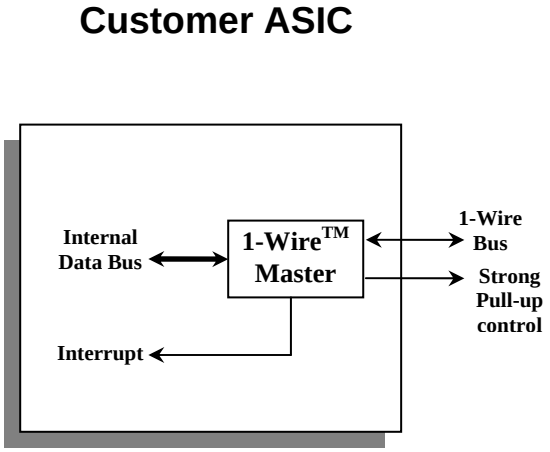
DS1WM
Synthesizable 1-Wire Bus Master

www.maxim-ic.com

FEATURES

- Memory maps into any standard byte-wide data bus.
- Eliminates CPU “bit-banging” by internally generating all 1-Wire timing and control signals.
- Generates interrupts to provide for more efficient programming.
- Search ROM Accelerator relieves CPU from any single bit operations on the 1-Wire® Bus.
- Capable of running off any system clock from 4 MHz to 128 MHz.
- Small size: all digital design, only 3470 gates.
- Applications include any circuit containing a 1-Wire communication bus.
- Supports standard and overdrive 1-Wire communication speeds
- Supports strong pull-up specifications.
- Master available in both Verilog and VHDL
- Supports single bit transmissions.
- Provides added support for long line conditions.



DESCRIPTION

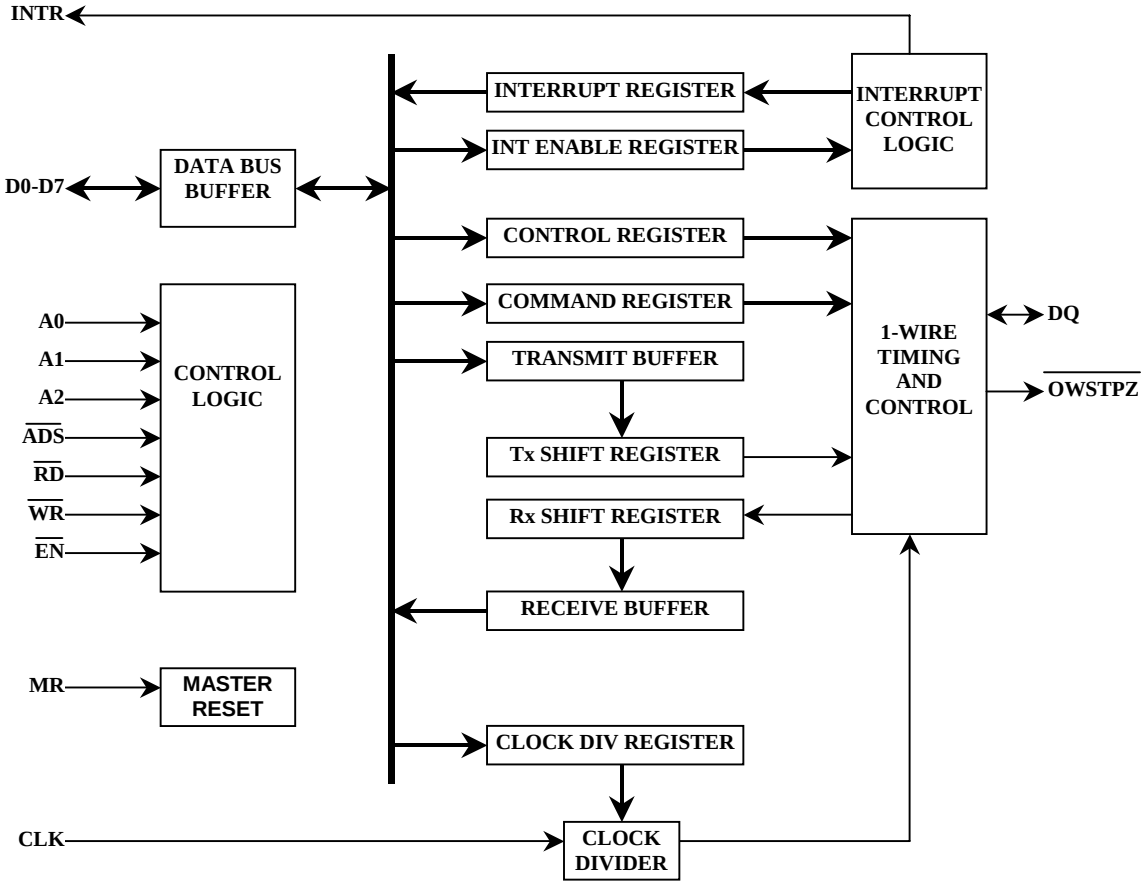
As more 1-Wire devices become available, more and more users have to deal with the demands of generating 1-Wire signals to communicate to them. This usually requires “bit-banging” a port pin on a microprocessor, and having the microprocessor perform the timing functions required for the 1-Wire protocol. While 1-Wire transmission can be interrupted mid-byte, it cannot be interrupted during the “low” time of a bit time slot; this means that a CPU will be idled for up to 60 microseconds for each bit sent and at least 480 microseconds when generating a 1-Wire reset. The 1-Wire Master helps users handle communication to 1-Wire devices in their system without tying up valuable CPU cycles. Integrated into a user’s ASIC as a 1-Wire port, the Verilog or VHDL core uses little chip area (3470 gates plus 2 bond pads).

This circuit is designed to be memory mapped into the user’s system and provides complete control of the 1-Wire bus through 8-bit or single commands. The host CPU loads commands, reads and writes data, and sets interrupt control through six individual registers. All of the timing and control of the 1-Wire bus are generated within. The host merely needs to load a command or data and then may go on about its business. When bus activity has generated a response that the CPU needs to receive, the 1-Wire Master sets a status bit and, if enabled, generates an interrupt to the CPU. In addition to write and read simplification, the 1-Wire Master also provides a Search ROM Accelerator function relieving the CPU from having to perform the complex single-bit operations on the 1-Wire bus.

1-Wire is a registered trademark of Dallas Semiconductor Corp., a wholly owned subsidiary of Maxim Integrated Products, Inc.

The operation of the 1-Wire bus is described in detail in the **Book of iButton® Standards** [1]; therefore, the details of that will not be discussed in this document. Each slave device, in general, has its own set of commands that are described in detail in that device's data sheet. The user is referred to those documents for detail on specific slave implementations.

BLOCK DIAGRAM



PIN DESCRIPTIONS

The following describes the function of all the block I/O pins. In the following descriptions, 0 represents logic low and 1 represents logic high.

A0, A1, A2, Register Select: Address signals connected to these three inputs select a register for the CPU to read from or write to during data transfer. A table of registers and their addresses is shown below.

iButton is a registered trademark of Dallas Semiconductor Corp., a wholly owned subsidiary of Maxim Integrated Products, Inc.

Register Addresses

A2	A1	A0	Register
0	0	0	Command Register (read/write)
0	0	1	Transmit Buffer (write), Receive Buffer (read)
0	1	0	Interrupt Register (read)
0	1	1	Interrupt Enable Register (read/write)
1	0	0	Clock Divisor Register (read/write)
1	0	1	Control Register (read/write)

$\overline{\text{ADS}}$, Address Strobe: The positive edge of an active Address Strobe ($\overline{\text{ADS}}$) signal latches the Register Select (A0, A1, A2) into an internal latch. Provided that setup and hold timings are observed, $\overline{\text{ADS}}$ may be tied low making the latch transparent.

