OMAP5910 Dual-Core Processor HDQ/1-Wire Interface Reference Guide

Literature Number: SPRU688
October 2003

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Preface

Read This First

About This Manual

The HDQ and 1-Wire battery monitoring serial interface module in the OMAP5910 device implements the hardware protocol of the master function of the TI/Benchmarq HDQ and the Dallas Semiconductor 1-Wire protocols. The module utilizes a command structure that is programmed into transmit−command registers. The received data is in the received−data register. The firmware is responsible for doing the correct sequencing in the command registers. The module only implements the hardware−interface layer of the protocols.

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h. For example, the following number is 40 hexadecimal (decimal 64): 40h.

Related Documentation From Texas Instruments

The following documents describe the OMAP5910 device and related peripherals. Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

OMAP5910 Dual-Core Processor MPU Subsystem Reference Guide (literature number SPRU671)

OMAP5910 Dual-Core Processor DSP Subsystem Reference Guide (literature number SPRU672)

OMAP5910 Dual-Core Processor Memory Interface Traffic Controller Reference Guide (literature number SPRU673)

OMAP5910 Dual-Core Processor System DMA Controller Reference Guide (literature number SPRU674)

OMAP5910 Dual-Core Processor LCD Controller Reference Guide (literature number SPRU675)

OMAP5910 Dual-Core Processor Universal Asynchronous Receiver/ Transmitter (UART) Devices Reference Guide (literature number SPRU676)

OMAP5910 Dual-Core Processor Universal Serial Bus (USB) and Frame Adjustment Counter (FAC) Reference Guide (literature number SPRU677)

OMAP5910 Dual-Core Processor Clock Generation and System Reset Management Reference Guide (literature number SPRU678)

OMAP5910 Dual-Core Processor General-Purpose Input/Output (GPIO) Reference Guide (literature number SPRU679)

OMAP5910 Dual-Core Processor MMC/SD Reference Guide (literature number SPRU680)

OMAP5910 Dual-Core Processor Inter-Integrated Circuit (I2C) Controller Reference Guide (literature number SPRU681)

OMAP5910 Dual-Core Processor Timer Reference Guide (literature number SPRU682)

OMAP5910 Dual-Core Processor Inter-Processor Communication Reference Guide (literature number SPRU683)

OMAP5910 Dual-Core Processor Camera Interface Reference Guide (literature number SPRU684)

OMAP5905 Dual-Core Processor Multichannel Serial Interface (MCSI) Reference Guide (literature number SPRU685)

OMAP5910 Dual-Core Processor Micro-Wire Interface Reference Guide (literature number SPRU686)

OMAP5910 Dual-Core Processor Real-Time Clock (RTC) Reference Guide (literature number SPRU687)

OMAP5910 Dual-Core Processor HDQ/1-Wire Interface Reference Guide (literature number SPRU688)

OMAP5910 Dual-Core Processor PWL, PWT, and LED Peripheral Reference Guide (literature number SPRU689)

OMAP5910 Dual-Core Processor Multichannel Buffered Serial Port (McBSP) Reference Guide (literature number SPRU708)

Trademarks

OMAP and the OMAP symbol are trademarks of Texas Instruments.

