when Reader First Applies Voltage
- Designed to meet UL#913 (4th Edit.); Intrinsically Safe Apparatus: Under Entity Concept for use in Class I, Division 1, Group A, B, C, and D Locations, contact Dallas Semiconductor for certification schedule

### ORDERING INFORMATION

PART	TEMP RANGE	PACKAGE
DS1977-F5	-40°C to +85°C	F5 iButton

### EXAMPLES OF ACCESSORIES

PART	DESCRIPTION
DS9096P	Self-Stick Adhesive Pad
DS9101	Multipurpose Clip
DS9093RA	Mounting Lock Ring
DS9093A	Snap-In Fob
DS9092	iButton Probe

**Note:** Some revisions of this device may incorporate deviations from published specifications known as errata. Multiple revisions of any device may be simultaneously available through various sales channels. For information about device errata, click here: [www.maxim-ic.com/errata](http://www.maxim-ic.com/errata).

## PHYSICAL SPECIFICATION

Size  
Weight DS1977

See mechanical drawing  
Ca. 3.3g

## ABSOLUTE MAXIMUM RATINGS

I/O Voltage to GND	-0.3V, +5.5V
I/O Sink Current	20mA
Junction Temperature	+150°C
Storage Temperature Range	-40°C to +85°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to the absolute maximum rating conditions for extended periods may affect device.

## ELECTRICAL CHARACTERISTICS

( $V_{PUP} = 2.8V$  to  $5.25V$ ,  $T_A = -40^\circ C$  to  $+85^\circ C$ .)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
<b>I/O Pin General Data</b>						
1-Wire Pullup Resistance	$R_{PUP}$	(Notes 1, 2)	0.6		2.2	$k\Omega$
Input Capacitance	$C_{IO}$	(Note 3)			5	nF
Input Load Current	$I_L$	I/O pin at $V_{PUP}$	1		10	$\mu A$
High-to-Low Switching Threshold	$V_{TL}$	(Notes 4, 5)	0.5		3.2	V
Input Low Voltage	$V_{IL}$	(Notes 1, 6)			0.30	V
Low-to-High Switching Threshold	$V_{TH}$	(Notes 4, 7)	0.7		3.4	V
Switching Hysteresis	$V_{HY}$	(Note 8)	0.15		N/A	V
Output-Low Voltage at 4mA	$V_{OL}$	(Note 9)			0.4	V
Recovery Time	$t_{REC}$	Standard speed, $R_{PUP} = 2.2k\Omega$ (Note 1)	5			$\mu s$
		Overdrive speed, $R_{PUP} = 2.2k\Omega$ (Note 1)	2			
		Overdrive speed, directly prior to reset pulse; $R_{PUP} = 2.2k\Omega$ (Note 1)	5			
Rising-Edge Hold-off Time	$t_{REH}$	Standard speed (Note 10)	0.5		5	$\mu s$
		Overdrive speed (Note 10)	0.5		2	
Timeslot Duration	$t_{SLOT}$	Standard speed (Note 1)	65			$\mu s$
		Overdrive speed (Note 1)	8			
<b>I/O Pin, 1-Wire Reset, Presence Detect Cycle</b>						
Reset Low Time	$t_{RSTL}$	Standard speed (Note 1)	480		640	$\mu s$
		Overdrive Speed (Note 1)	48		80	
Presence Detect High Time	$t_{PDH}$	Standard speed (Note 11)	15		60	$\mu s$
		Overdrive speed (Note 11)	2.5		6.5	
Presence Detect Fall Time	$t_{FPD}$	Standard speed, $V_{PUP} > 4.5V$ (Note 12)	1.5		5	$\mu s$
		Standard speed (Note 12)	1.5		8	
		Overdrive speed (Note 12)	0.15		1	
Presence Detect Low Time	$t_{PDL}$	Standard speed	60		240	$\mu s$
		Overdrive speed	8		24	

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Presence Detect Sample Time	$t_{MSP}$	Standard speed, $V_{PUP} > 4.5V$ (Note 1)		65		75	$\mu s$
		Standard speed (Note 1)		68		75	
		Overdrive speed (Note 1)		7.5		10.5	
<b>I/O Pin, 1-Wire Write</b>							
Write-0 Low Time	$t_{W0L}$	Standard speed (Note 1)		60		120	$\mu s$
		Overdrive speed (Note 1)		6		16	
Write-1 Low Time	$t_{W1L}$	Standard speed (Notes 1, 13)		5		$15 - \varepsilon$	$\mu s$
		Overdrive speed (Notes 1, 13)		1		$2 - \varepsilon$	
<b>I/O Pin, 1-Wire Read</b>							
Read Low Time	$t_{RL}$	Standard speed (Notes 1, 14)		5		$15 - \delta$	$\mu s$
		Overdrive speed (Notes 1, 14)		1		$2 - \delta$	
Read Sample Time	$t_{MSR}$	Standard speed, $V_{PUP} > 4.5V$ (Notes 1, 14)		$t_{RL} + \delta$		20	$\mu s$
		Standard speed (Notes 1, 14)		$t_{RL} + \delta$		15	
		Overdrive speed (Notes 1, 14)		$t_{RL} + \delta$		2	
<b>I/O Pin, Strong Pullup</b>							
Strong Pullup Read	$t_{SPUR}$	(Note 1)		2.64			ms
Strong Pullup Write	$t_{SPUW}$	(Note 1)		22.46			ms
Strong Pullup password verification	$t_{SPUV}$	(Note 1)		0.62			ms
<b>EEPROM</b>							
Programming Current	$I_{LPROG}$					7	mA
Write/Erase Cycles	$N_{CYCLE}$			100k			—
Data Retention	$t_{RET}$			10			years

- Note 1:** System requirement.
- Note 2:** Maximum allowable pullup resistance is a function of the number of 1-Wire devices in the system and 1-Wire recovery times. The specified value here applies to systems with only one device and with the minimum 1-Wire recovery times. For more heavily loaded systems, an active pullup such as that found in the DS2480 may be required.
- Note 3:** Capacitance on the data pin could be 5nF when power is first applied.
- Note 4:**  $V_{TL}$ ,  $V_{TH}$  are a function of the internal supply voltage.
- Note 5:** Voltage below which, during a falling edge on I/O, a logic '0' is detected.
- Note 6:** The voltage on I/O needs to be less or equal to  $V_{ILMAX}$  whenever the master drives the line low.
- Note 7:** Voltage above which, during a rising edge on I/O, a logic '1' is detected.
- Note 8:** After  $V_{TH}$  is crossed during a rising edge on I/O, the voltage on I/O has to drop by  $V_{HY}$  to be detected as logic '0'.
- Note 9:** The I-V characteristic is linear for voltages less than 1V.
- Note 10:** The earliest recognition of a negative edge is possible at  $t_{REH}$  after  $V_{TH}$  has been reached before.
- Note 11:** Highlighted numbers are NOT in compliance with the published iButton standards. See comparison table below.
- Note 12:** Interval during the negative edge on I/O at the beginning of a Presence Detect pulse between the time at which the voltage is 90% of  $V_{PUP}$  and the time at which the voltage is 10% of  $V_{PUP}$ .
- Note 13:**  $\varepsilon$  represents the time required for the pullup circuitry to pull the voltage on I/O up from  $V_{IL}$  to  $V_{TH}$ .
- Note 14:**  $\delta$  represents the time required for the pullup circuitry to pull the voltage on I/O up from  $V_{IL}$  to the input-high threshold of the bus master.

Parameter Name	Standard Values				DS1977 Values			
	Standard Speed		Overdrive Speed		Standard Speed		Overdrive Speed	
	min	max	min	max	min	max	min	max
$t_{SLOT}$ (incl. $t_{REC}$ )	61 $\mu s$	(undef.)	7 $\mu s$	(undef.)	65 $\mu s$ <sup>1)</sup>	(undef.)	8 $\mu s$ <sup>1)</sup>	(undef.)
$t_{RSTL}$	480 $\mu s$	(undef.)	48 $\mu s$	80 $\mu s$	480 $\mu s$	640 $\mu s$	48 $\mu s$	80 $\mu s$
$t_{PDH}$	15 $\mu s$	60 $\mu s$	2 $\mu s$	6 $\mu s$	15 $\mu s$	60 $\mu s$	2.5 $\mu s$	6.5 $\mu s$
$t_{PDL}$	60 $\mu s$	240 $\mu s$	8 $\mu s$	24 $\mu s$	60 $\mu s$	240 $\mu s$	8 $\mu s$	24 $\mu s$
$t_{W0L}$	60 $\mu s$	120 $\mu s$	6 $\mu s$	16 $\mu s$	60 $\mu s$	120 $\mu s$	6 $\mu s$	16 $\mu s$

1) Intentional change, longer recovery time requirement due to modified 1-Wire front end.

## APPLICATION

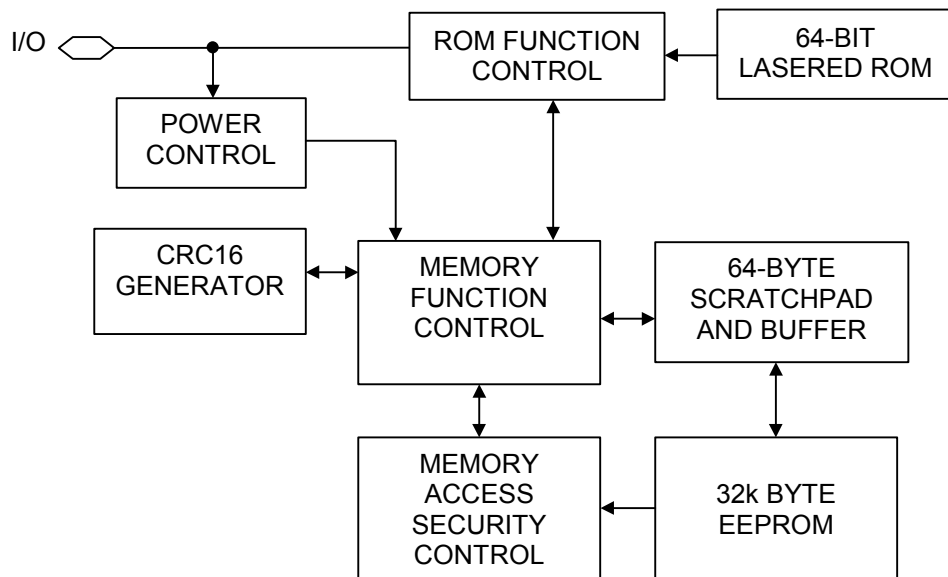
The DS1977 is an ideal device to store maintenance and inspection data of equipment or medical- and health-related data in digitally readable format. Due to its small size and rugged enclosure the device can be carried with a keyring to provide critical data in case of an emergency. The DS1977 can also serve as data shuttle to transport fleet management and vending machine data to an access point for upload into a remote server for further processing. Software for communication with the DS1977 is available for free download from the [iButton](#) website.