CLK, Clock Input: This is a (preferably) 50% duty cycle clock that can range from 4 MHz to 128 MHz. This clock provides the timing for the 1-Wire bus.

D7-D0, Data Bus: This bus comprises eight input/output lines. The bus provides bi-directional communications between the 1-Wire master and the CPU. Data, control words, and status information are transferred via this D7-D0 Data Bus.

DQ, 1-Wire Data Line: This open-drain line is the 1-Wire bi-directional data bus. 1-Wire slave devices are connected to this pin. This pin must be pulled high by an external resistor, nominally 5 kΩ.

$\overline{\text{EN}}$, Enable: When $\overline{\text{EN}}$ is low, the 1-Wire master is enabled; this signal acts as the device chip enable. This enables communication between the 1-Wire master and the CPU.

INTR, Interrupt: This line goes to its active state whenever any one of the interrupt types has an active high condition and is enabled via the Interrupt Enable Register. The INTR signal is reset to an inactive state when the Interrupt Register is read.

MR, Master Reset: When this input is high, it clears all the registers and the control logic of the 1-Wire master, and sets INTR to its default inactive state, which is HIGH.

$\overline{\text{RD}}$, Read: This pin drives the bus during a read cycle. When the circuit is enabled, the CPU can read status information or data from the selected register by driving $\overline{\text{RD}}$ low. $\overline{\text{RD}}$ and $\overline{\text{WR}}$ should never be low simultaneously; if they are, $\overline{\text{WR}}$ takes precedence.

$\overline{\text{STPZ}}$, Strong Pull-up Enable: This pin drives the gate of the p-channel transistor which bypasses the weak pull-up resistor in order to provide the slave device with a stiff power supply for high current applications.

$\overline{\text{WR}}$, Write: This pin drives the bus during a write cycle. When $\overline{\text{WR}}$ is low while the circuit is enabled, the CPU can write control words or data into the selected register. $\overline{\text{RD}}$ and $\overline{\text{WR}}$ should never be low simultaneously; if they are, $\overline{\text{WR}}$ takes precedence.

OPERATION – COMMANDS

The 1-Wire Master can generate two special commands on the bus in addition to reading and writing. The first is a 1-Wire reset, which must precede any command given on the bus. Secondly, the 1-Wire Master can be placed into Search ROM Accelerator mode to prevent the host from having to perform single bit manipulations of the bus during a Search ROM operation (0xF0h). For details on the reset or Search

ROM command see [1]. In addition to these two functions, the Command Register contains two bits to bypass the 1-Wire Master features and control the 1-Wire bus directly.

Command Register (Read/Write) **Default: 08h**

Addr. 00h	X	X	X	X	OW_IN	FOW	SRA	1WR
	MSB							**LSB**

Bit 3 - OW_IN: OW Input. This bit always reflects the current state of the 1-Wire line.

Bit 2 - FOW: Force One Wire. This bit can be used to bypass 1-Wire Master operations and drive the bus directly if needed. Setting this bit high will drive the bus low until it is cleared or the 1-Wire Master reset. While the 1-Wire bus is held low no other 1-Wire Master operations will function. By controlling the length of time this bit is set and the point when the line is sampled, any 1-Wire communication can be generated by the host controller. To prevent accidental writes to the bus, the EN_FOW bit in the CONTROL register must be set to a 1 before the FOW bit will function. This bit is cleared to a 0 on power-up or master reset.

Bit 1 - SRA: Search ROM Accelerator. When this bit is set, the 1-Wire Master will switch to Search ROM Accelerator mode. (See "Search ROM Accelerator Description" for the rest of the function description.) When this bit is set to 0, the master will function in its normal mode. This bit is cleared to 0 on a power-up or master reset.

Bit 0 - 1WR: 1-Wire Reset. If this bit is set a reset will be generated on the 1-Wire bus. Setting this bit automatically clears the SRA bit. The 1WR bit will be automatically cleared as soon as the 1-Wire reset completes. The 1-Wire Master will set the Presence Detect interrupt flag (PD) when the reset is complete and sufficient time for a presence detect to occur has passed. The result of the presence detect will be placed in the interrupt register bit PDR. If a presence detect pulse was received PDR will be cleared, otherwise it will be set.

Search ROM Accelerator Description

The Search ROM Accelerator Mode presupposes that a Reset followed by the Search ROM command (0xF0h) has already been issued on the 1-Wire bus. For details on how the Search ROM is actually done in the 1-Wire system, please see [1]. Simply put, the algorithm specifies that the bus master reads two bits (a bit and its complement), then writes a bit to specify which devices should remain on the bus for further processing.

After the 1-Wire Master is placed in Search ROM Accelerator mode, the CPU must send 16 bytes to complete a single Search ROM pass on the 1-Wire bus. These bytes are constructed as follows:

First byte

7	6	5	4	3	2	1	0
r_3	x_3	r_2	x_2	r_1	x_1	r_0	x_0

16^{th} byte

7	6	5	4	3	2	1	0
r_{63}	x_{63}	r_{62}	x_{62}	r_{61}	x_{61}	r_{60}	x_{60}

In this scheme, the index (values from 0 to 63, "n") designates the position of the bit in the ROM ID of a 1-Wire device. The character "x" marks bits that act as a filler and do not require a specific value (don't care bits). The character "r" specifies the selected bit value to write at that particular bit in case of a conflict during the execution of the ROM search.

For each bit position n (values from 0 to 63) the 1-Wire Master will generate three time slots on the 1-Wire bus. These are referenced as:

b0 for the first time slot (read data)
b1 for the second time slot (read data) and
b2 for the third time slot (write data).

The 1-Wire Master determines the type of time slot b2 (write 1 or write 0) as follows:

b2 = r_n if conflict (as chosen by the host)
= b_0 if no conflict (there is no alternative)
= 1 if error (there is no response)

The response bytes that will be in the data register for the CPU to read during a complete pass through a Search ROM function using the Search Accelerator consists of 16 bytes as follows:

first byte

7	6	5	4	3	2	1	0
r'_3	d_3	r'_2	d_2	r'_1	d_1	r'_0	d_0

et cetera

16^{th} byte

7	6	5	4	3	2	1	0
r'_{63}	d_{63}	r'_{62}	d_{62}	r'_{61}	d_{61}	r'_{60}	d_{60}

As before, the index designates the position of the bit in the ROM ID of a 1-Wire device. The character "d" marks the discrepancy flag in that particular bit position. The discrepancy flag will be 1 if there is a conflict or no response in that particular bit position and 0 otherwise. The character "r'" marks the actually chosen path at that particular bit position. The chosen path is identical to b2 for the particular bit position of the ROM ID.

To perform a Search ROM sequence one starts with all bits r_n being 0s. In case of a bus error, all subsequent response bits r'_n are 1's until the Search Accelerator is deactivated by writing 0 to bit 1 of the Command register. Thus, if r'_{63} and d_{63} are both 1, an error has occurred during the search procedure and the last sequence has to be repeated. Otherwise r'_n (n=0…63) is the ROM code of the device that has been found and addressed. When the Search ROM process is complete the SRA bit should be cleared in order to release the 1-Wire Master from Search ROM Accelerator Mode.

For the next Search ROM sequence one re-uses the previous set r_n (n=0…63) but sets r_m to 1 with "m" being the index number of the highest discrepancy flag that is 1 and sets all r_i to 0 with i > m. This process is repeated until the highest discrepancy occurs in the same bit position for two consecutive passes.