Contents

Figures

Tables

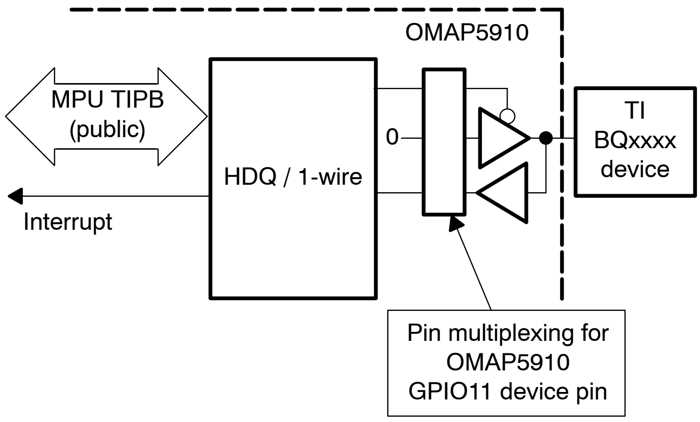
HDQ/1-Wire Interface

1 HDQ and 1-Wire Battery Monitoring Serial Interface

The HDQ and 1-Wire battery monitoring serial interface module in the OMAP5910 device implements the hardware protocol of the master function of the TI/Benchmarq HDQ and the Dallas Semiconductor 1-Wire protocols. The module utilizes a command structure that is programmed into transmit-command registers. The received data is in the received-data register. The firmware is responsible for doing the correct sequencing in the command registers. The module only implements the hardware-interface layer of the protocols. (See section 2 below for a description of the HDQ and 1-Wire Protocols.)

The HDQ and the 1-Wire modes are selectable in software, which must be done before any transmit and receive from the module is performed. The mode is assumed static during operation of the device.

Figure 1. HDQ and 1-Wire Overview



1.1 Software Interface

The base address for the HDQ regsiters is FFFB:C000.

No synchronization is provided by the hardware between the register-clock domain and the state-machine domain. This means that during a read the hardware has the capability to modify the receive buffer and it is also possible that any access to the transmit-write-data register corrupts the data that is being sent if a transmit is being performed.

However, these hazards can be avoided in software by observing the following limitations:

- A read is not performed from the interrupt status register or receive-buffer register unless the processor has been interrupted by the peripheral.
- After the release of the Go bit in the control-and-status register, no access to the TX-write-data buffer or the control-and-status registers is performed until the processor has been interrupted by the peripheral.
- Polling of the interrupt-status register is not allowed by software to determine if an interrupt was generated.
- No register access can be done to the module registers after the software puts the module in power-down mode (by setting bit 5 of the control-and-status register to 0), except to re-enable the clock.

Table 1. Register Memory Map Summary

Address	Name	Width	Type
FFFB:C000	TX-write data register	8	R/W
FFFB:C004	RX-receive buffer register	8	R
FFFB:C008	Control and status register	8	R/W
FFFB:C00C	Interrupt status, read to clear register	8	R/W

Table 2. TX Write Data Register Field Descriptions

Bits	Field	Value	Description	Access Type	Reset Value
31–8	Reserved		Reserved	R	000000h
7–0	Data		Write data (Used in both the HDQ and 1-Wire modes)	R/W	00h

Table 3. RX-Buffer Register Field Descriptions

Bits	Field	Value	Description	Access Type	Reset Value
31−8	Reserved		Reserved	R	XXXXXXh
7−0	Data		Next received character. HDQ module uses bits 7:0. 1-Wire module uses only bit 0.	R	XXh

Table 4. Interrupt Status Register Field Descriptions

Bits	Field	Value	Description	Access Type	Reset Value
31−8	Reserved		Reserved. Bit is set to 1 if cause of interrupt. Read clears all interrupts that have been set.	R	000000h
7−3	Reserved		Reserved—read 0s, writes ignored	R	0h
2	TX.INT		TX completed	R/C	0
1	RX.INT		Read complete	R/C	0
0	PDT0.INT		Presence detect/time-out: In the 1-Wire mode this is due to the presence detect, and in the HDQ mode this is due to the time-out on read.	R/C	0

Table 5. Control-and-Status Register Field Descriptions

Bits	Field	Value	Description	Access Type	Reset Value
31−8	Reserved		Reserved	R	000000h
7	SB	0	Byte mode for HDQ	R/W	0
		1	Single-bit mode for 1-Wire		
6	INTEN		Interrupt enable	R/W	0
		0	Disable interrupts		
		1	Enable interrupts		
5	PD		Power-down mode	R/W	0
		0	Disable clocks		
		1	Enable clocks		

Table 5. Control-and-Status Register Field Descriptions (Continued)