## OVERVIEW

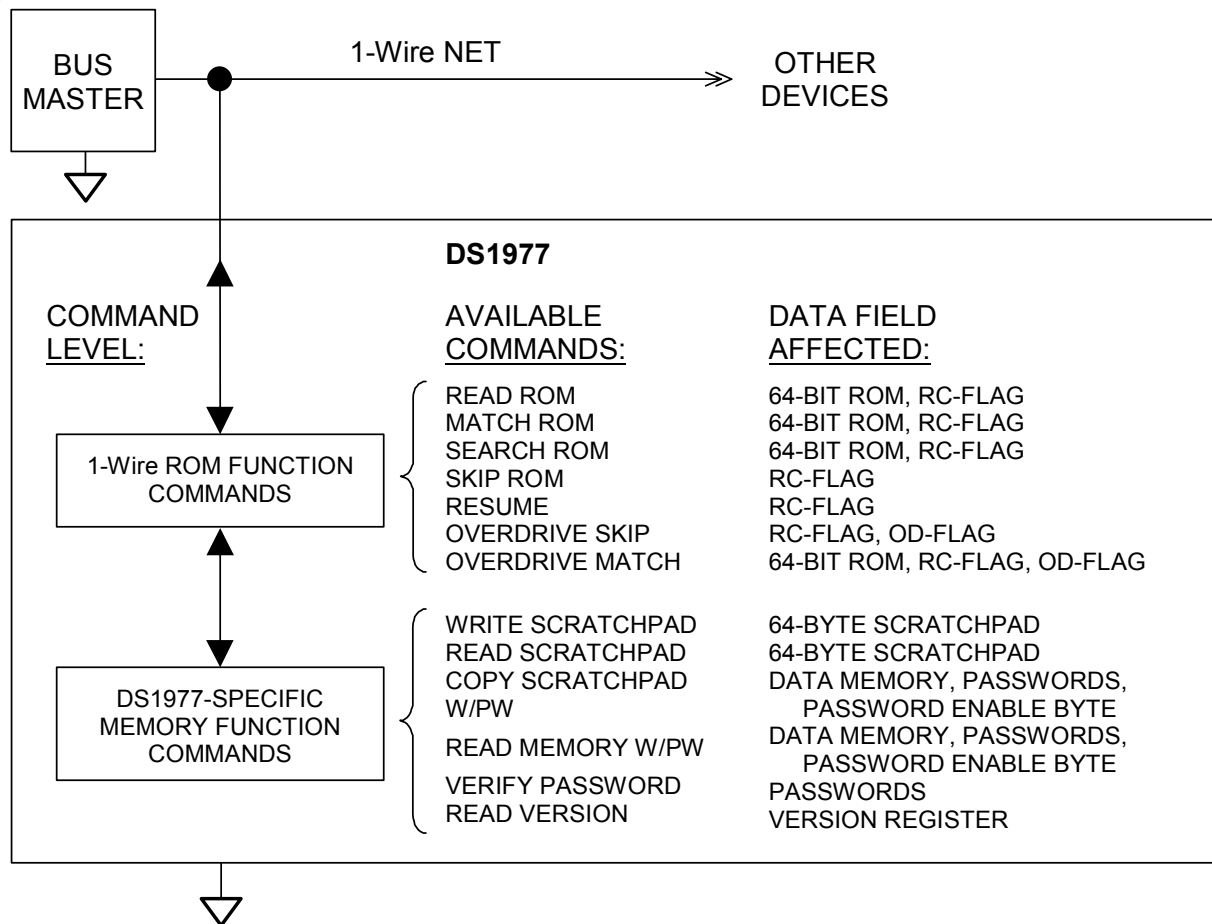
The block diagram in Figure 1 shows the relationships between the major control and memory sections of the DS1977. The device has four main data components: 1) 64-bit lasered ROM, 2) 512-bit scratchpad and buffer, 3) 32kB EEPROM, and 4) two password buffers. The passwords can only be written and verified, but never be read.

The hierarchical structure of the 1-Wire protocol is shown in Figure 2. The bus master must first provide one of the seven ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, 5) Overdrive-Skip ROM, 6) Overdrive-Match ROM or 7) Resume. Upon completion of an Overdrive ROM command byte executed at standard speed, the device will enter Overdrive mode, where all subsequent communication occurs at a higher speed. The protocol required for these ROM function commands is described in Figure 9. After a ROM function command is successfully executed, the memory and control functions become accessible and the master may provide any one of the six available commands. The protocol for these memory and control function commands is described in Figure 7. All data is read and written least significant bit first.

**Figure 1. DS1977 BLOCK DIAGRAM**



**Figure 2. HIERARCHICAL STRUCTURE FOR 1-WIRE PROTOCOL**

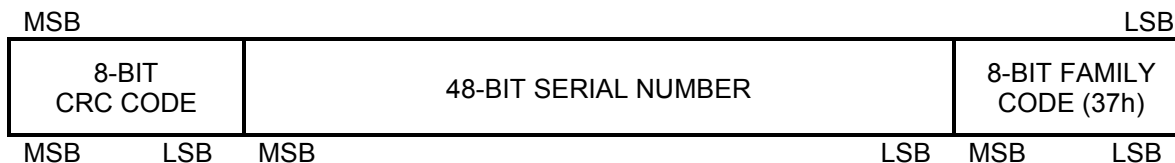


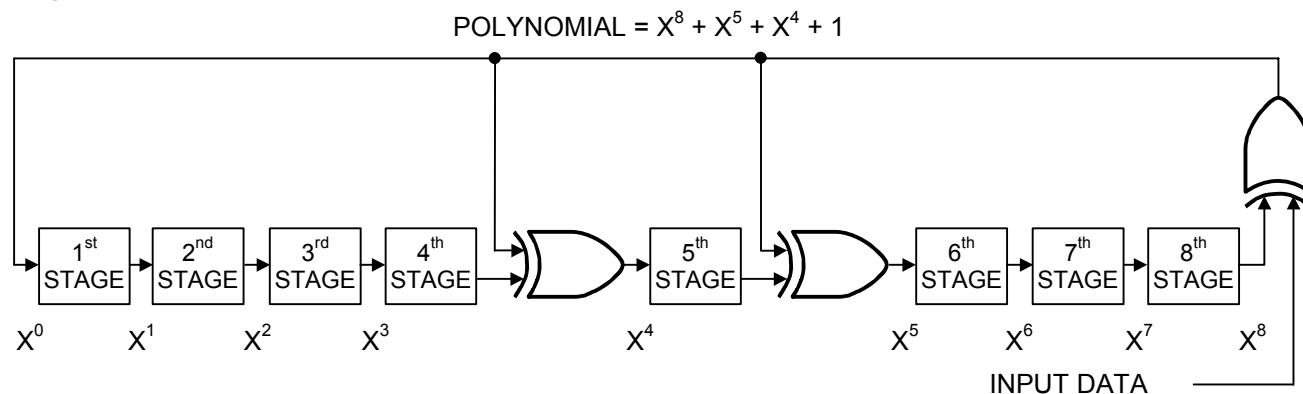
**64-BIT LASERED ROM**

Each DS1977 contains a unique ROM code that is 64 bits long. The first 8 bits are a 1-Wire family code. The next 48 bits are a unique serial number. The last 8 bits are a CRC of the first 56 bits. See Figure 3 for details. The 1-Wire CRC is generated using a polynomial generator consisting of a Shift and XOR gates as shown in Figure 4. The polynomial is  $X^8 + X^5 + X^4 + 1$ . Additional information about the Dallas 1-Wire Cyclic Redundancy Check is available in Application Note 27 and in the *Book of DS19xx iButton Standards*.

The Shift register bits are initialized to 0. Then starting with the least significant bit of the family code, one bit at a time is shifted in. After the 8<sup>th</sup> bit of the family code has been entered, then the serial number is entered. After the 48<sup>th</sup> bit of the serial number has been entered, the Shift register contains the CRC value. Shifting in the 8 bits of CRC returns the Shift register to all 0s.

**Figure 3. 64-BIT LASERED ROM**



**Figure 4. 1-WIRE CRC GENERATOR**

## MEMORY

The memory map of the DS1977 is shown in Figure 5. The 32kB of general-purpose EEPROM are located in pages 0 through 510. The passwords and the Password Control register take 17 bytes of page 511. The remaining bytes of page 511 are not accessible to the user. The scratchpad is an additional page that acts as a buffer when writing to the EEPROM memory or setting up a password, and when reading from the EEPROM.

**Figure 5. DS1977 MEMORY MAP**

64-Byte Intermediate Storage Scratchpad		
ADDRESS		
0000h to 003Fh	<b>64-Byte User EEPROM</b>	Page 0
0040h to 7F7Fh	<b>64-Byte User EEPROM</b>	Pages 1 To 509
7F80h to 7FBFh	<b>64-Byte User EEPROM</b>	Page 510
7FC0h to 7FC7h	<b>Read Access Password (A)</b>	
7FC8h to 7FCFh	<b>Full Access Password (B)</b>	
7FD0h	<b>Password Control Register</b>	
7FD1h to 7FFFh	<b>(No Function; Will Read FFh, Cannot be Written)</b>	

## SECURITY BY PASSWORD

The DS1977 is designed to use two passwords that control read access and full access. No password applies when reading from or writing to the scratchpad. Setting up a password or enabling/disabling the password checking is done in the same way as writing data to a memory location, only the address is different. Since they are located in the same memory page, both passwords can be redefined at the same time. Before changing passwords, disable passwords. When setting up a password, make sure that all 8 bytes of the password are defined. Otherwise the new password may be unknown. Always verify the scratchpad before issuing the copy scratchpad command. After a new password is successfully copied from the scratchpad to its memory location, erase the scratchpad by filling it with new data. Otherwise a copy of the password will remain accessible through the scratchpad until the DS1977 is disconnected from the 1-Wire line or undergoes a power-on reset.

## Read Access Password

This password only applies to the function "Read Memory with Password". If passwords are enabled (EPW = AAh, see Password Control register), the 64-bit data pattern that the 1-Wire master has to transmit with the command flow is compared to the passwords stored in the DS1977 iButton. The DS1977 delivers the requested data only if the password transmitted by the master was correct or if password checking is not enabled.

## Read Access Password Register

ADDR	b7	b6	b5	b4	b3	b2	b1	b0
7FC0h	RP7	RP6	RP5	RP4	RP3	RP2	RP1	RP0
7FC1h	RP15	RP14	RP13	RP12	RP11	RP10	RP9	RP8
—	—	—	—	—	—	—	—	—
7FC6h	RP55	RP54	RP53	RP52	RP51	RP50	RP49	RP48
7FC7h	RP63	RP62	RP61	RP60	RP59	RP58	RP57	RP56

There is only write access to this register. The Read Access Password needs to be transmitted exactly in the sequence RP0, RP1... RP62, RP63.

## Full Access Password

This password applies to the functions "Read Memory with Password" and "Copy Scratchpad with Password". If passwords are enabled (EPW = AAh, see Password Control register), the 64-bit data pattern that the 1-Wire master has to transmit with the command flow is compared to the passwords stored in the DS1977 iButton. The DS1977 executes the command only if the password transmitted by the master was correct or if password checking is not enabled.

## Full Access Password Register

ADDR	b7	b6	b5	b4	b3	b2	b1	b0
7FC8h	FP7	FP6	FP5	FP4	FP3	FP2	FP1	FP0
7FC9h	FP15	FP14	FP13	FP12	FP11	FP10	FP9	FP8
—	—	—	—	—	—	—	—	—
7FCEh	FP55	FP54	FP53	FP52	FP51	FP50	FP49	FP48
7FCFh	FP63	FP62	FP61	FP60	FP59	FP58	FP57	FP56

There is only write access to this register. The Full Access Password needs to be transmitted exactly in the sequence FP0, FP1... FP62, FP63.

## Password Control Register

The data pattern stored in the Password Control Register determines whether password checking is enabled. If password checking is enabled, the password transmitted is compared to the passwords stored in the device. Reading from or writing to the scratchpad does not require a password.

## Password Control Register Bitmap

ADDR	b7	b6	b5	b4	b3	b2	b1	b0
7FD0h	EPW							

## Register Details

BIT DESCRIPTION	BIT(S)	DEFINITION
EPW: Enable Passwords	b0 to b7	This byte enables or disables the password protection, which applies to reading from and writing to the memory except for the scratchpad. If the EPW bits form a pattern of 10101010 (AAh), the device will execute these commands only if the correct password is transmitted. The default pattern of EPW is different from AAh.

To enable password checking, the EPW bits need to form a binary pattern of 10101010 (AAh). If the EPW pattern is different from AAh, any password will be accepted, as long as it has a length of exactly 64 bits. Before enabling

passwords, check whether the new password has been successfully installed. See Verify Password command for details. Once enabled, changing the passwords or disabling password checking requires the knowledge of the current full-access password.