EXAMPLE – ACCELERATED ROM SEARCH

In this example, the host will use the 1-Wire Master to identify four different devices on the 1-Wire bus. The ROM data of the devices is as shown (LSB first):

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

1) The host issues a reset pulse by writing 0x01h to the Command Register. All slave devices respond simultaneously with a presence detect.
2) The host issues a Search ROM command by writing 0xF0h to the Transmit Buffer.
3) The host places the 1-Wire Master in Accelerator mode by writing 0x02 to the Command Register.
4) The host writes 0x00h to Transmit Buffer and reads the returning data from the Receive Buffer. This process is repeated for a total of 16 bytes. The data read will contain ROM4 in the r' locations and discrepancy bits set at d0 and d2 as shown (r' locations are underlined, most significant discrepancy is bolded):

RECEIVED DATA 1 = 1000**1**00100000001…

5) The host then de-interleaves the data to arrive at a ROM code of 00010001… with the last discrepancy at location d2.
6) The host writes 0x00h to the Command Register to exit Search Accelerator mode. The host is now free to send a command or read data directly from this device.
7) Steps 1-6 are now repeated to find the next device. The 16 bytes of data transmitted this time are identical to ROM4 up until the last discrepancy bit (d2 in this case) which is inverted and all data following is set to 0 as shown. The received data will contain ROM1 in the r' locations and bits d0 and d2 will be set again:

TRANSMITTED DATA 2 = 0000010000000000…
RECEIVED DATA 2 = 1000**1**10100010001…

8) Since the most significant discrepancy (d2) did not change, the next most (d0) will be used and the process repeats. Further iterations contain the data as shown:

TRANSMITTED DATA 3 = 0100000000000000…
RECEIVED DATA 3 = 11**1**00100010001 00…
TRANSMITTED DATA 4 = 0101000000000000…
RECEIVED DATA 4 = 11**1**10101000100 01…

9) At this point, the most significant discrepancy (d1) did not change so the next most (d0) should be used. However, d0 has now been reached for the second time and since there are no less significant discrepancies, the search is complete having found a total of four devices.

OPERATION – TRANSMITTING / RECEIVING DATA

Data sent and received from the 1-Wire master passes through the transmit/receive buffer location. The 1-Wire master is actually double buffered with separate transmit and receive buffers. Writing to this location connects the Transmit Buffer to the data bus, while reading connects the Receive Buffer to the data bus.

Transmit Buffer (Write) / Receive Buffer (Read) **Default: 00h**

Addr. 01h	Data7	Data6	Data5	Data4	Data3	Data2	Data1	Data0
	MSB							**LSB**

Writing a byte

To send a byte on the 1-Wire bus, the user writes the desired data to the Transmit Buffer. The data is then moved to the Transmit Shift Register where it is shifted serially onto the bus LSB first. A new byte of data can then be written to the Transmit Buffer. As soon as the Transmit Shift Register is empty, the data will be transferred from the Transmit Buffer and the process repeats. Each of these registers has a flag that may be used as an interrupt source. The Transmit Buffer Empty (TBE) flag is set when the Transmit Buffer is empty and ready to accept a new byte. As soon as a byte is written into the Transmit Buffer, TBE is cleared. The Transmit Shift Register Empty (TEMT) flag is set when the shift register has no data in it and is ready to accept a new byte. As soon as a byte of data is transferred from the Transmit Buffer, TEMT is cleared and TBE is set. Remember that proper 1-Wire protocol requires a reset before any bus communication.

Reading a byte

To read data from a slave device, the device must first be ready to transmit data depending on commands already received from the CPU. Data is retrieved from the bus in a similar fashion to a write operation. The host initiates a read by writing to the Transmit Buffer. The data that is then shifted into the Receive Shift Register is the wired-AND of the written data and the data from the slave device. Therefore in order to read a byte from a slave device the host must write 0xFFh. When the Receive Shift Register is full the data is transferred to the Receive Buffer where it can be accessed by the host. Additional bytes can now be read by sending 0xFFh again. If the slave device is not ready to transmit, the data received will be identical to that which was transmitted. The Receive Buffer Register can also generate interrupts. The Receive Buffer flag (RBF) is set when data is transferred from the Receive Shift Register and cleared when the host reads the register. If RBF is set, no further transmissions should be made on the 1-Wire bus or else data may be lost, as the byte in the Receive Buffer will be overwritten by the next received byte. See the timing diagrams for details of the byte reception operation. Generating a 1-Wire reset on the bus is covered under Command Operations. Interrupt flags are explained in further detail under Interrupt Operations. Write and read operations are detailed in the timing diagrams.

OPERATION – FLAGS AND INTERRUPTS

Flags from current status, transmit, receive, and 1-Wire reset operations are located in the Interrupt Register. The Presence Detect flag (PD), OW_LOW, and OW_SHORT are cleared when the Interrupt Register is read. The other flags are cleared automatically when the Transmit and Receive Buffers are written to or read from. All of these flags can generate an interrupt on the INTR pin if the corresponding enable bit is set in the Interrupt Enable Register. To clear the INTR signal, the Interrupt Register must be read. Reading the Interrupt Register always sets the INTR pin inactive even if all flags are not cleared.

Interrupt Register (Read Only) **Default: 0Eh**

Addr. 02h	OW_LOW	OW_SHORT	RSRF	RBF	TEMT	TBE	PDR	PD
	MSB							**LSB**

OW_LOW: One Wire Low. This flag will be set to 1 when the 1-Wire line is low while the master is in idle signaling that a slave device has issued a presence pulse on the 1-Wire (DQ) line. A read to the Interrupt Register will clear this bit.

OW_SHORT: One Wire Short. This flag will be set to a 1 when the 1-Wire line was low before the master was able to send out the beginning of a reset or a time slot. When this flag is 0, it indicates that the 1-Wire line was high as expected prior to all resets and time slots. A read to the Interrupt Register will clear this bit.

RSRF: Receive Shift Register Full. This flag will be set to 1 when there is a byte of data waiting in the Receive Shift Register. When this bit is 0, it indicates that the Receive Shift Register is either empty or currently receiving data. This bit will be cleared by the hardware when data in the Receive Shift Register is transferred to the Receive Buffer. A read to the Interrupt Register will have no effect on this bit.

RBF: Receive Buffer Full. This flag will be set to 1 when there is a byte of data waiting to be read in the Receive Buffer. When this bit is 0, it indicates that the Receive Buffer has no new data to be read. This bit will be cleared when the byte is read from the Receive Buffer. A read to the Interrupt Flag Register will have no effect on this bit. However, following a read of the Interrupt Register, while Enable Receive Buffer Full Interrupt (ERBF) is set to 1, if the ERBF is not cleared and the value is not read from the Receive Buffer, the interrupt will fire again.

TEMT: Transmit Shift Register Empty. This flag will be set to 1 when there is nothing in the Transmit Shift Register and it is ready to receive the next byte of data to be transmitted from the Transmit Buffer. When this bit is 0, it indicates that the Transmit Shift Register is busy sending out data. This bit is cleared when data is transferred from the Transmit Buffer to the Transmit Shift Register. A read to the Interrupt Register will have no effect on this bit.

TBE: Transmit Buffer Empty. This flag will be set to 1 when there is nothing in the Transmit Buffer and it is ready to receive the next byte of data. When it is 0, it indicates that the Transmit Buffer is waiting for the Transmit Shift Register to finish sending its current data before updating it. This bit is cleared when data is written to the Transmit Buffer. A read to the Interrupt Register will have no effect on this bit.

PDR: Presence Detect Result. When a presence detect interrupt occurs, this bit will reflect the result of the presence detect read – it will be 0 if a slave was found, or 1 if no part was found. A read to the Interrupt Register will have no effect on this bit.