Bits	Field	Value	Description	Access Type	Reset Value
4	GO		Go bit Write 1 to send the appropriate commands. Bit returns to 0 after the command is complete.	R/W	0
3	PRES		Presence detect received, 1-Wire mode only.	R	0
		0	Not detected		
		1	Detected		
2	INIT		Write 1 to this bit, and set the Go bit, to send Initialization pulse. Bit returns to 0 after pulse is sent.	R/W	0
1	R/W		R/W bit (determines if next command is read or write)	R/W	0
		0	Write		
		1	Read		
0	MODE		Mode	R/W	0
		0	HDQ		
		1	1-Wire		

2 HDQ and 1-Wire Protocols

As stated earlier, the HDQ and the 1-Wire modes are selectable in software, and this must be done before any transmit and receive from the module is performed. The mode is assumed static during operation of the device. From a timing perspective, both the 1-Wire and the HDQ protocols use HDQ timing.

2.1 Functional Description

The interface module is designed to work with both the HDQ and the 1-Wire protocols. The protocols use a single wire to communicate between the master and the slave. The protocols employ a return-to-1 mechanism, where after any command the line is pulled up to a high. This necessitates an external pullup.

An open-drain configuration is used on the wire. Therefore, the HDQ pin only actively drives low and goes to the high-impedance state, otherwise allowing an external pullup resistor to pull the wire high.

The MODE bit of the control-and-status register selects whether the HDQ or the 1-Wire protocol is to be used. This bit should not be modified during active data cycles on the interface. Therefore, it is recommended that the bit be only modified as part of boot-up configuration. HDQ is the default mode.

2.1.1 Receive and Transmit Operation

The receive and the transmit operations are performed with respect to the timing that is specified in the HDQ protocol. This is done to keep the hardware interface section compatible between the two devices. In essence the 1-Wire mode runs at slower speeds than the capabilities of the mode. The differences between the protocol at the hardware layer are described in the following subsections.

2.1.1.1 HDQ Mode (Default)

In the HDQ mode, the firmware does not require the host to create an initialization pulse to the slave. However, the slave can be reset using an initialization pulse (also referred to as a break pulse). The pulse is created by setting the appropriate bit in the control-and-status register. The slave does not respond with a presence pulse, as it does in the 1-Wire protocol.

In a typical write to the slave, two bytes of data are sent to the slave. This is the command/address byte followed by the data that must be written. In a typical read, one command/address byte is sent to the slave, and the slave returns a byte of data.

The master implementation is a byte engine. The sending of the ID, command/address, and data is the responsibility of the firmware. The master engine provides for only one data TX register.

HDQ is a return-to-1 protocol. This means that after a data byte (either command/address + write data for writes, or just command/address for reads) is sent to the slave, the host pulls the line high. This is accomplished in the OMAP5910 device by setting the line to high (with an external pullup). The slave pulls the line low to initiate a transaction. This is the case when a read happens and the slave must send the read data back to the host.

If the host initiates a read and data is not received in a specified interval (the slave does not pull the line low within this time), rather a time-out status bit is set. This indicates that a read was not successfully completed. On successful completion, the time-out bit is cleared. The bit remains set or cleared until the next transaction by the host.

An interrupt condition indicates either a TX-complete, RX-complete, or time-out condition. The read of the interrupt-status register clears all the interrupt conditions. Only one interrupt signal is sent to the MPU and only an overall mask bit exists for the enabling and disabling of the interrupt. Each of the interrupt conditions cannot be individually masked.

The following sequence must be performed by the programmer for the reads and writes to the slave:

Write operation:

1) Write the command or data value to the TX-write data register.

2) Write 0 to the R/W bit of the control-and-status register to indicate a write.

3) Write 1 to the Go bit of the control-and-status register to start the actual transmit. Steps 2 and 3 can be done at the same time.

   The following events occur after the Go bit is written:

   a) The hardware sends the byte from the TX data register.

   b) The time-out bit always is cleared in a write, because the hardware has no acknowledge mechanism from the slave.

   c) The completion of the operation sets the TX.INIT bit in the interrupt-status register. If interrupts are masked, no interrupt is generated. The interrupt-status register is always cleared at the beginning of any read or write operation.

   d) At the end of the write, the Go bit is cleared.