## VERSION REGISTER

The DS1977 includes a read-only Version register, which is not a component of the memory map. Therefore, a special command is used to read this register. The Chip Revision number enables application software to automatically use the appropriate software driver in case of different logical behavior.

### Version Register Bitmap

b7	b6	b5	b4	b3	b2	b1	b0
VER2	VER1	VER0	0	0	0	0	0

Bits 0 to 4 have no function. They always read 0.

### Register Details

BIT DESCRIPTION	BIT(S)	DEFINITION
(N/A)	b0 to b4	These bits are all 0.
VER: Chip Revision Indicator	b5 to b7	Chip revision code. The initial version of the DS1977 will have all revision bits set to 0.

## Figure 6. ADDRESS REGISTERS

Target Address (TA1)	T7	T6	T5	T4	T3	T2	T1	T0
Target Address (TA2)	T15	T14	T13	T12	T11	T10	T9	T8
Ending Address with Data Status (E/S) (Read Only)	AA	PF	E5	E4	E3	E2	E1	E0

## ADDRESS REGISTERS AND TRANSFER STATUS

Because of the serial data transfer, the DS1977 employs three address registers, called TA1, TA2, and E/S (Figure 6). Registers TA1 and TA2 must be loaded with the target address to which the data will be written or from which data will be sent to the master upon a Read command. Register E/S acts like a byte counter and Transfer Status register. It is used to verify data integrity with write commands. Therefore, the master only has read access to this register. The lower six bits of the E/S register indicate the address of the last byte that has been written to the scratchpad. This address is called Ending Offset. Bit 6 of the E/S register, called PF, is set if the number of data bits sent by the master is not an integer multiple of 8 or if the data in the scratchpad is not valid due to a loss of power. A valid write to the scratchpad will clear the PF bit. Note that the lowest six bits of the target address also determine the address within the scratchpad, where intermediate storage of data will begin. This address is called byte offset. If the target address for a Write command is 103Ch for example, then the scratchpad will store incoming data beginning at the byte offset 3Ch and will be full after only four bytes. The corresponding ending offset in this example is 3Fh. For best economy of speed and efficiency, the target address for writing should point to the beginning of a new page, i.e., the byte offset will be 0. Thus the full 64-byte capacity of the scratchpad is available, resulting also in the ending offset of 3Fh. However, it is possible to write one or several contiguous bytes somewhere within a page. The ending offset together with the Partial Flag support the master checking the data integrity after a Write command. The highest valued bit of the E/S register, called AA is valid only if the PF flag reads 0. If PF is 0 and AA is 1, a copy has taken place. The AA bit is cleared when the device receives a write scratchpad command.



## WRITING WITH VERIFICATION

To write data to the DS1977, the scratchpad has to be used as intermediate storage. First the master issues the Write Scratchpad command to specify the desired target address, followed by the data to be written to the scratchpad. Under certain conditions (see Write Scratchpad command) the master will receive an inverted CRC16 of the command, address and data at the end of the write scratchpad command sequence. Knowing this CRC value, the master can compare it to the value it has calculated itself to decide whether the communication was successful and proceed to the Copy Scratchpad command. If the master could not receive the CRC16, it has to send the Read Scratchpad command to read back the scratchpad to verify data integrity. As preamble to the scratchpad data, the DS1977 repeats the target address TA1 and TA2 and sends the contents of the E/S register. If the PF flag is set, data did not arrive correctly in the scratchpad or there was a loss of power since data was last written to the scratchpad. The master does not need to continue reading; it can start a new trial to write data to the scratchpad. Similarly, a set AA flag together with a cleared PF flag indicates that the Write command was not recognized by the device. If everything went correctly, both flags are cleared and the ending offset indicates the address of the last byte written to the scratchpad; the master can continue reading and verifying every data byte. After the master has verified the data, it has to send the Copy Scratchpad command. This command must be followed exactly by the data of the three address registers TA1, TA2, and E/S. The master may obtain the contents of these registers by reading the scratchpad or derive it from the target address and the amount of data to be written. As soon as the DS1977 has received these bytes correctly and the master has provided an acceptable password, the DS1977 will copy the scratchpad data to the requested location beginning at the target address.

## MEMORY FUNCTION COMMANDS

The “Memory Function Flow Chart” (Figure 7) describes the protocols necessary for accessing the memory and the special function registers of the DS1977. Examples on how to use these functions to operate the DS1977 are included at the end of this document, preceding the Electrical Characteristics section. The communication between master and DS1977 takes place either at standard speed (default, OD = 0) or at Overdrive Speed (OD = 1). If not explicitly set into the Overdrive mode the DS1977 assumes regular speed.

### Write Scratchpad Command [0Fh]

This command is used to specify the target address and to write data to the scratchpad for verification before the transfer to the EEPROM can be initiated. After issuing the write scratchpad command, the master must first provide the 2-byte target address, followed by the data to be written to the scratchpad. The data will be written to the scratchpad starting at the byte offset (T5:T0). The ending offset (E5: E0) will be the byte offset at which the master stops writing data. Only full data bytes are accepted. If the last data byte is incomplete its content will be ignored and the partial byte flag PF will be set. When writing to a password address, internal circuitry of the chip will force the 3 least significant address bits to 0. Only full 8-byte passwords are accepted. The ending offset will be 07 or 0F, depending on the password(s) to be changed.

When executing the Write Scratchpad command the CRC generator inside the DS1977 (Figure 13) calculates an inverted CRC over the entire data stream, starting at the command code and ending at the last data byte sent by the master. This CRC is generated using the CRC16 polynomial by first clearing the CRC generator and then shifting in the command code (0FH) of the Write Scratchpad command, the Target Addresses TA1 and TA2 as supplied by the master and all the data bytes. The master may end the Write Scratchpad command at any time. However, if the ending offset is 3Fh, the master may send 16 read-time slots and will receive the CRC generated by the DS1977.

The memory address range of the DS1977 is 0000h to 7FFFh (Figure 5). There is no user-access to the address range 7FD1h to 7FFFh. If the master sends a target address higher than this, the internal circuitry of the chip will set the most significant address bit to zero as it is shifted into the internal address register. The Read Scratchpad command will reveal the target address as it will be used by the DS1977. The master will identify such address modifications by comparing the target address read back to the target address transmitted. If the master does not read the scratchpad, a subsequent copy scratchpad command will not work since the most significant bits of the target address the master sends will not match the value the DS1977 expects.

### Read Scratchpad Command [AAh]

This command is used to verify scratchpad data and target address. After issuing the Read Scratchpad command, the master begins reading. The first 2 bytes will be the target address. The next byte will be the ending offset/data status byte (E/S) followed by the scratchpad data beginning at the byte offset (T5:T0), as shown in Figure 6.

Regardless of the actual ending offset the master may continue reading data until the end of the scratchpad after which it will receive an inverted CRC16 of the command code, Target Addresses TA1 and TA2, the E/S byte, and the scratchpad data starting at the byte offset, which is determined by the target address. After the CRC is read, the bus master will read logical 1s from the DS1977 until a reset pulse is issued.

### **Copy Scratchpad with Password [99h]**

This command is used to transfer data from the scratchpad to the memory. After issuing the copy scratchpad command, the master must provide a 3-byte authorization pattern, which can be obtained by reading the scratchpad for verification. This pattern must exactly match the data contained in the three address registers (TA1, TA2, E/S, in that order). Next the master must send a valid full-access password, or, if passwords are not enabled, 8 dummy bytes. Now the master must provide power by bypassing the 1-Wire pullup resistor with an electronic switch, generating a "strong pullup". If authorization pattern and password are accepted, the AA (Authorization Accepted) flag will be set and the copy will begin. Copy takes 10ms maximum during which the voltage on the 1-Wire bus must not fall below 2.8V. After the copy is completed, the master turns off the strong pullup and begins reading from the 1-Wire. A pattern of alternating 1's and 0's will indicate that the copy command was executed successfully. If the copy command was disturbed due to lack of power or for other reasons (see Figure 7-2, "strong pullup valid?"), the master will read a constant stream of FFh bytes until it sends a 1-Wire reset pulse. In this case the destination memory may be incompletely programmed requiring a write scratchpad and copy scratchpad be repeated to ensure proper programming of the EEPROM. This requires careful consideration when designing application software that writes to the DS1977 in an intermittent contact environment.

The data to be copied is determined by the three address registers (TA1, TA2, E/S). The scratchpad data from the beginning offset through the ending offset will be copied to memory, starting at the target address. Anywhere from 1 to 64 bytes may be copied to memory with this command.

### **Read Memory with Password [69h]**

This command is used to read the entire memory, except for the passwords. After issuing the command, the master must provide the 2-byte target address. Next the master must send a valid read access password, or, if passwords are not enabled, 8 dummy bytes. Now the master must provide power by bypassing the 1-Wire pullup resistor with an electronic switch, generating a "strong pullup". If the password was accepted, EEPROM data beginning at the specified target address and ending at the page boundary will be loaded into the scratchpad starting at the beginning offset. This transfer takes 5 ms maximum during which the voltage on the 1-Wire bus must not fall below 2.8V. After the transfer is completed, the master turns off the strong pullup and begins reading from the 1-Wire. When the end of the memory page (end of scratchpad) is reached, the master will receive an inverted CRC16 of the command, target address and page data. If the master wants to read more data and the end of the memory is not yet reached, it again has to activate the strong pullup. This will transfer a full 64-byte page of memory data to the scratchpad from where the master can read it by issuing read-time slots. This transfer only takes place if the DS1977 receives enough power through the 1-Wire line (see Figure 7-3, "strong pullup valid?"). The loop of strong pullup and reading 64 bytes can be repeated until the end of the memory is reached, at which point the master will read logic 1's.

### **Verify Password [C3h]**

This command allows the user to verify whether the process of updating a password was successful, eliminating the risk of a weak programming of the memory cells that actually store the password. The command allows verifying one password at a time. After issuing the command code, the master must send the memory address of the password to be verified. Next the master transmits the password itself and generates a strong pullup to provide the power for the password comparison. This takes 5ms maximum, during which the voltage on the 1-Wire bus must not fall below 2.8V. After the comparison is completed, the master turns off the strong pullup and begins reading from the 1-Wire line. A pattern of alternating 1's and 0's indicates that the verification was successful, i. e., the password supplied by the master matches the one stored in the DS1977. If the passwords do not match, the master will read a constant stream of FFh bytes until it sends a reset pulse.

Before changing a password, first disable the use of passwords. Then using Write Scratchpad, Read Scratchpad and Copy Scratchpad, write the new password to its respective memory location. Now use Verify Password to double-check whether the password reads correctly from the EEPROM memory. If the verification is successful, it is safe to again enable passwords.