PD: Presence Detect. After a 1-Wire Reset has been issued, this flag will be set to 1 after the appropriate amount of time for a presence detect pulse to have occurred. This flag will be 0 when the master has not issued a 1-Wire Reset since the previous read of the Interrupt Register. This bit is cleared when the Interrupt Register is read.

OPERATION – FLAGS AND INTERRUPTS (continued)

The Interrupt Enable Register allows the system programmer to specify the source of interrupts which will cause the INTR pin to be active, and to define the active state for the INTR pin. When a Master Reset is received **all** bits in this register are cleared to 0 disabling all interrupt sources and setting the active state of the INTR pin to LOW. This means the INTR pin will be pulled high since all interrupts are disabled. The INTR pin is reset to an inactive state by reading the Interrupt Register.

Interrupt Enable Register (Read / Write) **Default: 00h**

Addr. 03h	EOWL	EOWSH	ERSF	ERBF	ETMT	ETBE	IAS	EPD
	MSB							**LSB**

EOWL: Enable One Wire Low Interrupt. Setting this bit to a 1 enables the One Wire Low Interrupt. If set, INTR will be asserted when the OW_LOW flag is set. Clearing this bit disables OW_LOW as an active interrupt source.

EOWSH: Enable One Wire Short Interrupt. Setting this bit to a 1 enables the One Wire Short Interrupt. If set, INTR will be asserted when the OW_SHORT flag is set. Clearing this bit disables OW_SHORT as an active interrupt source.

ERSF: Enable Receive Shift Register Full Interrupt. Setting this bit to a 1 enables the Receive Shift Register Full Interrupt. If set, INTR will be asserted when the RSRF flag is set. Clearing this bit disables RSRF as an active interrupt source.

ERBF: Enable Receive Buffer Full Interrupt. Setting this bit to a 1 enables the Receive Buffer Full Interrupt. If set, INTR will be asserted when the RBF flag is set. Clearing this bit disables RBF as an active interrupt source.

ETMT: Enable Transmit Shift Register Empty Interrupt. Setting this bit to a 1 enables the Transmit Shift Register Empty Interrupt. If set, INTR will be asserted when the TEMT flag is set. Clearing this bit disables TEMT as an active interrupt source.

ETBE: Enable Transmit Buffer Empty Interrupt. Setting this bit to a 1 enables the Transmit Buffer Empty Interrupt. If set, INTR will be asserted when the TBE flag is set. Clearing this bit disables TBE as an active interrupt source.

IAS: INTR Active State. This bit determines the active state for the INTR pin. If this bit is a 1, the INTR pin is active high; if it is a 0, the INTR pin is active low.

EPD: Enable Presence Detect Interrupt. Setting this bit to a 1 enables the Presence Detect Interrupt. If set, INTR will be asserted when the PD flag is set. Clearing this bit disables PD as an active interrupt source.

OPERATION – CLOCK DIVISOR

All 1-Wire timing patterns are generated using a base clock of 1.0 MHz. The 1-Wire Master will generate this clock frequency internally given an external reference on the CLK pin. The external clock must have a frequency from 4 to 128 MHz and a 50% duty cycle is preferred. The Clock Divisor Register controls the internal clock divider and provides the desired reference frequency. This is done in two stages: first a prescaler divides by 1, 3, 5, or 7, then the remaining circuitry divides by 2, 4, 8, 16, 32, 64, or 128.

Clock Divisor Register **Default: 00h**

Addr. 04h	CLK_EN	X	X	DIV2	DIV1	DIV0	PRE1	PRE0
	MSB							**LSB**

The clock divisor must be configured before communication on the 1-Wire bus can take place as well as setting the CLK_EN bit to a 1. This register is set to 0x00h if a master reset occurs. Use the table below to find the proper register value based on the CLK reference frequency. For example, the user would write 0x87h to this location when providing a 15 MHz input clock.

Clock Divisor Register Settings for Input Clock Rates

Min CLK Frequency (MHz)	Max CLK Frequency (MHz)	Divider Ratio	DIV2	DIV1	DIV0	PRE1	PRE0
4.0	< 5.0	4	0	1	0	0	0
5.0	< 6.0	5	0	0	0	1	0
6.0	< 7.0	6	0	0	1	0	1
7.0	< 8.0	7	0	0	0	1	1
8.0	< 10.0	8	0	1	1	0	0
10.0	< 12.0	10	0	0	1	1	0
12.0	< 14.0	12	0	1	0	0	1
14.0	< 16.0	14	0	0	1	1	1
16.0	< 20.0	16	1	0	0	0	0
20.0	< 24.0	20	0	1	0	1	0
24.0	< 28.0	24	0	1	1	0	1
28.0	< 32.0	28	0	1	0	1	1
32.0	< 40.0	32	1	0	1	0	0
40.0	< 48.0	40	0	1	1	1	0
48.0	< 56.0	48	1	0	0	0	1
56.0	< 64.0	56	0	1	1	1	1
64.0	< 80.0	64	1	1	0	0	0
80.0	< 96.0	80	1	0	0	1	0
96.0	< 112.0	96	1	0	1	0	1
112.0	< 128.0	112	1	0	0	1	1

OPERATION – Master Control

Various 1-Wire devices utilize different communication protocols in order to properly function and report back to the master in an efficient manner. The Control Register adds the robustness needed to handle all of the major conditions expected from each iButton and 1-Wire chip family.

Control Register **Default: 00h**

Addr. 05h	X	OD	BIT_CTL	STP_SPLY	STPEN	EN_FOW	PPM	LLM
	MSB							**LSB**

OD: Overdrive. Setting this bit to a 1 will place the master into Overdrive mode that effectively changes the master's 1-Wire timings to match those outlined for Overdrive in the **Book of iButton Standards.** Clearing this bit to a 0 leaves the master operating in standard mode speed.

BIT_CTL: Bit Control. Setting this bit to a 1 will place the master into its "Bit Banging" mode of operation. In this mode, only the least significant bit of the Transmit/Receive register would be sent/received before enabling the interrupt flags that signal the end of the transmission. Clearing this bit to 0 leaves the master operating in full byte boundaries.

STP_SPLY: Strong Pull-up Supply. Setting this bit to a 1 while STPEN is also set to a 1 will enable the STPZ output while the master is in an IDLE state. This will provide a stiff supply to devices requiring high current during operations. Clearing this bit to a 0 disables the STPZ output while the master is in an IDLE state. The STP_SPLY bit is a don't care if STPEN is set to a 0.

STPEN: Strong Pull-up Enable. Setting this bit to a 1 enables the strong pull-up output enable (STPZ) pin's functionality which allows this output pin to enable an external strong pull-up any time the master is not pulling the line low or waiting to read a value from a slave device. This functionality is used for meeting the recovery time requirement in Overdrive mode and long-line standard communications. Clearing this bit to a 0 will disable the STPZ output pin.

EN_FOW: Enable Force One Wire. Setting this bit to a 1 will enable the functionality of the Force One Wire (FOW) register bit in the Command Register. Clearing this bit will disable the functionality of the FOW bit.

PPM: Presence Pulse Masking Mode. Setting this bit to a 1 will enable Presence Pulse Masking Mode. This mode causes the master to initiate the falling edge of a presence pulse during a 1-Wire Reset before the fastest slave would initiate one. This enables the master to prevent the larger amount of ringing caused by the slave devices when initiating a low on the DQ line. If the PPM bit is set, the PDR result bit in the Interrupt Register will always be set to a 0 showing that a slave device was on the line even if there were none. Clearing this bit to a 0 disables the Presence Pulse Masking Mode.