4) Software must read the interrupt-status register to clear the interrupt.

5) Repeat for each successive byte.

Read operation:

1) Write the command value to the TX-write data register.

2) Write 0 to R/W bit, 1 to the Go bit, and wait for the TX-complete interrupt.

3) Write 1 to the R/W bit of the control-and-status register to indicate a read.

4) Write 1 to the Go bit of the control-and-status register to start the actual read. Steps 3 and 4 can be done at the same time.

   The following events occur after the Go bit is written:

   a) The hardware detects afalling edge of the line (created by the slave) and receives 8 bits of data in the RX-receive-buffer register. The first bit that is received from the slave is the LSB and the last bit is the MSB of the byte. The master performs this step as soon as the slave sends the data irrespective of the state of the Go bit. However, an RX-complete interrupt is generated only when the Go bit is written by the software.

   b) If a time-out occurs, a time-out bit is set in the control-and-status register.

   c) The completion of the operation sets the RX.INIT bit in the interrupt-status register. If interrupts are masked, no interrupt is generated. The interrupt-status register is always cleared at the beginning of either a read or a write operation.

   d) At the end of the read, the Go bit is cleared. It is also cleared if a time-out is detected.

5) Software must read the interrupt-status register, to determine if the RX was complete or whether there was a time-out.

6) Software does a read of the RX buffer register to retrieve the read data from the slave.

7) Repeat for each successive byte.

In the HDQ mode, the address/command is only written once to the slave. However, after the first byte is received, if an RX-complete interrupt is received, the software must initiate the read of the second byte by writing the Go bit of the control-and-status register. The first byte that was received is shadowed and provided to the software, while the hardware is fetching the second byte of data.

2.1.1.2 1-Wire Mode

This section highlights the primary differences between the HDQ and the 1-Wire protocols.

In the 1-Wire mode, the firmware must send an initialization pulse to all of the slaves connected on the interface. All slaves respond with a presence pulse.

The initialization pulse is sent by setting the init bit and the Go bit in the control-and-status register. A presence detect is indicated in the appropriate bit of the register. If no presence pulse is received, then a time-out bit is set in the control-and-status register. The initialization bit is cleared at the end of the initialization pulse. Also, the presence detect and the time-out bits are cleared at the end of the initialization pulse, if a presence detect is received. The time-out bit has no other significance in this mode; that is, unlike in HDQ mode, it is always cleared during a read operation.

1-Wire mode is a bit-by-bit protocol for a read. Unlike HDQ, which sends eight bits of data on a read, the slave must be clocked by the host in 1-Wire protocol for each bit. At the end of the command/address byte, the line is pulled high and the host creates a low-going edge to initiate a bit read from the slave. The host then pulls the line high, and the slave either pulls the line low to indicate a 0 or does not drive the line to indicate a 1. The host repeats the operation for each bit read.

The first bit that is received is the LSB and the last bit is the MSB in the RX data register.

An interrupt condition indicates either a TX-complete, RX-complete, or time-out condition. The read of the interrupt-status register clears all the interrupt conditions. Only one interrupt signal is sent to the MPU and only an overall mask bit exists for the enabling and disabling of the interrupt. Each of the interrupt conditions cannot be individually masked.

The following sequence must be performed by the programmer for the reads and writes to the slave:

Write operation:

1) Write the ID, command, or data value to the TX-write data register.

2) Write 0 to the R/W bit of the control-and-status register to indicate a write.

3) Write 1 to the Go bit of the control-and-status register to start the actual transmit. Steps 2 and 3 can be done at the same time.

   The following events occur after the Go bit is written:

   a) The hardware sends the one byte of the TX-write-data register.

   b) The time-out bit is always cleared in a write.

   c) The completion of the operation sets the TX.INIT bit in the interrupt-status register. If interrupts are masked, no interrupt is generated. The interrupt-status register is always cleared at the beginning of any read or write operation.

   d) At the end of the write, the Go bit is cleared.