**Figure 7-1. MEMORY/CONTROL FUNCTION FLOW CHART**

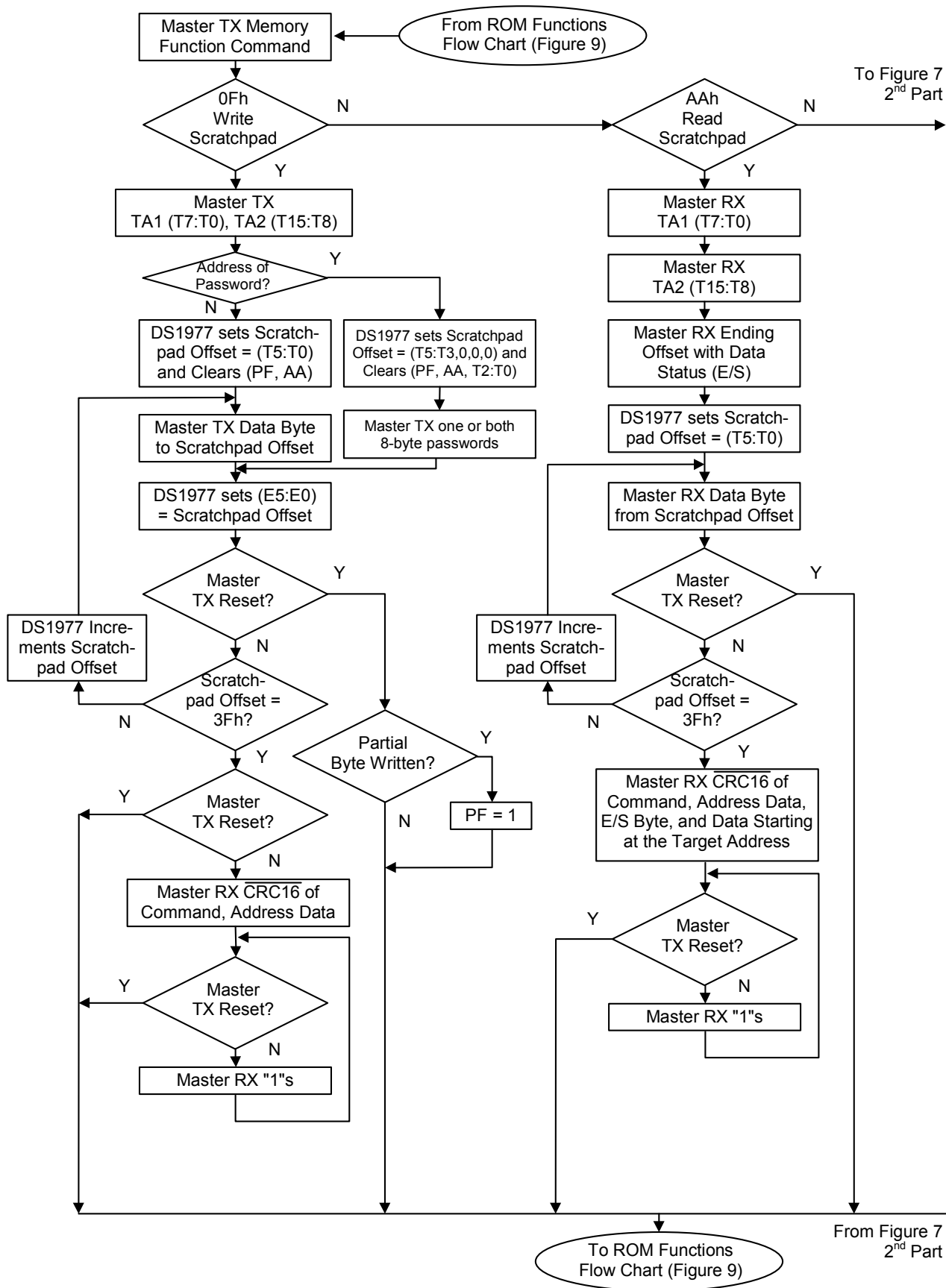
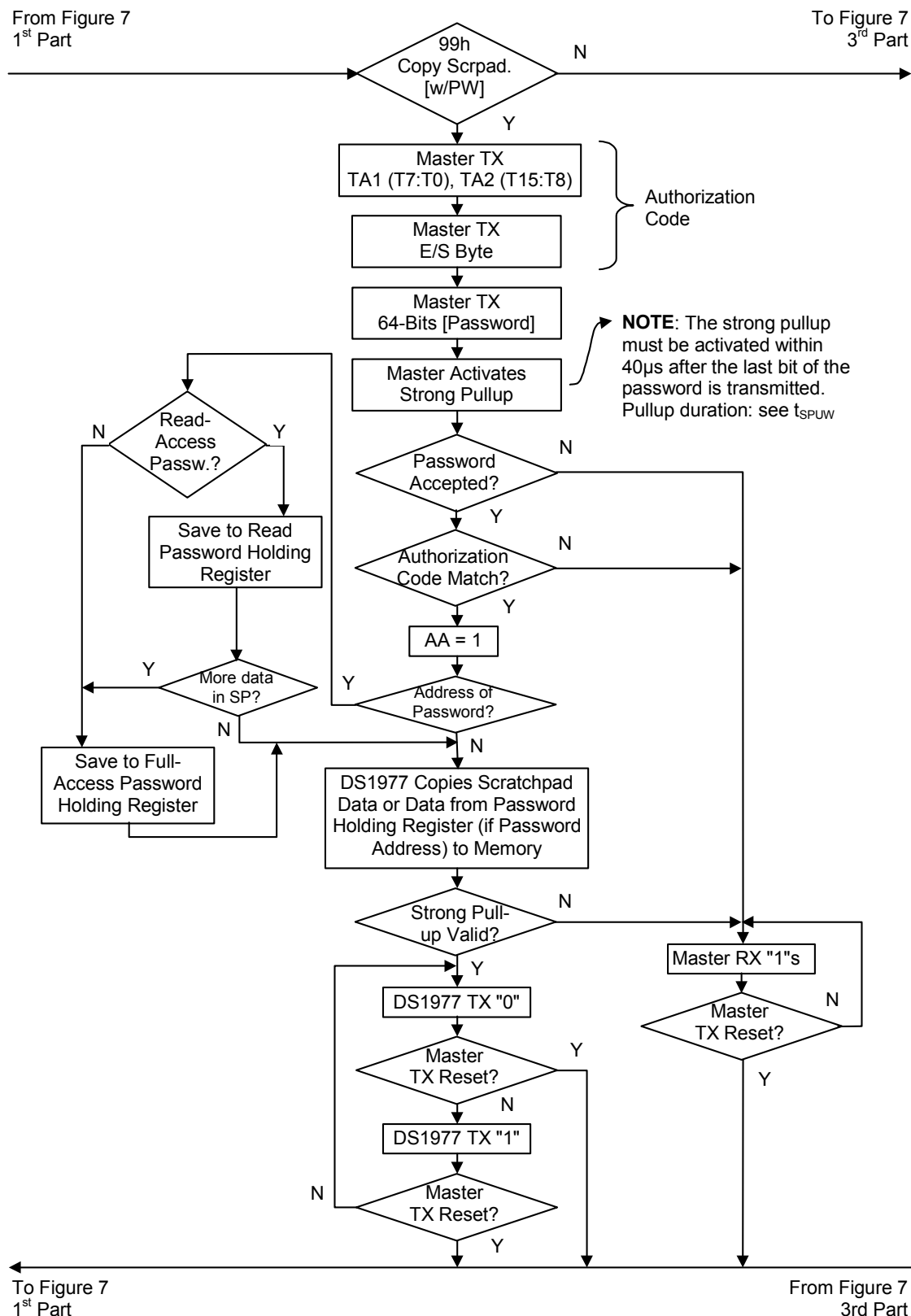
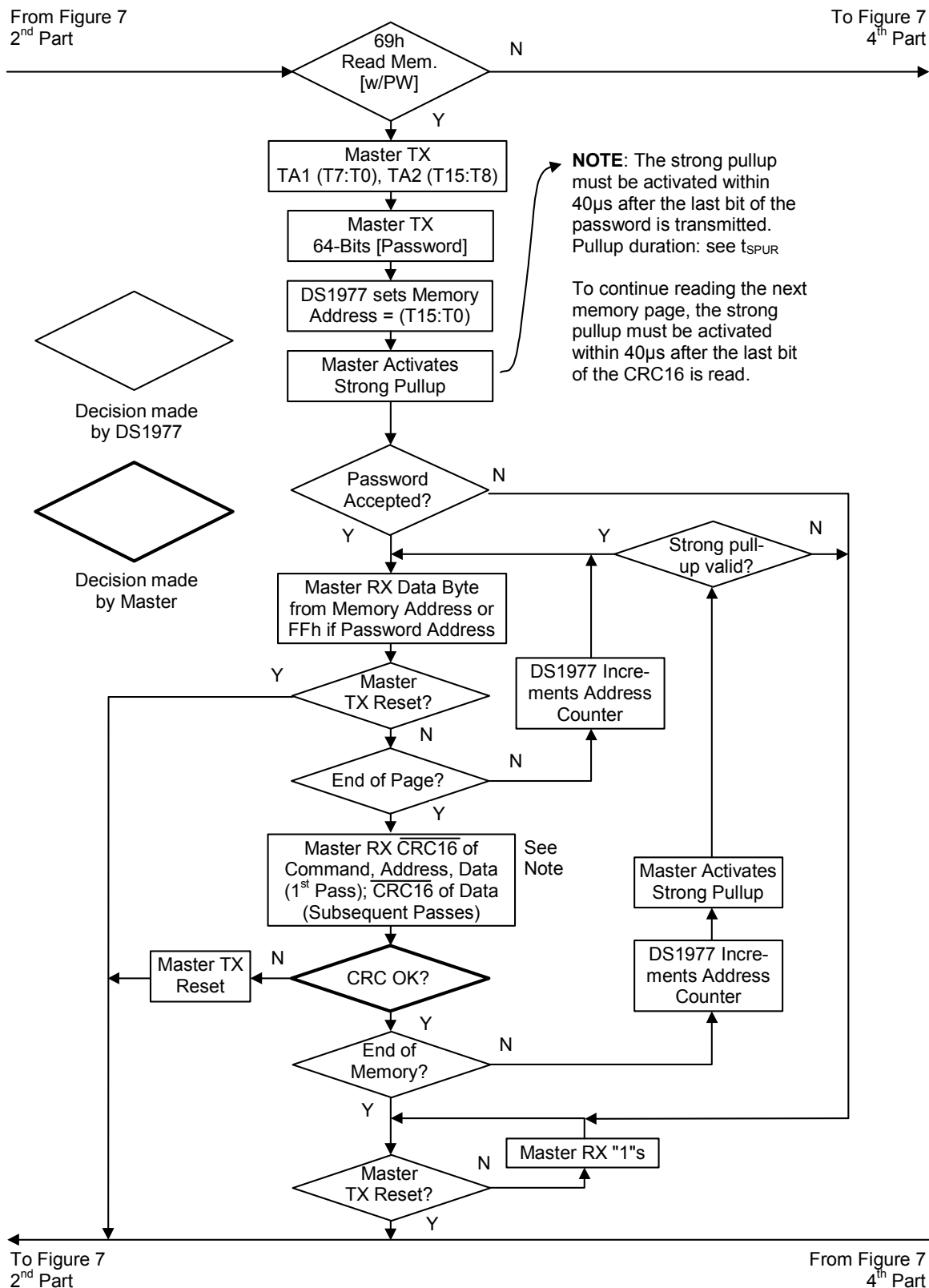