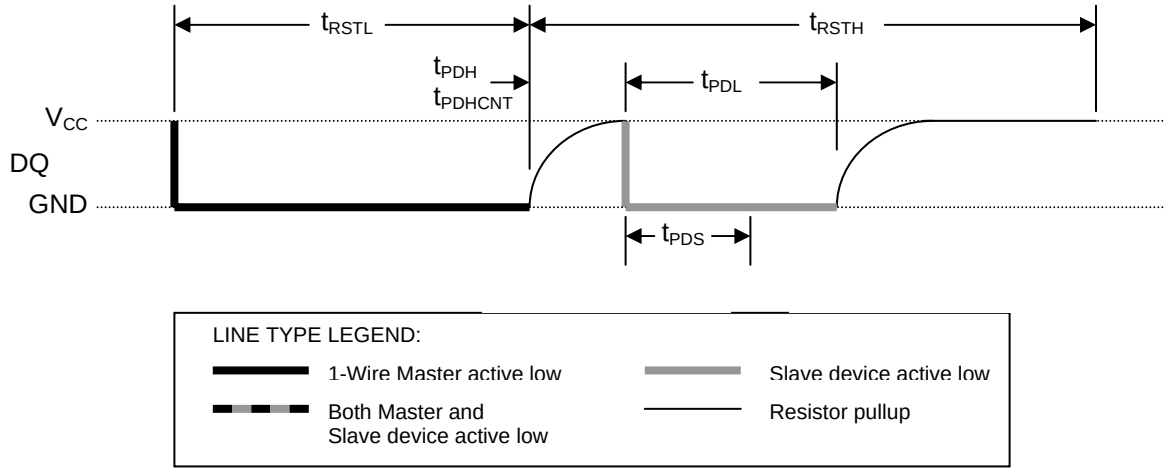
LLM: Long Line Mode. Setting this bit to a 1 will enable Long Line Mode timings on the 1-Wire line during standard mode communications. This mode effectively moves the write one release, the data sampling, and the time slot recovery times out to roughly 8µs, 22µs, and 14µs respectively. This provides a less strict environment for long line transmissions. Clearing this bit to 0 leaves the write one release, the data sampling, and the time slot recovery times at roughly 5µs, 15µs, and 7µs respectively.

I/0 SIGNALING

The 1-Wire bus requires strict signaling protocols to insure data integrity. The four protocols used by the 1-Wire Master are the initialization sequence (Reset Pulse followed by Presence Pulse), Write 0, Write 1, and Read Data. The master initiates all of these types of signaling except the Presence Pulse.

The initialization sequence required to begin any communication with the bus slave is shown in Figure 1. A Presence Pulse following a Reset Pulse indicates the slave is ready to accept a ROM Command. The 1-Wire Master transmits a reset pulse for t_{RSTL}. The 1-Wire bus line is then pulled high by the pull-up resistor. After detecting the rising edge on the DQ pin, the slave waits for t_{PDH} and then transmits the Presence Pulse for t_{PDL}. The master samples the bus at t_{PDS} after the slave responds to test for a valid presence pulse or after waiting for t_{PDHCNT} following the start of t_{RSTH}. The result of this sample is stored in the PDR bit of the Interrupt Register. The reset time slot ends t_{RSTH} after the master releases the bus.

1-WIRE INITIALIZATION SEQUENCE (RESET PULSE AND PRESENCE PULSE) Figure 1.



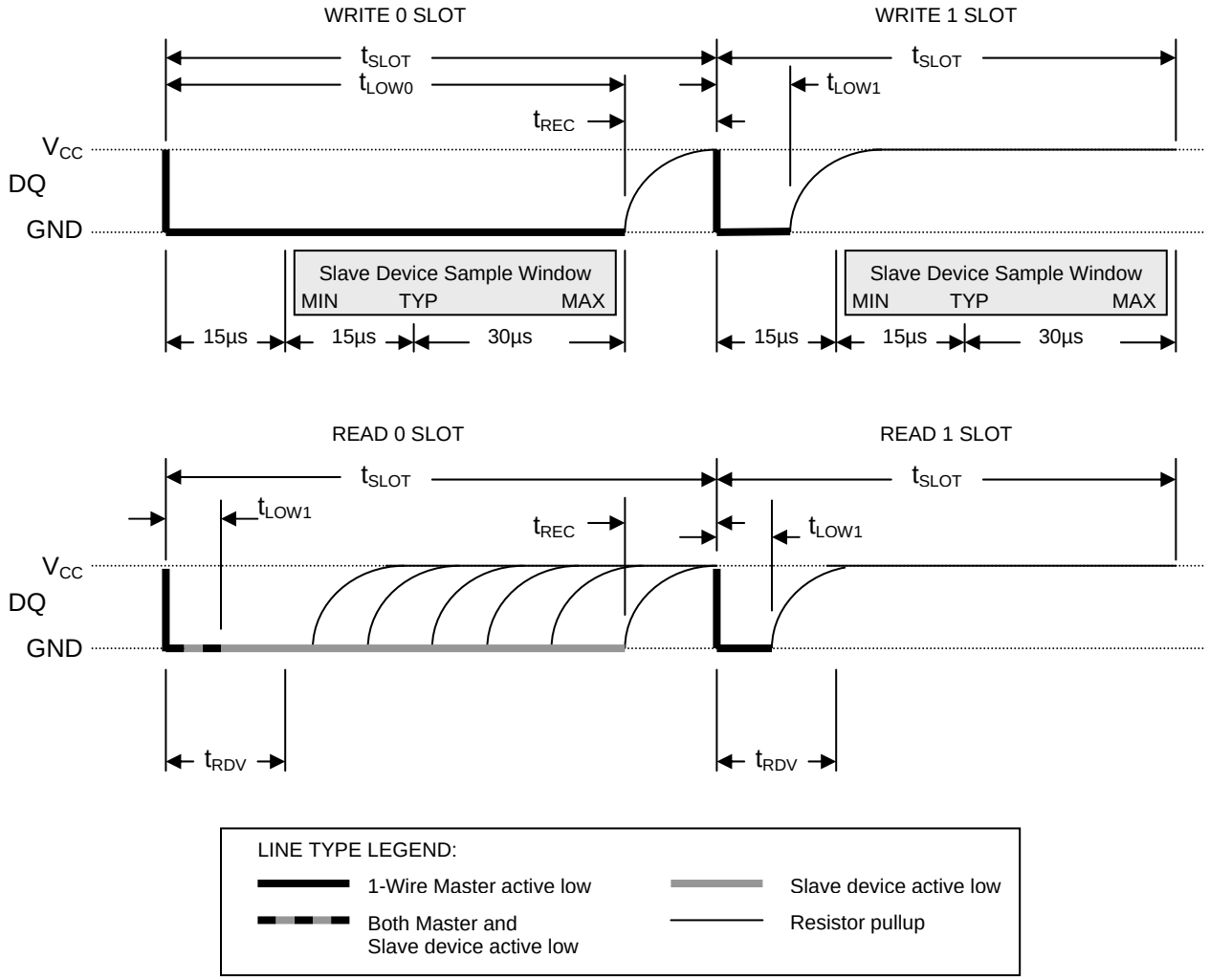
WRITE TIME SLOTS

A write time slot is initiated when the 1-Wire Master pulls the 1-Wire bus line from a logic high (inactive) level to a logic low level. The master generates a Write 1 time slot by releasing the line at t_{LOW1} and allowing the line to pull up to a logic high level. The line is held low for t_{LOW0} to generate a Write 0 time slot. A slave device will sample the 1-Wire bus line between 15 and 60µs after the line falls. If the line is high when sampled, a Write 1 occurs. If the line is low when sampled, a Write 0 occurs (see Figure 2).