4) If an interrupt is enabled, software must read the interrupt-status register to clear the interrupt.

5) Repeat for each successive byte.

Read operation:

1) Write the ID value to the TX-write data register.

2) Write 0 to R/W bit and 1 to the Go bit and wait for TX-complete interrupt.

3) Write the command value to the TX-write register.

4) Write 0 to R/W bit and 1 to the Go bit and wait for TX-complete interrupt.

5) Write 1 to the R/W bit of the control-and-status register to indicate a read.

6) Write 1 to the Go bit of the control-and-status register to start the actual read. Steps 5 and 6 can be done at the same time.

   The following events occur after the Go bit is written:

   a) The hardware creates a falling edge of the line (created by the slave), and clocks 8 bits of data into the RX-receive-buffer register. The first bit that is received from the slave is the LSB and the last bit is the MSB of the byte.

   b) The time-out (PDT0-INT) bit is always cleared in a read.

   c) The completion of the operation sets the RX_INT bit in the interrupt-status register. If interrupts are masked, no interrupt is generated. The interrupt-status register is always cleared at the beginning of any read or write operation.

   d) At the end of the read, the Go bit is cleared. It is also cleared if a time-out is detected.

7) If interrupt is enabled, software must read the interrupt-status register to determine if RX was completed or whether there was a time-out.

8) Software does a read of the RX buffer register to retrieve the read data from the slave.

9) Repeat for each successive byte.

2.1.1.3 1-Wire Bit Mode Operation

A single-bit mode can be entered by writing to the SB bit in the control-and-status register. In this mode, only one bit of data is received each time from the slave. After the bit is received, an RX-complete interrupt is generated. Bit 0 of the receive buffer is updated each time a bit is received.

The single-bit mode has no effect in the HDQ mode, as HDQ does not support the single-bit protocol.

2.1.2 Timing Diagrams

Figure 2 through Figure 4 show the timing diagram for the read, reset, and write. In the HDQ mode , the reset pulse only contains the initialization and not the presence pulse. The timing required for the various signals are specified in the *BQ2023*, *Single-Wire Advanced Battery Monitor IC for Cellular and PDA Applications* (literature number SLUS480).

The master works at the timing of the HDQ interface, which encompasses the HDQ and the 1-Wire timing. Therefore, in the 1-Wire mode, the master runs slower than the full performance capability of the protocol.