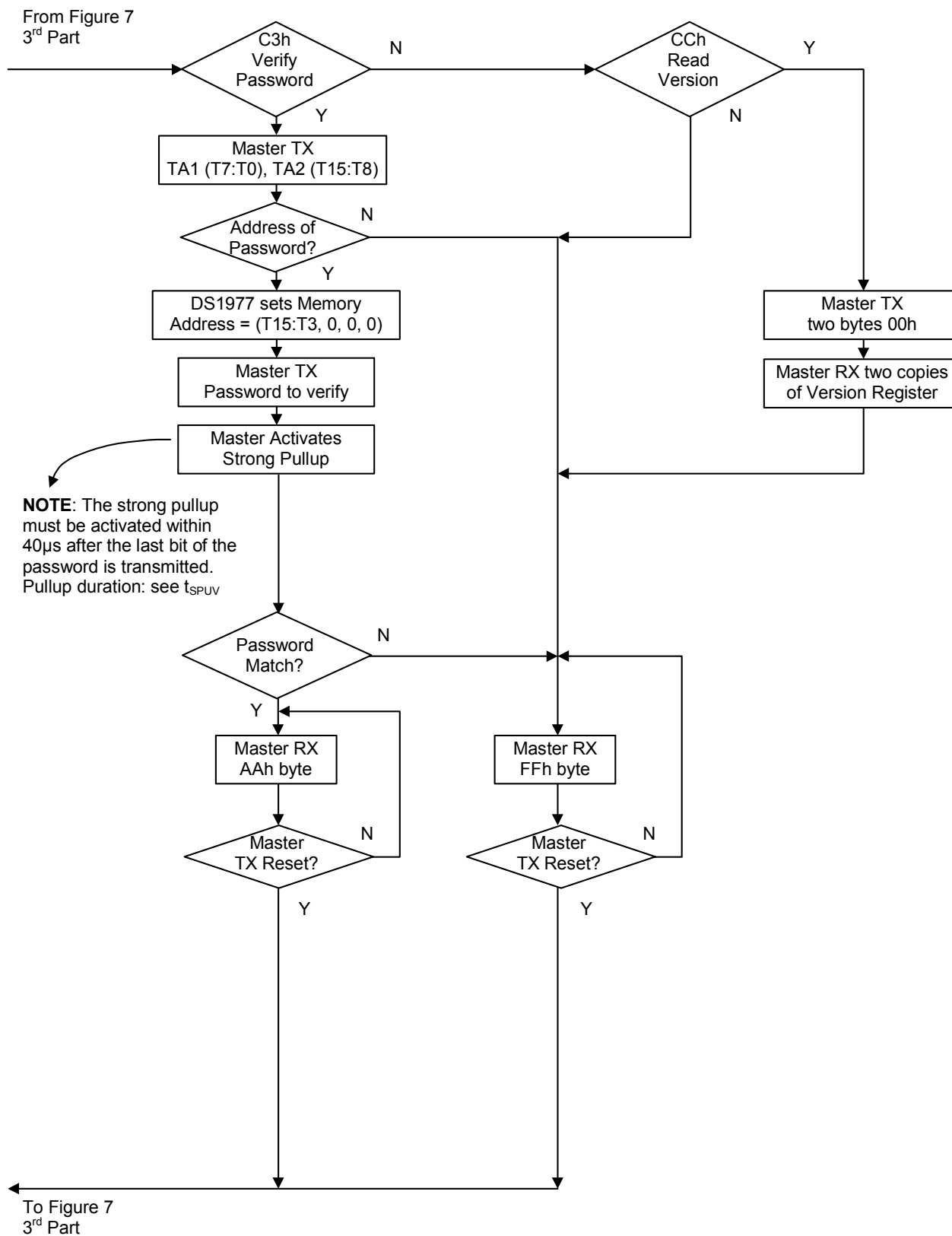
Figure 7-2. MEMORY/CONTROL FUNCTION FLOW CHART



**Figure 7-3. MEMORY/CONTROL FUNCTION FLOW CHART**



**Figure 7-4. MEMORY/CONTROL FUNCTION FLOW CHART**



## Read Version Command [CCh]

This command allows the master to read the chip revision code of the DS1977. After issuing the command code, the master sends two 00h-bytes to access the version register. With the next 16 time slots the master receives two copies of the content of the version register. Additional read-time slots will read logic 1's. Only the upper 3 bits of the version register are valid. The lower 5 bits will all read 0.

## 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 DS1977 is a slave device. The bus master is typically a microcontroller or PC. For small configurations the 1-Wire communication signals can be generated under software control using a single port pin. A second port pin is required to control the strong pullup to supply power for the commands Copy Scratchpad with Password, Read Memory with Password and Verify Password. Alternatively, the DS2480B 1-Wire line driver chip or serial port adapters based on this chip (DS9097U series) are can be used. This simplifies the hardware design and frees the microprocessor from responding in real-time.

The discussion of this bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing). The 1-Wire protocol defines bus transactions in terms of the bus state during specific 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 open drain or tri-state outputs. The 1-Wire port of the DS1977 is open-drain with an internal circuit equivalent to that shown in Figure 8.

A multi-drop bus consists of a 1-Wire bus with multiple slaves attached. At standard speed the 1-Wire bus has a maximum data rate of 15.3 kbits per second. The speed can be boosted to 125 kbits per second by activating the Overdrive mode. The value of the pullup resistor primarily depends on the network size and load conditions. For most applications the optimal value of the pullup resistor will be approximately 2.2k $\Omega$  for standard speed and 1.5k $\Omega$  for Overdrive speed.

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 $\mu$ s (Overdrive speed) or more than 120 $\mu$ s (standard speed), one or more devices on the bus may be reset.

## TRANSACTION SEQUENCE

The protocol for accessing the DS1977 through the 1-Wire port is as follows:

- Initialization
- ROM Function Command
- Memory Function Command
- Transaction/Data

Illustrations of the transaction sequence for the various memory function commands are found later in this document.

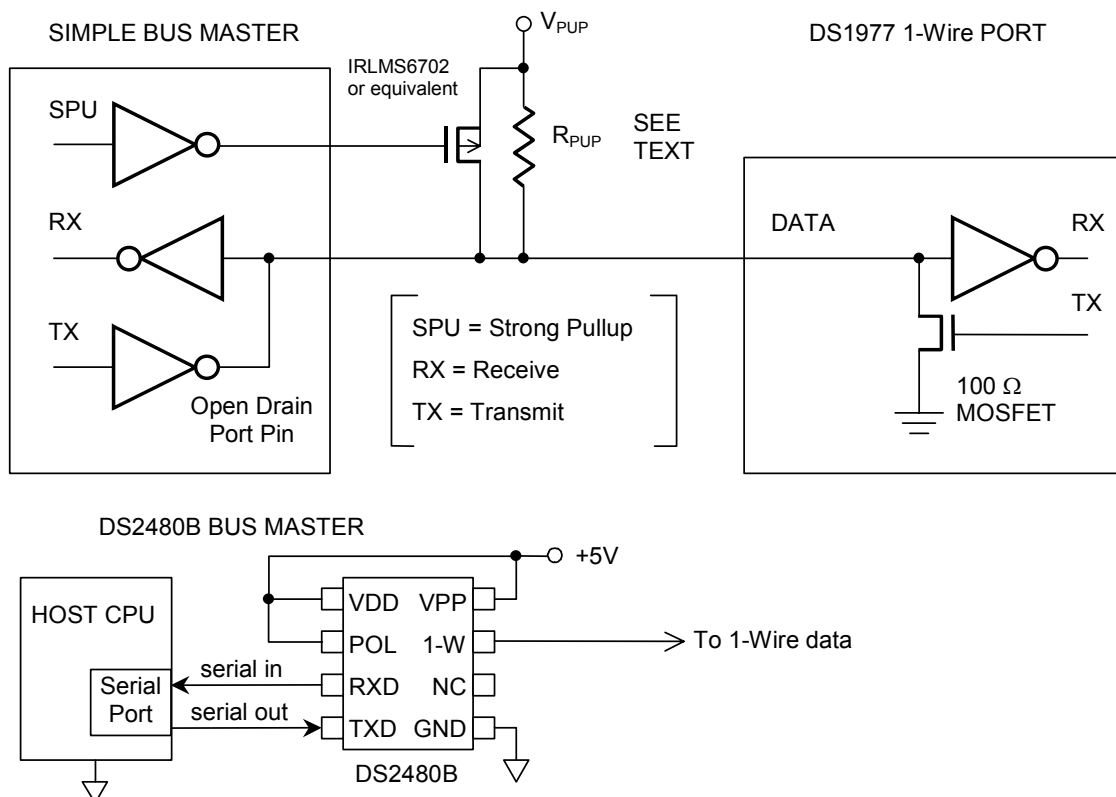
## INITIALIZATION

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 DS1977 is on the bus and is ready to operate. For more details, see the 1-Wire Signaling section.

## 1-Wire ROM FUNCTION COMMANDS

Once the bus master has detected a presence, it can issue one of the eight ROM function commands. All ROM function commands are 8 bits long. A list of these commands follows (refer to flowchart in Figure 9).

## Figure 8. HARDWARE CONFIGURATION



### READ ROM [33H]

This command allows the bus master to read the DS1977'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 DS1977 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 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 DS1977 on a multidrop bus. Only the DS1977 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.

### 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 search ROM process is the repetition of a simple three-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, three-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 Application Note 187 for a comprehensive discussion of the 1-Wire search algorithm.