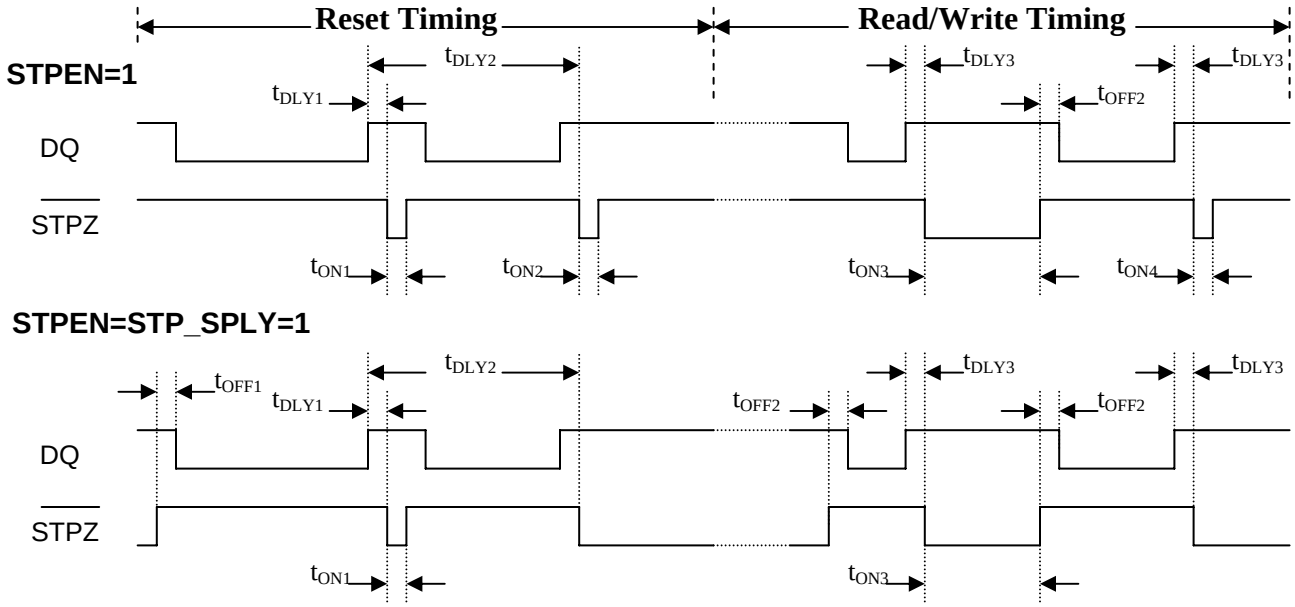
READ TIME SLOTS

A read time slot is initiated when the 1-Wire Master pulls the bus low for at least 1µs and then releases it. If the slave device is responding with a 0 it will continue to hold the line low for up to 60µs, otherwise it will release it immediately. The master will sample the data t_{RDV} from the start of the read time slot. The master will end the read slot after a time of $t_{SLOT.}$ See Figure 2 for more information.

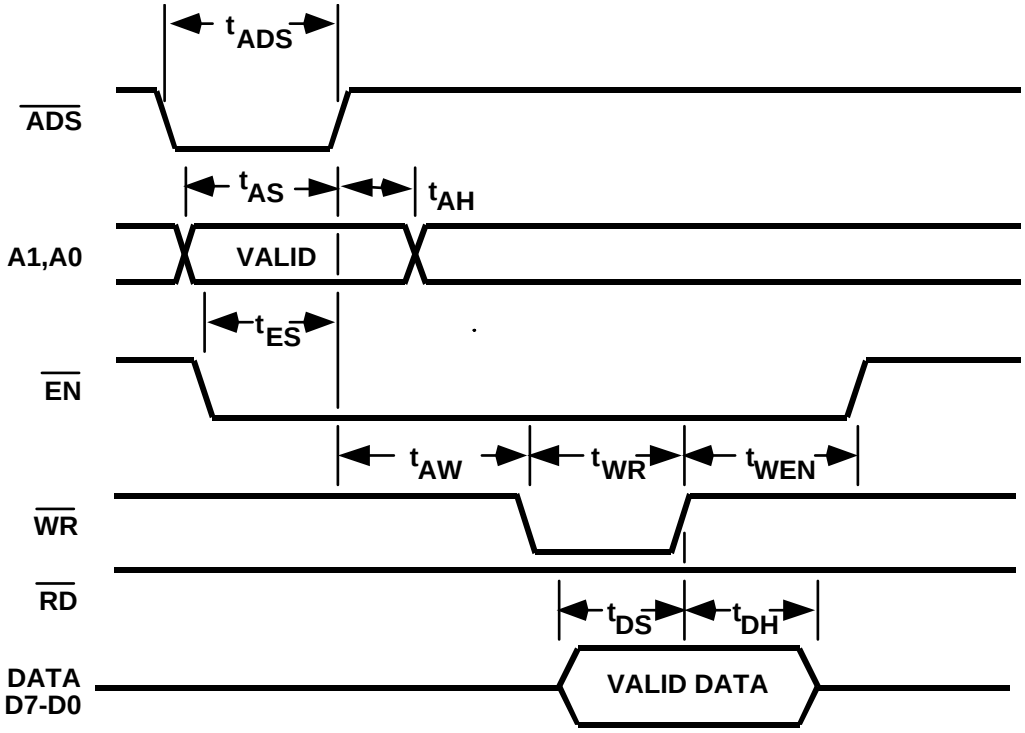
1-WIRE WRITE AND READ TIME SLOTS Figure 2.



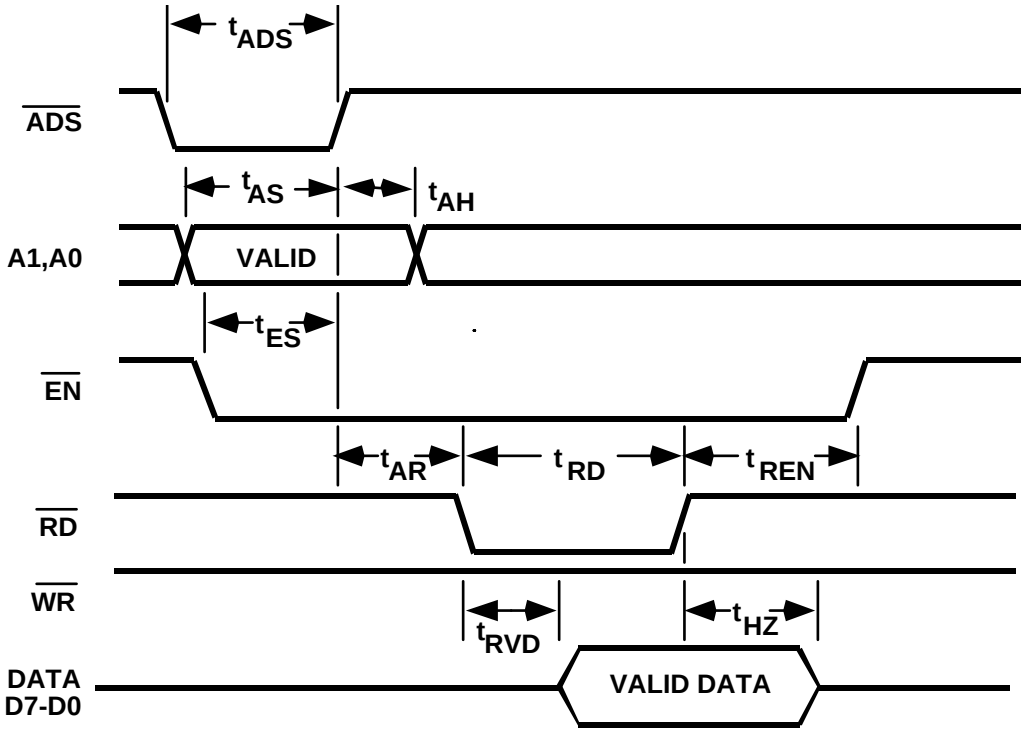
1-WIRE STPZ RESET AND READ WRITE TIMING Figure 3.