Figure 2. Read-Timing Diagram

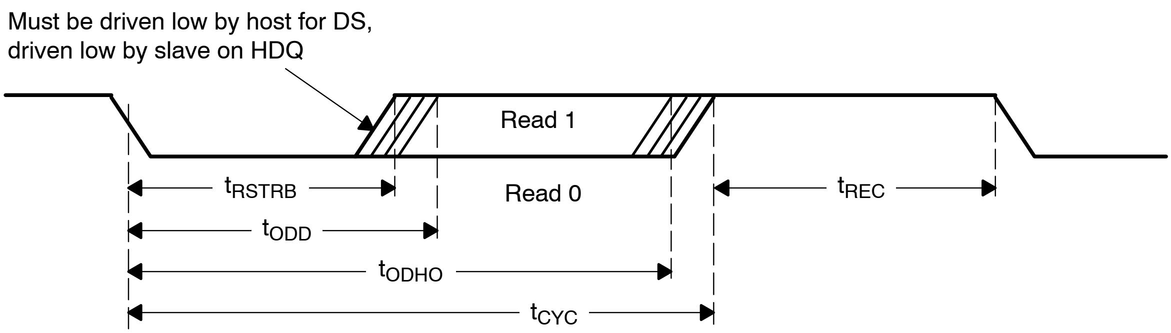


Figure 3. Reset-Timing Diagram

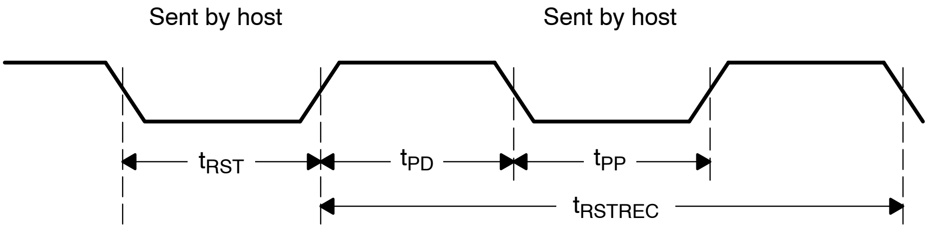
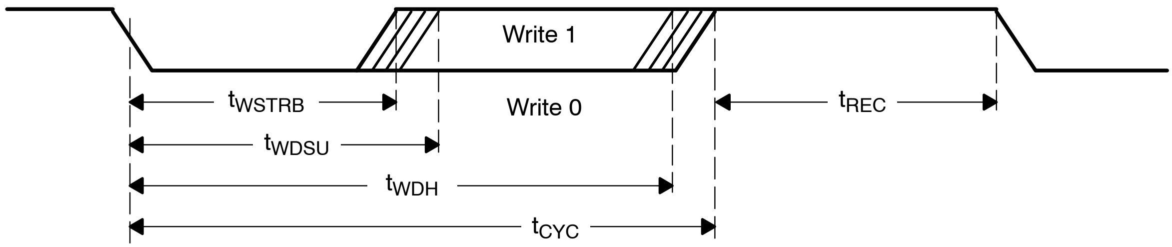
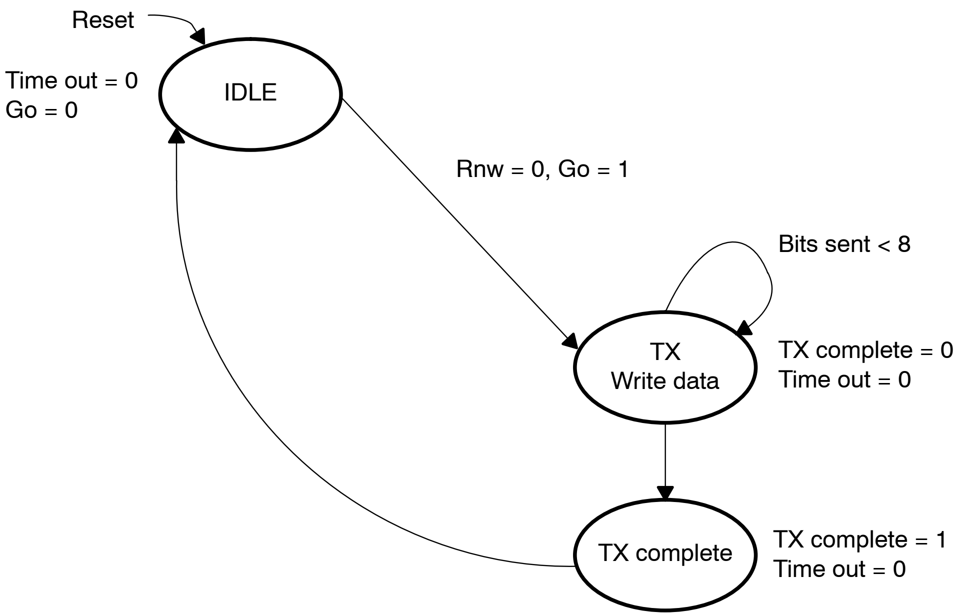