### 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).



**Figure 9-1. ROM FUNCTIONS FLOW CHART**

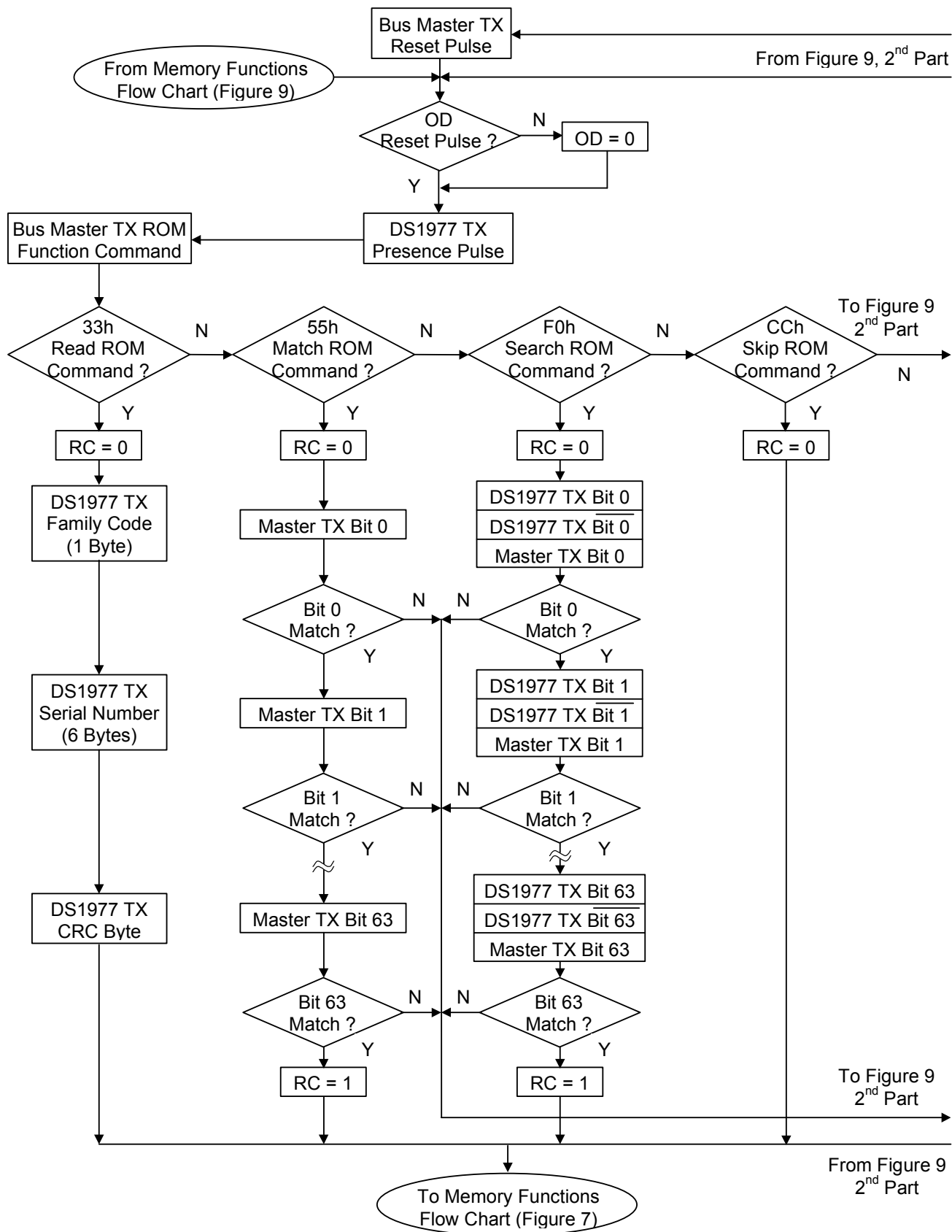
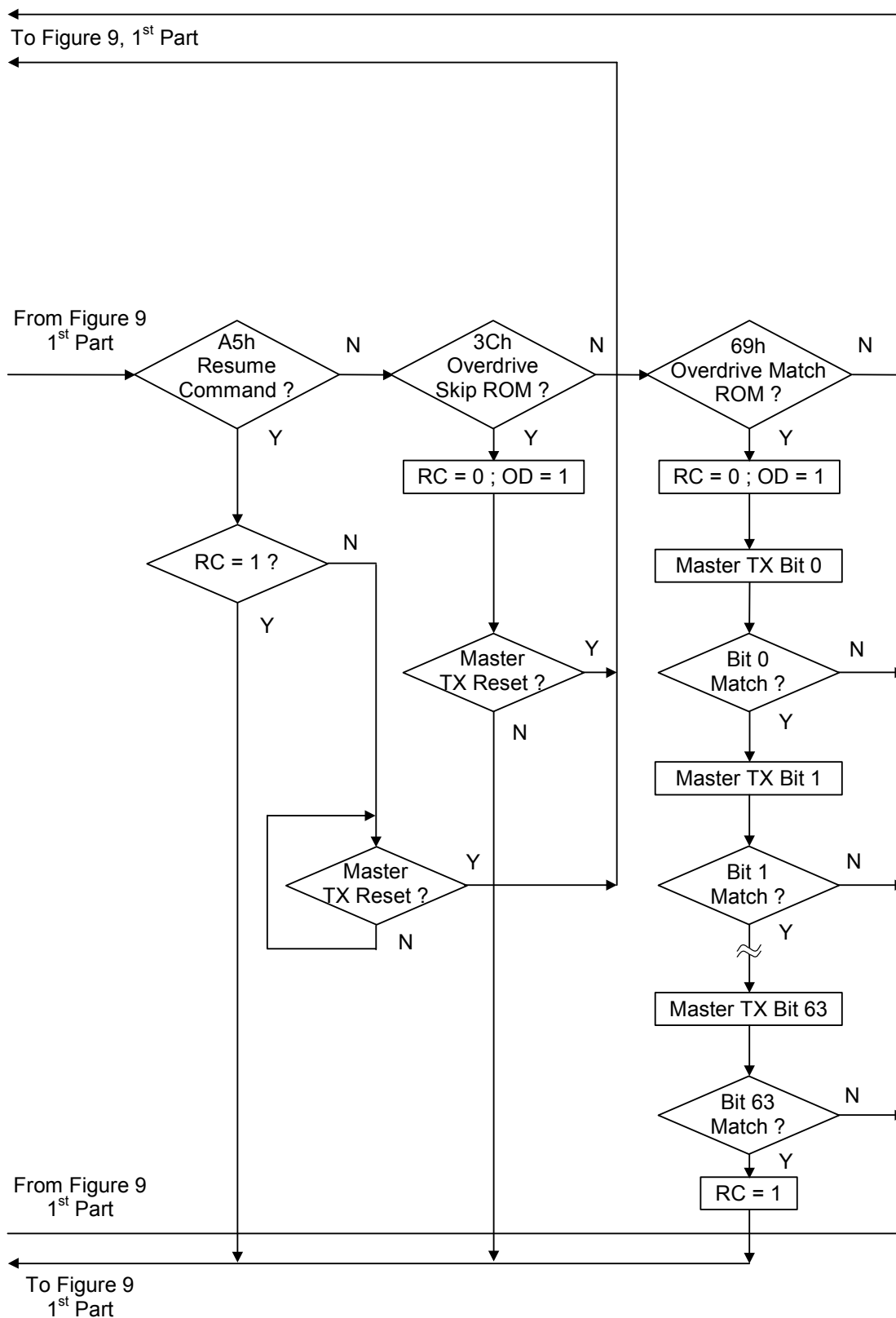


Figure 9-2. ROM FUNCTIONS FLOW CHART



## RESUME COMMAND [A5h]

The Resume Command function maximizes the data throughput in a multidrop environment. This function checks the status of the RC bit and, if it is set, directly transfers control to the Memory/Control functions, similar to a Skip ROM command. The only way to set the RC bit is through successfully executing the Match ROM, Search ROM or Overdrive Match ROM command. Once the RC bit is set, the device can repeatedly be accessed through the Resume Command function. Accessing another device on the bus will clear the RC bit, preventing two or more devices from simultaneously responding to the Resume Command function.

## 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 DS1977 in the Overdrive mode ( $OD = 1$ ). All communication following this command has to occur at Overdrive speed until a reset pulse of minimum 480 $\mu$ s duration resets all devices on the bus to standard speed ( $OD = 0$ ).

When issued on a multidrop bus this command will set all Overdrive-supporting devices into Overdrive mode. To subsequently address a specific Overdrive-supporting device, a reset pulse at Overdrive speed has to be issued followed by a Match ROM or Search ROM command sequence. This will speed up 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 pulldowns will produce a wired-AND result).

## OVERDRIVE MATCH ROM [69H]

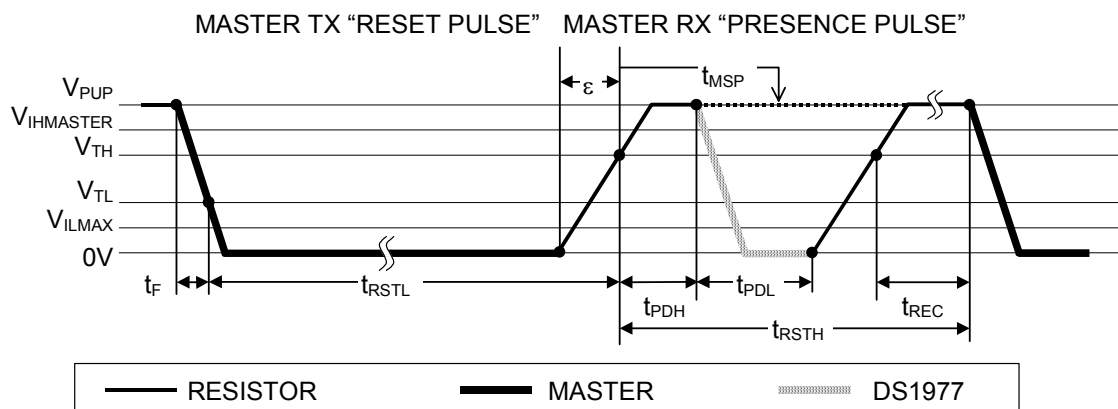
The Overdrive Match ROM command followed by a 64-bit ROM sequence transmitted at Overdrive Speed allows the bus master to address a specific DS1977 on a multidrop bus and to simultaneously set it in Overdrive mode. Only the DS1977 that exactly matches the 64-bit ROM sequence will respond to the subsequent memory function command. Slaves already in Overdrive mode from a previous Overdrive Skip or Match command will remain in Overdrive mode. All overdrive-capable slaves will return to standard speed at the next Reset Pulse of minimum 480 $\mu$ s duration. The Overdrive Match ROM command can be used with a single or multiple devices on the bus.

## 1-Wire SIGNALING

The DS1977 requires strict protocols to ensure data integrity. The protocol consists of five types of signaling on one line: Reset Sequence with Reset Pulse and Presence Pulse, Write-Zero, Write-One Read-Data, and strong pullup to supply power over the 1-Wire line. Except for the presence pulse the bus master initiates all these signals. The DS1977 can communicate at two different speeds, standard speed and Overdrive Speed. If not explicitly set into the Overdrive mode, the DS1977 will communicate at standard speed. While in Overdrive mode the fast timing applies to all waveforms.

To get from idle to active, the voltage on the 1-Wire line needs to fall from  $V_{PUP}$  below the threshold  $V_{TL}$ . To get from active to idle, the voltage needs to rise from  $V_{ILMAX}$  past the threshold  $V_{TH}$ . The time it takes for the voltage to make this rise, in Figure 10 as  $\epsilon$ , and its duration depends on the pullup resistor (RPUP) used and capacitance of the 1-Wire network attached. The voltage  $V_{ILMAX}$  is relevant for the DS1977 when determining a logical level, not triggering any events.

The initialization sequence required to begin any communication with the DS1977 is shown in Figure 10. A Reset Pulse followed by a Presence Pulse indicates the DS1977 is ready to receive data, given the correct ROM and memory function command. If the bus master uses slew-rate control on the falling edge, it must pull down the line for  $t_{RSTL} + t_F$  to compensate for the edge. A  $t_{RSTL}$  duration of 480 $\mu$ s or longer will exit the Overdrive mode returning the device to standard speed. If the DS1977 is in Overdrive Mode and  $t_{RSTL}$  is no longer than 80 $\mu$ s the device will remain in Overdrive mode.

**Figure 10. INITIALIZATION PROCEDURE “RESET AND PRESENCE PULSES”**

After the bus master has released the line it goes into receive mode (RX). Now the 1-Wire bus is pulled to  $V_{PUP}$  via the pullup resistor or, in case of a DS2480B driver, by active circuitry. When the threshold  $V_{TH}$  is crossed, the DS1977 waits for  $t_{PDH}$  and then transmits a Presence Pulse by pulling the line low for  $t_{PDL}$ . To detect a presence pulse, the master must test the logical state of the 1-Wire line at  $t_{MSP}$ .

The  $t_{RSTH}$  window must be at least the sum of  $t_{PDHMAX}$ ,  $t_{PDLMAX}$ , and  $t_{RECMIN}$ . Immediately after  $t_{RSTH}$  is expired, the DS1977 is ready for data communication. In a mixed population network  $t_{RSTH}$  should be extended to minimum 480 $\mu$ s at standard speed and 48 $\mu$ s at Overdrive speed to accommodate other 1-Wire devices.

## READ/WRITE-TIME SLOTS

Data communication with the DS1977 takes place in time slots, which carry a single bit each. Write-time slots transport data from bus master to slave. Read-time slots transfer data from slave to master. The definitions of the write and read-time slots are illustrated in Figure 11.

All communication begins with the master pulling the data line low. As the voltage on the 1-Wire line falls below the threshold  $V_{TL}$ , the DS1977 starts its internal timing generator that determines when the data line will be sampled during a write-time slot and how long data will be valid during a read-time slot.

## MASTER-TO-SLAVE

For a **write-one** time slot, the voltage on the data line must have crossed the  $V_{THMAX}$  threshold after the write-one low time  $t_{W1LMAX}$  is expired. For a **write-zero** time slot, the voltage on the data line must stay below the  $V_{THMIN}$  threshold until the write-zero low time  $t_{W0LMIN}$  is expired. For most reliable communication the voltage on the data line should not exceed  $V_{ILMAX}$  during the entire  $t_{WOL}$  window. After the  $V_{THMAX}$  threshold has been crossed, the DS1977 needs a recovery time  $t_{REC}$  before it is ready for the next time slot.