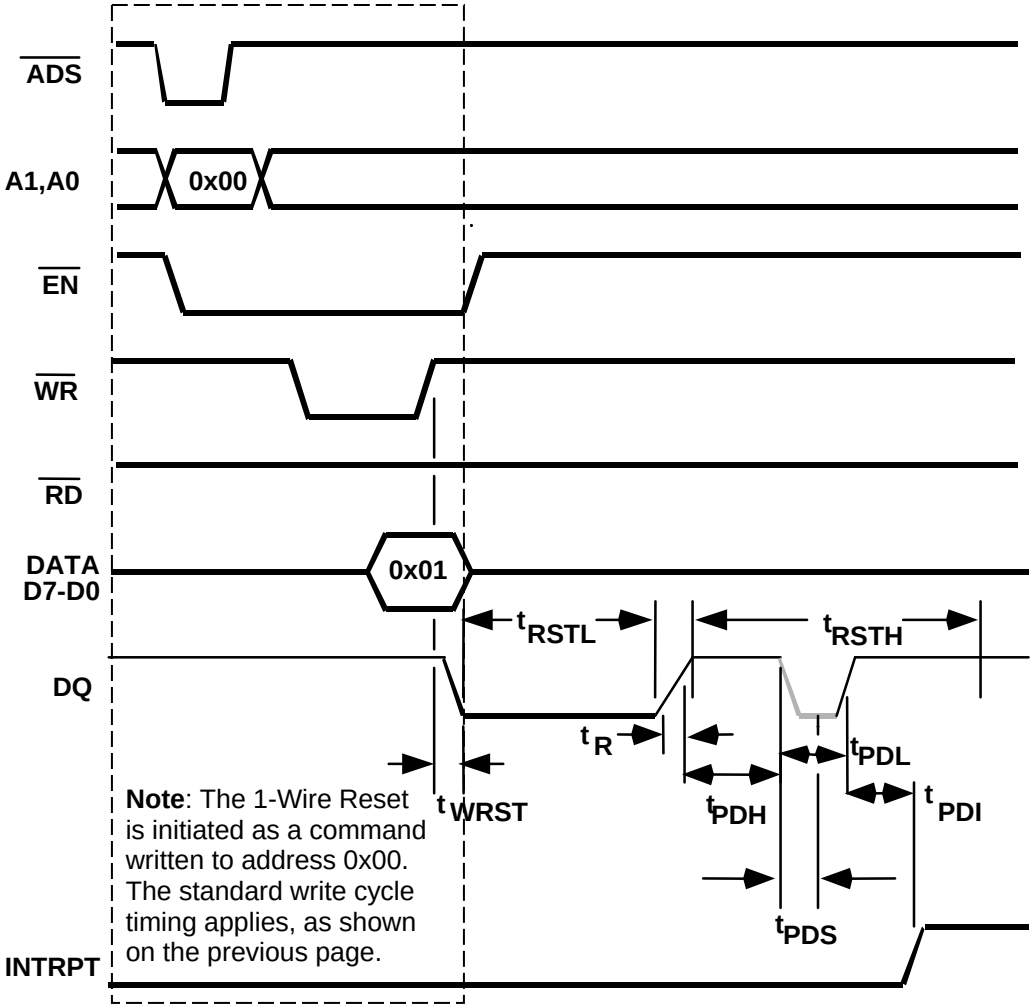
Write Cycle



Read Cycle



GENERATING A 1-WIRE RESET



TIMING SPECIFICATIONS:

Data Bus Interface Timing

Symbol	Parameter	Conditions	Min	Max	Units
t_{ADS}	Address Strobe Width	Note 1,3	60		ns
t_{AH}	Address Hold Time	Note 1,3	0		ns
t_{AR}	Address Latch to Read	Note 1,3	60		ns
t_{AS}	Address Setup Time	Note 1,3	60		ns
t_{AW}	Address Latch to Write	Note 1,3	60		ns
t_{DH}	Data Hold Time	Note 1	30		ns
t_{DS}	Data Setup Time	Note 1	30		ns
t_{ES}	Enable Setup Time	Note 1	60		ns
t_{HZ}	RD to Floating Data Delay	Note 1	0	100	ns
t_{PDI}	Presence Detect to INTR	Note 1	0	100	ns
t_{RD}	RD Strobe Width	Note 1	125		ns
t_{REN}	Enable Hold Time from RD	Note 1	20		ns
t_{RVD}	Delay from RD to Data	Note 1		60	ns
t_{WEN}	Enable Hold Time from WR	Note 1	20		ns
t_{WR}	WR Strobe Width	Note 1	100		ns
t_{WRST}	WR High to Reset	Note 1	0	100	ns

One Wire Interface Timing (Note 2)

Symbol	Parameter	Conditions	Min	Max	Units
t_{LOW0}	Write 0 Low Time	Standard	62.4	78	µs
		Overdrive	8	10	µs
t_{LOW1}	Write 1 Low Time	Standard	4.8	6	µs
		Standard – Long Line Mode	7.2	9	µs
		Overdrive	0.8	1	µs
t_{PDH}	Presence Detect High	Standard	15	60	µs
		Overdrive	2	6	µs
t_{PDL}	Presence Detect Low	Standard	60	240	µs
		Overdrive	6	24	µs
t_{PDS}	Presence Detect Sample	Standard	24	31	µs
		Standard – Long Line Mode	30.4	38	µs
		Overdrive	2.4	4	µs
t_{RDV}	Read Data Value	Standard	12	15	µs
		Standard – Long Line Mode	20	25	µs
		Overdrive	1.6	2	µs
t_{REC}	Recovery Time	Standard	6.4	8	µs
		Standard – Long Line Mode	11.2	14	µs
		Overdrive	4	5	µs
t_{RSTH}	Reset Time High	Standard	508.8	636	µs
		Overdrive	59.2	74	µs
t_{RSTL}	Reset Time Low	Standard	500.8	626	µs
		Overdrive	50.4	63	µs
t_{SLOT}	Time Slot	Standard	68.8	86	µs
		Overdrive	12	15	µs
τ	Timebase Period		0.8	1	µs

One Wire Strong Pull-up Control (STPZ) Timing (Note 2)

Symbol	Parameter	Conditions	Min	Max	Units	Notes
t_{ON1}	Active Time for Presence Detect	Standard	6.4	8	µs	
		Overdrive	0.8	1	µs	
t_{ON2}	Active Time for Presence Detect Recovery	Standard	8	10	µs	
		Overdrive	8	10	µs	
t_{ON3}	Active Time for Write 1 Recovery	Standard	51.2	64	µs	4,5
		Overdrive	7.2	9	µs	4,5
t_{ON4}	Active Time for Write 0 Recovery	Standard	6.4	8	µs	4,5
		Overdrive	0.8	1	µs	4,5
t_{DLY1}	Delay Time for Presence Detect	Standard	0.8	1	µs	
		Overdrive	0.8	1	µs	
t_{DLY2}	Delay Time for Presence Detect Recovery	Standard	399.2	499	µs	6
		Overdrive	31.2	39	µs	6
t_{DLY3}	Delay Time for Write1/Write0 Recovery	Standard	0.8	1	µs	
		Overdrive	0.8	1	µs	
t_{OFF1}	Turn Off Time for 1-Wire Reset	Standard	1.6	2	µs	
		Overdrive	1.6	2	µs	
t_{OFF2}	Turn Off Time for Write1/Write0	Standard	0.8	1	µs	7
		Overdrive	0.8	1	µs	7