Figure 4. Write-Timing Diagram



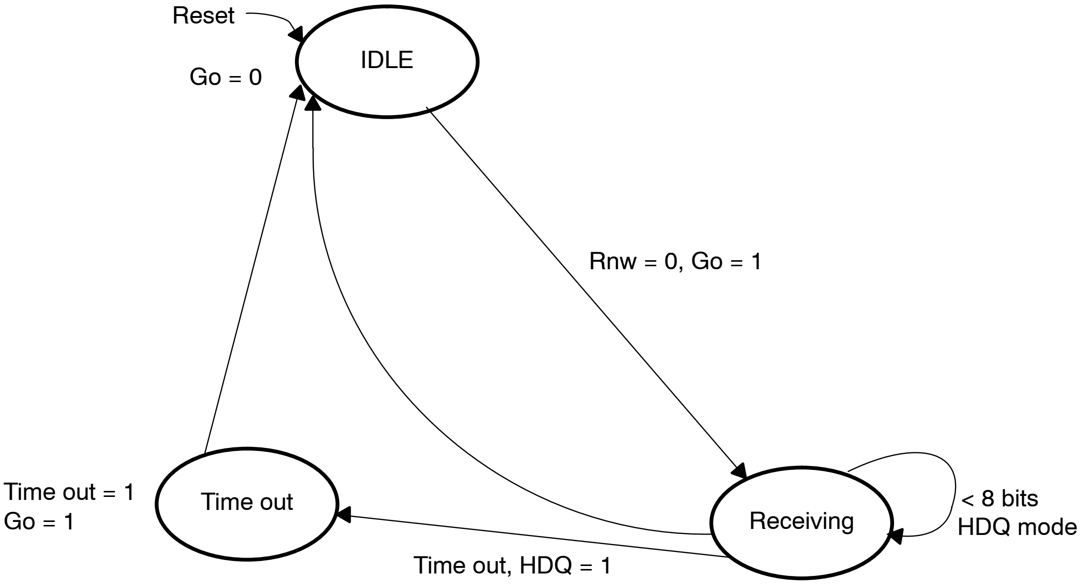
2.1.3 Write-State Diagram

Figure 5. Write-State Machine #1



2.1.4 Read-State Diagram

Figure 6. Read-State Machine #1



2.1.5 Status Flags

The control-and-status register contains status flags from the transmitter (TX_INT), the receiver (RXINT), and the presence detect /time-out (PDT0_INT) logic.

The presence condition detected status flag is contained in the control-and-status register. This is valid only in the 1-Wire mode. It is cleared when the host sends an initialization pulse and then is set to 1 if a pulse is received; otherwise it stays cleared at 0.

2.1.6 Interrupts

The following interrupt status is provided by the module:

- Transmitter complete (TX_INT)

  A write of one byte was completed. Successful or unsuccessful completion is not indicated, because there is no acknowledge from the slave in either the HDQ or 1-Wire modes. Cleared at the beginning of the write command.

- Read complete (RX_INT)

  Indicates successful completion of a byte read in both modes. Cleared at the beginning of each read command.

- Presence detect/time-out (PDT0_INT)
  - In the 1-Wire mode, it indicates that it is now valid to check the presence detect received bit. Cleared at the beginning of each initialization sequence.
  - In the HDQ mode, it indicates that after a read command was issued by the host, the slave did not pull the line low within the specified time. In the HDQ mode, the bit is cleared at beginning of each read command.

Only one interrupt is generated to the MPU, based on any of the above interrupt-status conditions. A read to the interrupt-status register clears all the status bits that have been set.

The interrupt can be masked by setting the INTEN bit to ’0’ in the control-and-status register.

A read of the interrupt-status register clears the interrupt. If there is a pending interrupt the interrupt line stays low and no low-high-low transition is created. The interrupt therefore must be handled as a level interrupt (where a low-going edge is not needed) in an upstream interrupt handler (or processor).

2.2 Power-Down Mode

Writing a ’0’ to the PD bit in the control-and-status register shuts off the clock to the state machines. The state machines are reset when the clock is disabled, and if any transaction is being performed it is aborted into the reset state. The register values are not affected by disabling the clock. Register accesses are illegal after the software puts the module in power-down mode, other than a write of a ’1’ to the PD bit to take the HDQ/1-Wire peripheral out of power-down mode.

Index