I^2C Battery Gauge Interface Using MAX II CPLDs

December 2007, ver. 1.0 Application Note 493

Introduction

Rechargeable batteries are used in a wide number of applications today. However, their usefulness is sometimes hampered by the fact that they cannot communicate with the user. This design example demonstrates how to use a MAX® II device as an I^2C battery gauge interface. The end product is a low-cost intelligent device that improves communication between the battery gauge and the host system.

Battery Gauge on the I^2C Bus

A battery gauge is a device which continuously monitors the state of a battery's capacity and displays it to the user in a simple form. Often in embedded systems, the status of the battery supply is monitored remotely. Communication between the embedded system and battery gauge can be facilitated through an industry standard two-wire I^2C bus and an interfacing device. Because of its low overhead, the two-wire communication link minimizes the number of traces required on the PCB to monitor the power source. A MAX II CPLD can easily serve as the interfacing device. The I^2C interface allows the MAX II CPLD to access readings from the battery gauge remotely. An I^2C serial interface consists of a Data line (SDA) and a Clock line (SCL). Both the lines are bidirectional and are pulled up to the V_{CC} of the system. This bus is also a common bus for any other I^2C-compliant device in the system.

Using a MAX II Device to Interface with a I^2C Battery Gauge

In this implementation, the battery gauge acts as slave to the MAX II CPLD and has a software programmable address. At any time, the CPLD acting as the master can read the exact state of battery charge through the battery gauge interface. The design uses an inexpensive battery gauge device, such as the Maxim DS2745. Information about the Maxim DS2745 is available at:
www.maxim-ic.com/quick_view2.cfm/qv_pk/4994

Because the design is made for illustration, it does not implement a complete I^2C master. This design, therefore, does not accommodate multimaster capability and is only meant to demonstrate the capability of the MAX II CPLD to read the DS2745 battery gauge via an I^2C interface.

Figure 1 shows the block diagram of the I^2C battery gauge. The bidirectional lines SCL and SDA form the input and output of the system. Table 1 shows I^2C signals that are used in the design.

Figure 1. I²C Interface between Battery Gauge and CPLD

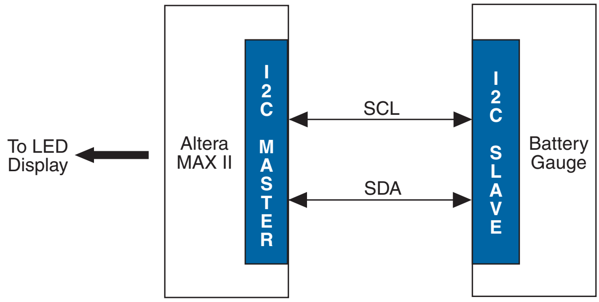


Table 1. I²C Interface Pin Description

Signal	Purpose	Direction
SCL	I²C Clock	Output
SDA	I²C Serial Data	Bidirectional

In order to retrieve the status of the battery from the gauge, the master initiates the communication cycle by sending a Start condition on the bus. The Start condition, which consists of pulling the SDA line low while SCL is high, is followed by a 7-bit slave address and a write command (90h). After receiving these signals, the slave sends back an acknowledgment to the master.

Figure 2. Master and Slave Communications

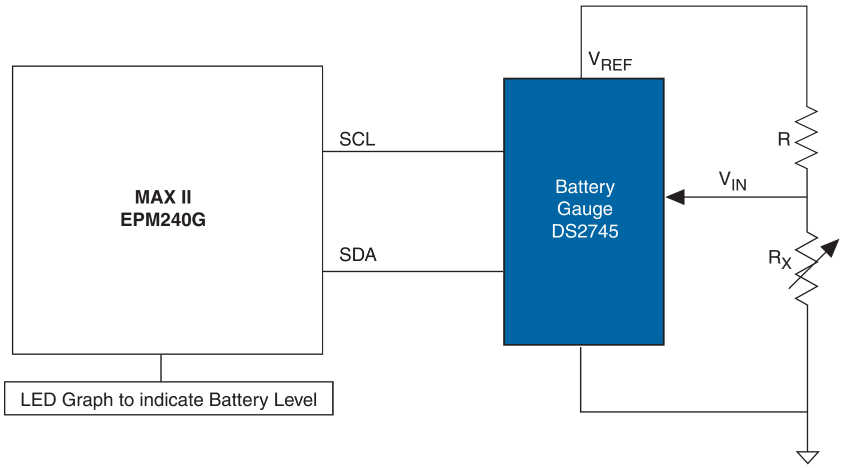
S	Device Address (90h)	ACK	Memory Address (0h)	ACK	Sr	Device Address (91h)	Gauge Output (MSB)	ACK	Gauge Output (LSB)	ACK	P
1-bit	8-bit Address with write	1-bit	8-bit	1-bit	1-bit	8-bit Address with write	8-bit	1-bit	8-bit	1-bit	1-bit