NOTES:

1. These values will depend upon the process used to realize the circuit. Values shown are for example purposes only.
2. The 1-Wire Master timing values are the times depending upon the internal logic. If the I/O drivers are slow, these times will change accordingly.
3. If $\overline{ADS}$ is tied low, t_{AR} and t_{AW} are referred from t_{ES} ; thus $\overline{RD}$ or $\overline{WR}$ must occur at least $t_{ES} + t_{AR}$ or $t_{ES} + t_{AW}$ after $\overline{EN}$ goes low.
4. There is no timing difference for sending out and receiving bits within a byte. The difference comes when the last bit of the byte is finished being sent out. At this point, the signal is either enabled continuously until the next reset or time slot begins, or enabled only for t_{on3} or t_{on4}.
5. When performing a read versus a write time slot, the master provides the same active time for write 1 and write 0. However, the input from the DQ line is sensed every 1µs for a high value. If DQ is high, the STPZ signal is enabled. If the DQ line is low, the STPZ signal remains disabled until the high is sensed. In all write time slots, a high is sensed immediately.
6. This parameter is the time delay until the master begins to monitor the DQ input level. If the line is already high, then STPZ will be enabled. If not, it will wait to enable STPZ until the next state machine clock after the DQ line has recovered.
7. The very first bit in a byte transmission will have an extended t_{OFF2} of 4µs due to the order of states the master's state machine runs through.

REFERENCES:

[1] *Book of iButton Standards*, Dallas Semiconductor, online at http://www.ibutton.com/iButtons/standard.pdf

REVISION HISTORY

0.1 – June 15, 1999

1. First release

0.2 – July 12, 1999

1. Clarification added that $\overline{RD}$ and $\overline{WR}$ should never be low simultaneously; if they are, $\overline{WR}$ takes precedence.
2. First draft of search ROM driver code example added in section 6.0.

0.3 – August 17, 1999

1. $\overline{EN}$ is **not** latched by $\overline{ADS}$; $\overline{ADS}$ only controls an internal address latch, which may be made transparent by tying $\overline{ADS}$ low. $\overline{EN}$ is a level-enabled signal, and does not need to be latched.
2. Changed lower clock rate to 3.1 MHz. Updated timing specifications to reflect this.
3. Removed DIVSEL0, DIVSEL1, and DIVSEL2 pins. Clock division selection is now performed by writing to the Clock Divisor Register; this makes it necessary to add an additional address line, A2, in order to select this register.
4. The 1-Wire master is now double-buffered for receive and transmit operations; the data register is no longer a physical register but two registers – the transmit buffer and the receive buffer. These two buffers are memory mapped to the same location, where a write operation selects the transmit buffer, and a read selects the receive buffer. Flags TBE, TEMT, RBF, and RSRF are defined to signal when buffers are empty or full.
5. Setting the 1-Wire reset bit in the command register now automatically disables the Search ROM accelerator bit.
6. Interrupts are automatically cleared by reading the interrupt register.
7. The result of a presence detect is now reported in the Interrupt Register instead of in the data register. This allows the PD interrupt service routine to read the result of a presence detect interrupt when it reads the interrupt register.
8. Changed the way the Interrupt and Interrupt Enable Registers work – it was backward initially. The Interrupt Register now is more of a status register, whose bits get ANDed with the Interrupt Enable Register to determine if any of the flags set in that register cause the INTR pin to go active. The active state of the INTR pin is now programmable; default is HIGH on Master Reset.
9. Removed example code. Will need to be rewritten to comprehend changes made in specification of the hardware device.

0.4 – August 20, 1999

1. Clarified that Master Reset causes INTR to go to its inactive state. This is further clarified in section 4.5 by defining the reset state of the Interrupt Enable Register as cleared to all zeros, masking all interrupt sources and defining the active state of the INTR pin as low. This implies that INTR will go HIGH upon MR.
2. By restricting the lower clock frequency to 3.2MHz, internal timing can now be between 1µs and 1.25µs.
3. Corrected several grammatical, typographical, and spelling errors.
4. Since the internal clock is the result of different division ratios, the duty cycle may not be 50%. This is not a problem for the circuit, so all references to high and low times of the internal timebase have been deleted. The internal clock period is now referred to as τ, to simplify timing diagrams.
5. Timing diagrams have been updated to reflect changes in nomenclature, and to clarify timing.
6. Note 4 added to timing specification table.
7. Section 6 renamed to " Applications Hints and Examples". Notes were added in this section regarding 1-Wire wave shaping and power management.

0.5 – August 24, 1999

1. In Section 1.0, changed “four registers” to “five registers”, to reflect current actual register count. Note that receive and transmit registers are actually 1 register in the memory map.
2. Corrected block diagram to reflect 1.25µs timebase maximum.
3. Removed voltage specifications on logic high and low in Section 3.0, as these will be process-specific.
4. Changed lower clock rate to 3.2MHz in the text description of the CLK pin.
5. Removed t_R specification.
6. t_{PDS} is now specified from the falling edge of the DQ line after the line has been released by the master. The master will wait for up to 60µs to detect a falling edge; but if the edge occurs before 60µs, the master will wait 30µs after that edge to sample the data line to read the presence detect.
7. Fixed errors in timing specification table left over from the change to maximum internal timebase period of 1.25µs.

0.6 – September 17, 1999

1. TBE and TEMT default states changed to 1 instead of 0, to reflect their actual state (empty) upon a master reset.

1.0 – September 20, 1999

1. Changed the operation of the interrupt register. The RBF and RSRF flags are no longer automatically cleared when a read operation is performed on the Interrupt Register. Doing so would allow for data to be overwritten if the interrupt handler did not do a read of the receive buffer immediately following the interrupt. These flags are now cleared when the data has actually been read or shifted out.

2. Revision 1.000 of the VHDL code is complete, and this specification reflects the current operation of that VHDL code. Thus, the revision number for this specification is changed to 1.0 and t OPERATION

31 October, 1999.

3. Revision 1.2 of the datasheet complete. Format has been changed to look like a standard DS datasheet. Verilog version of the code is also complete. Both types undergoing testing on the bench.

21 November, 2000.

1. Receive Data no longer double buffered. Data is now immediately transferred to the Receive Buffer even if already full.
2. Shift Register flags removed from the Interrupt and Interrupt Enable registers. No longer needed.
3. DQO and DQI added to the Command Register to control the bus directly.
4. Low pulse widths for a write 1 changed from 1µs to 2µs and for a write 0 from 75µs to 30µs.
5. Interrupt Register – Flags are no longer cleared on a read except for PD. INTR is set inactive automatically on a read.
6. Gate count updated to 1492 to reflect newest revision on page 1.

13 September, 2001.

1. Updated specification to match the changes made to the Verilog version of the master in the DS80C400. These changes have been placed into the Verilog standalone version for which this document pertains.
2. Added new interrupts, the Control register, and all of the features included in the Control register.

18 August, 2004

1. Updated Verilog version to not hardwire IAS bit. Was hard-wired to active-low. It now performs as specified in this datasheet.
2. VHDL Version updated to match this new spec, version number in source updated to 2.0.

3. Gate count updated to 3470 to reflect newest revision on page 1.

14 May 2007
1. First release of revamped Verilog Testbench
2. Re-write of VHDL with RTL confirmed equivalent to Verilog