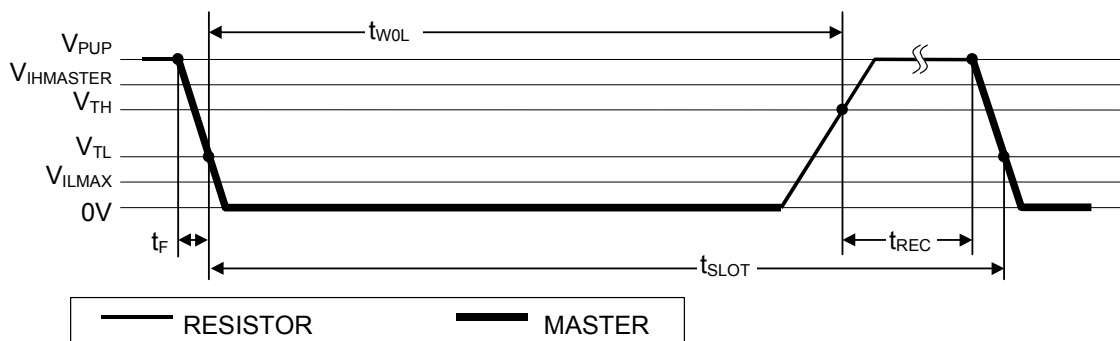
**Figure 11. READ/WRITE TIMING DIAGRAM**

### Write-One Time Slot

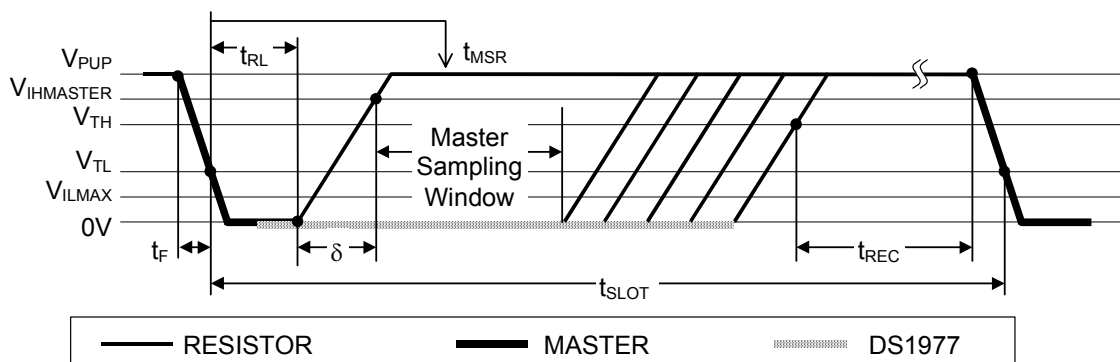


Figure 11 (continued)

## Write-Zero Time Slot



## Read-Data Time Slot



## SLAVE-TO-MASTER

A **read-data** time slot begins like a write-one time slot. The voltage on the data line must remain below  $V_{TLMIN}$  until the read low time  $t_{RL}$  is expired. During the  $t_{RL}$  window, when responding with a 0, the DS1977 will start pulling the data line low; its internal timing generator determines when this pulldown ends and the voltage starts rising again. When responding with a 1, the DS1977 will not hold the data line low at all, and the voltage starts rising as soon as  $t_{RL}$  is over.

The sum of  $t_{RL} + \delta$  (rise time) on one side and the internal timing generator of the DS1977 on the other side define the master sampling window ( $t_{MSRMIN}$  to  $t_{MSRMAX}$ ) in which the master must perform a read from the data line. For most reliable communication,  $t_{RL}$  should be as short as permissible and the master should read close to but no later than  $t_{MSRMAX}$ . After reading from the data line, the master must wait until  $t_{SLOT}$  is expired. This guarantees sufficient recovery time  $t_{REC}$  for the DS1977 to get ready for the next time slot.

## IMPROVED NETWORK BEHAVIOR

1-Wire networks can only be terminated during transients controlled by the bus master (1-Wire driver) and are therefore susceptible to noise of various origins. Depending on the physical size and topology of the network, reflections from end points and branch points can add up or cancel each other to some extent. Such reflections are visible as glitches or ringing on the 1-Wire communication line. A glitch during the rising edge of a time slot can cause a slave device to lose synchronization with the master and, as a consequence, result in a search ROM command coming to a dead end. For better performance in network applications, the DS1977 uses a new 1-Wire front end, which makes it less sensitive to noise and also reduces the magnitude of noise injected by the slave device itself.

The 1-Wire front end of the DS1977 differs from traditional slave devices in four characteristics.

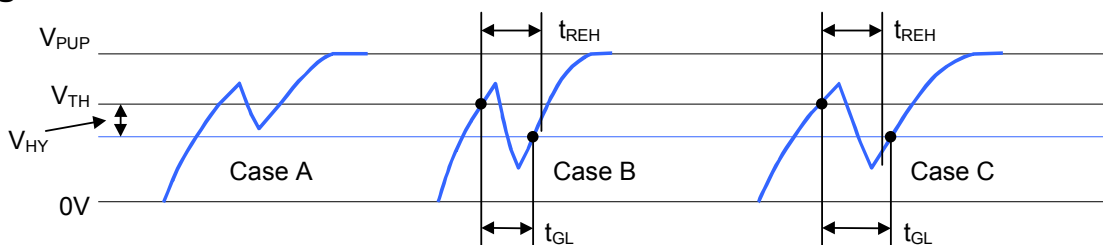
- 1) The falling edge of the presence pulse has a controlled slew rate. This provides a better match to the line impedance than a digitally switched transistor, converting the high frequency ringing known from traditional

devices into a smoother low-bandwidth transition. The slew rate control is specified by the parameter  $t_{FPD}$ , which has different values for standard and Overdrive speed.

- 2) There is additional low-pass filtering in the circuit that detects the falling edge at the beginning of a time slot. This reduces the sensitivity to high-frequency noise. This additional filtering does not apply at Overdrive speed.
- 3) There is a hysteresis at the low-to-high switching threshold  $V_{TH}$ . If a negative glitch crosses  $V_{TH}$  but doesn't go below  $V_{TH} - V_{HY}$ , it will not be recognized (Figure 12, Case A). The hysteresis is effective at any 1-Wire speed.
- 4) There is a time window specified by the rising edge hold-off time  $t_{REH}$  during which glitches will be ignored, even if they extend below  $V_{TH} - V_{HY}$  threshold (Figure 12, Case B,  $t_{GL} < t_{REH}$ ). Deep voltage droops or glitches that appear late after crossing the  $V_{TH}$  threshold and extend beyond the  $t_{REH}$  window cannot be filtered out and will be taken as beginning of a new time slot (Figure 12, Case C,  $t_{GL} \geq t_{REH}$ ).

Only devices which have the parameters  $t_{FPD}$ ,  $V_{HY}$  and  $t_{REH}$  specified in their electrical characteristics use the improved 1-Wire front end.

## Figure 12. NOISE SUPPRESSION SCHEME



## CRC GENERATION

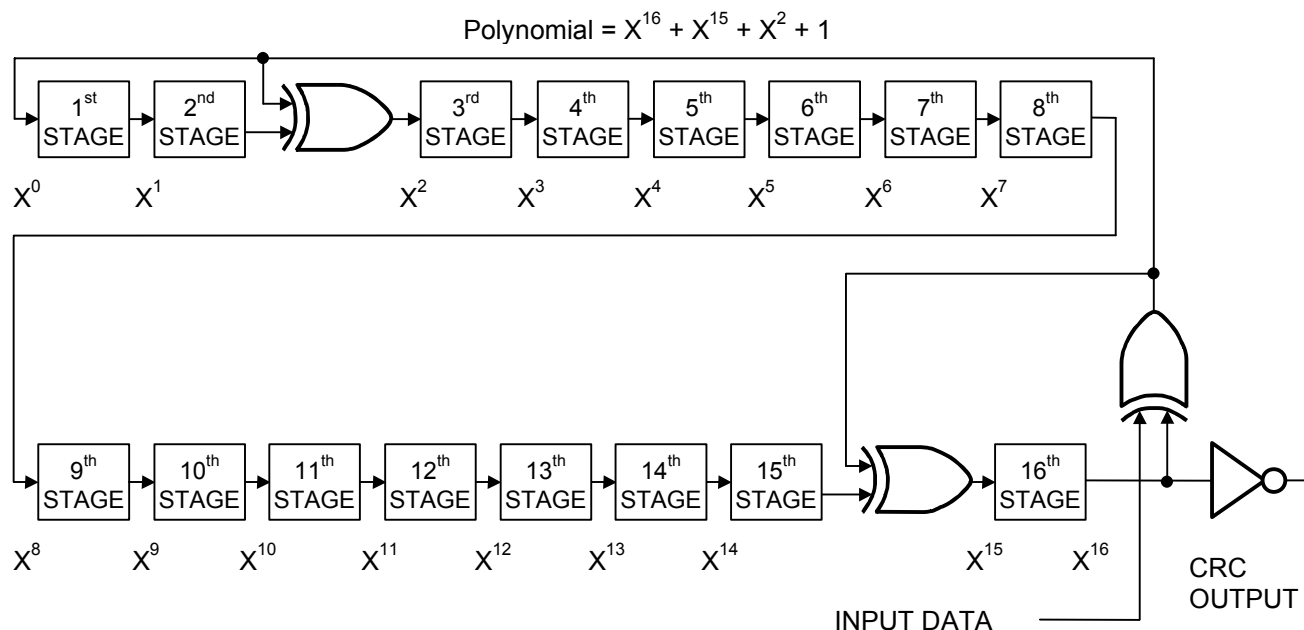
With the DS1977 there are two different types of CRCs (Cyclic Redundancy Checks). One CRC is an 8-bit type and is stored in the most significant byte of the 64-bit ROM. The bus master can compute a CRC value from the first 56 bits of the 64-bit ROM and compare it to the value stored within the DS1977 to determine if the ROM data has been received error-free. 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. 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 the memory using the Read Memory with Password command and for fast verification of a data transfer when writing to or reading from the scratchpad. In contrast to the 8-bit CRC, the 16-bit CRC is always communicated in the inverted form. A CRC-generator inside the DS1977 chip (Figure 13) will calculate a new 16-bit CRC as shown in the command flow chart of Figure 9. 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 reread the portion of the data with the CRC error. With the initial pass through the Read Memory with Password flow chart, the 16-bit CRC value is the result of shifting the command byte into the cleared CRC generator, followed by the 2 address bytes and the data bytes. The password is excluded from the CRC calculation. Subsequent passes through the Read Memory with Password flow chart will generate a 16-bit CRC that is the result of clearing the CRC generator and then shifting in the data bytes.

With the Write Scratchpad command the CRC is generated by first clearing the CRC generator and then shifting in the command code, the Target Addresses TA1 and TA2 and all the data bytes. The DS1977 will transmit this CRC only if the data bytes written to the scratchpad include scratchpad ending offset 3Fh. The data may start at any location within the scratchpad.

With the Read Scratchpad command the CRC is generated by first clearing the CRC generator and then shifting in the command code, the Target Addresses TA1 and TA2, the E/S byte, and the scratchpad data starting at the target address. The DS1977 will transmit this CRC only if the reading continues through the end of the scratchpad, regardless of the actual ending offset.

For more information on generating CRC values see Application Note 27.