S: Start
P: Stop
ACK: Acknowledgement
Sr: Repeat Start

Master Write   Slave Write

After receiving the acknowledgment (battery gauge DS2745), the master sends the memory address (0Ch), which corresponds to the voltage register on the slave. The DS2745 acknowledges this memory address, and then the master generates a Repeat Start (Sr). After the Repeat Start, the CPLD (master) sends the device address again, followed by a read, this time (91h). The DS2745 acknowledges it as the same. The battery data is transferred on the SDA line to the master in the next two 8-bit read operations; the MSB is sent first followed by the LSBs. The master sends an acknowledgment after every 8 bits that are received. A Stop (P) condition is generated by the master by leaving the SDA line high when SCL is high after the read is complete. The slave returns Data in two's complement form, which is converted to a bar form for display.

Figure 3. I^2C Battery Gauge Demonstration Scheme



Implementation

This design example can be implemented with an EPM240 device or any other MAX II CPLD and a I^2C battery gauge device such as the Maxim DS2745. This design communicates with the Maxim DS2745 battery gauge through the I^2C bus and displays the constantly updated voltage readings without requiring any external stimulus.

The voltage status returned by the DS2745 battery gauge is 11-bit wide and is in 2's complement form. The voltage readings are converted into a viewer friendly form with eight voltage magnitude levels by the MAX II CPLD. These are subsequently displayed with the help of an LED array with eight LEDs.

The following details the implementation of this design example on the MDN-B2 demo board. Table 2 lists the EPM240G pin assignments for this design example.

Table 2. Design Example Using the MDN-B2 Demo Board

EPM240G Pin Assignments			
Signal	**Pin**	**Signal**	**Pin**
SCLK	pin 39	SDA	pin 40
led_level[0]	pin76	led_level[1]	pin 75
led_level[2]	pin 74	led_level[3]	pin 73
led_level[4]	pin 72	led_level[5]	pin 71
led_level[6]	pin 70	led_level[7]	pin 69

Assign unused pins as **input-tristated** in the Quartus® II software. You must also enable the **Auto Open Drain** setting on the SCLK and SDA pins. To do this, on the Assignments menu, click **Settings** and then select **Analysis and Synthesis Settings** to enable the **Auto Open-Drain** setting. These settings are followed by a compilation cycle.

Design Notes

To demonstrates this design on the MDN-B2 Demo Board, complete the following:

1. Turn on the power to the demo board (using slide switch SW1)

2. Download the design on to the MAX II CPLD through the JTAG header JP5 on the demo board and a conventional programming cable (ByteBlaster™ II or USB-Blaster™). Keep SW4 on the demo board pressed while starting the programming process. After programming, turn off the power and remove the JTAG connector.

3. Mount the DS2745 battery gauge module (supplied with the MDN-B2 demo board) on the header JP3 of the demo board. Make sure that the red mark on the connector meets pin #1 on JP3.

4. Turn on the power to the MDN-B2 demo board.

5. Observe that the red LED array on the demo board vary with changing voltages on the battery gauge module. This voltage can be varied by changing the preset on the battery gauge module.

Source Code

This design example has been implemented in Verilog and successful operation has been demonstrated using the MDN-B2 demo board, as described in this document. The source code, testbench, and complete Quartus II project are available at:

www.altera.com/literature/an/an493_design_example.zip

Conclusion

As illustrated in this design example, MAX II CPLDs provide cost-effective and versatile solutions for interfacing industry standard system interfaces such as I^2C. Their low power enables them to provide power-efficient solutions in battery operated systems. Their internal oscillator, user flash memory, and multi-volt features enable them to provide cost and board-space effective solutions, which are of paramount significance in most of today's battery-operated systems.

Additional Resources

- MAX II CPLD Homepage: www.altera.com/products/devices/cpld/max2/mx2-index.jsp
- MAX II Device Literature: www.altera.com/literature/lit-max2.jsp
- MAX II Power-Down Designs: www.altera.com/support/examples/max/exm-power-down.html
- MAX II Application Notes:
  AN 422: Power Management in Portable Systems Using MAX II CPLDs
  AN 428: MAX II CPLD Design Guidelines

Document Revision History

Table 3 shows the revision history for this application note.

Table 3. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
December 2007 v1.0	Initial release.	—

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support:
www.altera.com/support/
Literature Services:
literature@altera.com

I.S. EN ISO 9001