**Figure 13. CRC16 HARDWARE DESCRIPTION AND POLYNOMIAL****COMMAND-SPECIFIC 1-Wire COMMUNICATION PROTOCOL—LEGEND**

SYMBOL	DESCRIPTION
RST	1-Wire Reset Pulse Generated by Master
PD	1-Wire Presence Pulse Generated by Slave
Select	Command and Data to Satisfy the ROM Function Protocol
WS	Command "Write Scratchpad"
RS	Command "Read Scratchpad"
CPS	Command "Copy Scratchpad with Password"
RM	Command "Read Memory with Password"
VP	Command "Verify Password"
RV	Command "Read Version"
TA	Target Address TA1, TA2
TA-E/S	Target Address TA1, TA2 with E/S Byte
<data to EOS>	Transfer of as Many Data Bytes as are Needed to Reach the Scratchpad Offset 3Fh
<data to EOP>	Transfer of as Many Data Bytes as are Needed to Reach the End of a Memory Page
<PW/dummy>	Transfer of 8 Bytes that Either Represent a Valid Password or Acceptable Dummy Data
<64 bytes>	Transfer of 64 Bytes
<data>	Transfer of an Undetermined Amount of Data
00h	Transmission of One Byte 00h
Password	Transmission of Password
Version	Transmission of Device Version Number
CRC16\	Transfer of an Inverted CRC16
FF loop	Indefinite Loop Where the Master Reads FF Bytes
AA loop	Indefinite Loop Where the Master Reads AA Bytes
Strong Pullup	Data Transfer to/from EEPROM (Data or Passwords Memory); No Activity on the 1-Wire Bus Permitted During this Time

## COMMAND-SPECIFIC 1-WIRE COMMUNICATION PROTOCOL—COLOR CODES

Master to slave	Slave to master	Strong Pullup
-----------------	-----------------	---------------

### WRITE SCRATCHPAD, REACHING THE END OF THE SCRATCHPAD (CANNOT FAIL)

RST	PD	Select	WS	TA	<data to EOS>	CRC16\	FF loop
-----	----	--------	----	----	---------------	--------	---------

### WRITE SCRATCHPAD, NOT REACHING THE END OF THE SCRATCHPAD (CANNOT FAIL)

RST	PD	Select	WS	TA	<data>	RST	PD
-----	----	--------	----	----	--------	-----	----

### READ SCRATCHPAD (CANNOT FAIL)

RST	PD	Select	RS	TA-E/S	<data to EOS>	CRC16\	FF loop
-----	----	--------	----	--------	---------------	--------	---------

### COPY SCRATCHPAD WITH PASSWORD (SUCCESS)

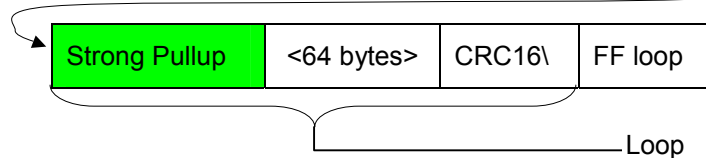
RST	PD	Select	CPS	TA-E/S	<PW/dummy>	Strong Pullup	AA loop
-----	----	--------	-----	--------	------------	---------------	---------

### COPY SCRATCHPAD WITH PASSWORD (FAIL TA-E/S OR PASSWORD)

RST	PD	Select	CPS	TA-E/S	<PW/dummy>	Strong Pullup	FF loop
-----	----	--------	-----	--------	------------	---------------	---------

### READ MEMORY WITH PASSWORD (SUCCESS)

RST	PD	Select	RM	TA	<PW/dummy>	Strong Pullup	<data to EOP>	CRC16\
-----	----	--------	----	----	------------	---------------	---------------	--------



### READ MEMORY WITH PASSWORD (FAIL PASSWORD)

RST	PD	Select	RM	TA	<PW/dummy>	Strong Pullup	FF loop
-----	----	--------	----	----	------------	---------------	---------



**VERIFY PASSWORD (SUCCESS)**

RST	PD	Select	VP	TA	Password	Strong Pullup	AA loop
-----	----	--------	----	----	----------	---------------	---------

**VERIFY PASSWORD (FAIL ADDRESS OR PASSWORD)**

RST	PD	Select	VP	TA	Password	Strong Pullup	FF loop
-----	----	--------	----	----	----------	---------------	---------

**READ VERSION (CANNOT FAIL)**

RST	PD	Select	RV	00h	00h	Version	Version	FF loop
-----	----	--------	----	-----	-----	---------	---------	---------

**COMMUNICATION EXAMPLES**

The examples in this section demonstrate the use of the memory functions in typical situations. The first example shows how to read the ROM and the version register. In the second example, passwords are installed. The third example shows how to write a couple of bytes and how to read adjacent memory pages.

**EXAMPLE 1**

**Task:** Read the ROM and the version register

With only a single DS1977 connected to the bus master, the communication is as follows:

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	(Reset)	Reset Pulse
RX	(Presence)	Presence Pulse
TX	33h	Issue Read ROM Command
RX	<8 Bytes ROM ID>	Read ROM ID
TX	CCh	Issue Read Version Register Command
TX	00h, 00h	Write Two 00h Bytes
RX	<Version>, <Version>	Read Chip Version Code Twice
RX	FFh	Additional Reads Result in FFh Bytes
TX	(Reset)	Reset Pulse
RX	(Presence)	Presence pulse

**EXAMPLE 2**

**Task:** Install and activate passwords; passwords are currently not activated

This task is broken into the following steps:

1. Write new passwords to scratchpad
2. Read Scratchpad
3. Copy scratchpad
4. Verify new passwords
5. Activate password

With only a single DS1977 connected to the bus master, the communication is as follows:

MASTER MODE		DATA (LSB FIRST)	COMMENTS
Step 1	TX	(Reset)	Reset Pulse
	RX	(Presence)	Presence Pulse
	TX	CCh	Issue Skip ROM Command
	TX	0Fh	Issue Write Scratchpad Command
	TX	C0h	TA1, Target Address = C0h (Password Start Address)
	TX	7Fh	TA2, Target Address = <u>7FC0h</u>
	TX	<Read Password>	Write 8-Byte Read Password to Scratchpad
	TX	<Full-Access Password>	Write 8-Byte Full-Access Password to Scratchpad
	TX	(Reset)	Reset Pulse
	RX	(Presence)	Presence Pulse
Step 2	TX	CCh	Issue Skip ROM Command
	TX	AAh	Issue Read Scratchpad Command
	RX	C0h	Read TA1, Target Address = C0h
	RX	7Fh	Read TA2, Target Address = <u>7FC0h</u>
	RX	0Fh	Read E/S-Byte
	RX	<16 Bytes>	Read Both Passwords from Scratchpad and Compare to what was Written
	TX	(Reset)	Reset Pulse
	RX	(Presence)	Presence Pulse
Step 3	TX	CCh	Issue Skip ROM Command
	TX	99h	Issue Copy Scratchpad with Password Command
	TX	C0h	TA1, Target Address = C0h
	TX	7Fh	TA2, Target Address = <u>7FC0h</u>
	TX	0Fh	E/S-byte
	TX	<8 Bytes>	Transmit 8 Dummy Bytes as Password, Because Passwords are Not Yet Enabled
	(—)	(Activate Strong Pullup for $t_{\text{PROG}}$ )	Supply Power for Programming
	RX	AAh	Read to Check for Programming Success; AAh Means Success
	TX	(Reset)	Reset Pulse
	RX	(Presence)	Presence Pulse
	Step 4	TX	CCh
TX		C3h	Issue Verify Password Command
TX		C0h	TA1, Target Address = C0h (Read Password Address)
TX		7Fh	TA2, target address = <u>7FC0h</u>
TX		<Read Password>	Transmit Read Password
(—)		(Activate Strong Pullup for $t_{\text{PROG}}$ )	Supply Power for Password Comparison
RX		AAh	Check for Password Match; AAh = Match
TX		(Reset)	Reset Pulse
RX		(Presence)	Presence Pulse
TX		CCh	Issue Skip ROM Command
TX		C3h	Issue Verify Password Command
TX		C8h	TA1, Target Address = C8h (Full-Access

MASTER MODE	DATA (LSB FIRST)	COMMENTS
		Password Address)
TX	7Fh	TA2, Target Address = <u>7FC8h</u>
TX	<Full-Access Password>	Transmit Full-Access Password
(—)	(Activate Strong Pullup for $t_{\text{PROG}}$ )	Supply Power for Password Comparison
RX	AAh	Check for Password Match; AAh = Match
TX	(Reset)	Reset Pulse
RX	(Presence)	Presence Pulse
Step 5 TX	CCh	Issue Skip ROM Command
TX	0Fh	Issue Write Scratchpad Command
TX	D0h	TA1, Target Address = D0h (Password Control Register Address)
TX	7Fh	TA2, Target Address = <u>7FD0h</u>
TX	AAh	Write Password Enabling Pattern
TX	(Reset)	Reset Pulse
RX	(Presence)	Presence Pulse
TX	CCh	Issue Skip ROM Command
TX	AAh	Issue Read Scratchpad Command
RX	D0h	Read TA1, Target Address = D0h
RX	7Fh	Read TA2, Target Address = <u>7FD0h</u>
RX	10h	Read E/S-Byte
RX	AAh	Verify Password Enabling Pattern
TX	(Reset)	Reset Pulse
RX	(Presence)	Presence Pulse
TX	CCh	Issue Skip ROM Command
TX	99h	Issue Copy Scratchpad with Password Command
TX	C0h	TA1, Target Address = D0h
TX	7Fh	TA2, Target Address = <u>7FD0h</u>
TX	10h	E/S-Byte
TX	<8 Bytes>	Transmit 8 Dummy Bytes as Password, Because Passwords are Not Yet Enabled
(—)	(Activate Strong Pullup for $t_{\text{PROG}}$ )	Supply Power for Programming
RX	AAh	Read to Check for Programming Success; AAh Means Success
TX	(Reset)	Reset Pulse
RX	(Presence)	Presence Pulse

Instead of always using Skip ROM, one could use Read ROM first to learn the device's ROM identification (see Example 1). For the next access one would use the Match ROM command and send the correct ROM identification to address the device. Subsequent accesses could use the Resume command. This procedure ensures that devices cannot be swapped during a communication session.

### EXAMPLE 3

**Task:** write 10 data bytes starting at address 00A0h in page 2; read memory pages 2 and 3. The device has passwords installed and activated. This task is broken into the following steps:

1. Write data to scratchpad
2. Read Scratchpad
3. Copy scratchpad
4. Read the entire memory page 3
5. Continue reading through the end of page 4

With only a single DS1977 connected to the bus master, the communication is as follows:

MASTER MODE		DATA (LSB FIRST)	COMMENTS
Step 1	TX	(Reset)	Reset Pulse
	RX	(Presence)	Presence Pulse
	TX	CCh	Issue Skip ROM Command
	TX	0Fh	Issue Write Scratchpad Command
	TX	A0h	TA1, Target Address = A0h (Start Address)
	TX	00h	TA2, Target Address = 00A0h
	TX	<10 Data Bytes>	Write Data Bytes to Scratchpad
	TX	(Reset)	Reset Pulse
	RX	(Presence)	Presence Pulse
Step 2	TX	CCh	Issue Skip ROM Command
	TX	AAh	Issue Read Scratchpad Command
	RX	A0h	Read TA1, Target Address = A0h
	RX	00h	Read TA2, Target Address = 00A0h
	RX	29h	Read E/S-Byte
	RX	<10 Bytes>	Read from Scratchpad and Compare to what was Written
	TX	(Reset)	Reset Pulse
	RX	(Presence)	Presence Pulse
Step 3	TX	CCh	Issue Skip ROM Command
	TX	99h	Issue Copy Scratchpad with Password Command
	TX	A0h	TA1, Target Address = A0h
	TX	00h	TA2, Target Address = 00A0h
	TX	29h	E/S-Byte
	TX	<Full-Access Password>	Transmit Full-Access Password (8 Bytes)
	(—)	(Activate Strong Pullup for $t_{\text{PROG}}$ )	Supply Power for Programming
	RX	AAh	Read to Check for Programming Success; AAh Means Success
	TX	(Reset)	Reset Pulse
		RX	(Presence)
Step 4	TX	CCh	Issue Skip ROM Command
	TX	69h	Issue Read Memory with Password Command
	TX	80h	TA1, Target Address = 80h
	TX	00h	TA2, Target Address = 0080h
	TX	<Read Password>	Transmit Read Password (8 Bytes)
	(—)	(Activate Strong Pullup for $t_{\text{PROG}}$ )	Supply Power for Reading
	RX	<64 Bytes>	Read Data from Page 2
	RX	<2 Bytes CRC16>	Read Inverted CRC16
Step 5	(—)	(Activate Strong Pullup for $t_{\text{PROG}}$ )	Supply Power for Reading
	RX	<64 Bytes>	Read Data from Page 3
	RX	<2 Bytes CRC16>	Read Inverted CRC16
	TX	(Reset)	Reset Pulse
	RX	(Presence)	Presence Pulse