INTRODUCTION

The Inter-Integrated Circuit (I²C®) bus is a two-wire bidirectional bus used for low speed, short-distance communication between integrated circuits. Developed by Philips¹ in the early 1980s for use amongst ICs on a single board, I²C today is increasingly being used in multiboard applications as new bus extensions and control devices help overcome the original 400 pF maximum allowable load capacitance.

Often, in multocard applications such as blade servers or digitally-controlled power converters, it is desired that each interface be isolated to allow for trouble-free card insertion/removal or for safety considerations. However, isolating I²C interfaces is complicated by the bidirectional nature of the I²C bus. This characteristic is not compatible with the unidirectional behavior of optocouplers.

This application note provides a brief overview of the I²C bus (focusing on its physical layer), discusses the challenges in isolating I²C interfaces, and describes iCoupler® solutions for isolating I²C interfaces.

I²C OVERVIEW

The I²C interface is defined by The I²C-Bus Specification, Version 2.1, January 2000 (NXP Semiconductors). This interface consists of two wires: serial data (SDA) and serial clock (SCL). These wires convey information to and from devices connected to the bus, each of which is identified by a unique address. At any instance, a device can be a transmitter or a receiver although some devices may operate only as one or the other. In addition, at any instance, a device can be a master or a slave. A master is a device that initiates a data transfer by addressing another device while a slave is a device that is addressed by a master.

The I²C bus allows for more than two devices to be connected to it and for multiple master/slave relationships to exist. The operation of the bus under such circumstances is defined by an arbitration procedure defined in the I²C standard.

Note that the master/slave designations are independent of whether a device is a transmitter or a receiver. For example, in a sequence in which a master initiates a data transfer from a slave, it is first a transmitter (as it addresses the slave), then a receiver (as it receives the data from the slave), and then a transmitter again (as it terminates the transfer). In a similar fashion, the slave is first a receiver, then a transmitter, and then a receiver.

The I²C bus operates on the principle of open-drain/open-collector wire-AND functionality (see Figure 1). All devices connected to the bus must be at a logic high state in order for the bus to be at a logic high state. When this situation exists for both the SDA and SCL lines, the bus is considered to be free for a device to initiate a data transfer. Both SDA and SCL are bidirectional lines to support the ability of devices to take on both transmitter and receiver roles.

¹ In 2006, Philips spun out their semiconductor operations (including their I²C portfolio) to create a new independent company called NXP Semiconductors.
#TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>I²C Overview</td>
<td>1</td>
</tr>
<tr>
<td>The Challenge of Isolated I²C Interfaces</td>
<td>3</td>
</tr>
<tr>
<td>iCoupler I²C Isolation Solutions</td>
<td>4</td>
</tr>
<tr>
<td>ADuM1250/ADuM1251 Usage Notes</td>
<td>5</td>
</tr>
</tbody>
</table>
Table 1. I2C Logic Levels

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Min</th>
<th>Max</th>
<th>Unit</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic Low Input Voltage</td>
<td>$V_{IL}$</td>
<td>$-0.5$</td>
<td>$0.3 \times V_{DD}$</td>
<td>V</td>
<td>Standard mode allows for a fixed-input specification ($-0.5$ V min, $+1.5$ V max)</td>
</tr>
<tr>
<td>Logic High Input Voltage</td>
<td>$V_{IH}$</td>
<td>$0.7 \times V_{DD}$</td>
<td>$V_{DD} + 0.5$</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Logic Low Output Voltage (for 3 mA sink current)</td>
<td>$V_{OL,1}$</td>
<td>0</td>
<td>0.4</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>$V_{DD} &gt; 2$ V</td>
<td>$V_{OL,3}$</td>
<td>0</td>
<td>$0.2 \times V_{DD}$</td>
<td>V</td>
<td>Standard mode does not allow for $V_{DD} &lt; 2$ V</td>
</tr>
<tr>
<td>$V_{DD} &lt; 2$ V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data transfer can occur at up to either 100 kbps (standard mode), 400 kbps (fast mode), 1 Mbps (fast mode plus), or 3.4 Mbps (high speed mode). There is no limit to the number of devices that can be connected to the bus—as long as a 400 pF bus limit is not exceeded. The logic levels for I2C are shown in Table 1.

An important aspect of the I2C interface is that SDA logic transitions can only occur while the SCL clock signal is low. Also, the start and stop signals that bound the transmittal of data are SDA logic transitions that occur while the SCL clock signal is high. Therefore, it is important that the SCL signal be stable in both its low and high states to avoid communication problems on the bus.

THE CHALLENGE OF ISOLATED I2C INTERFACES

The bidirectional nature of the I2C interface presents special challenges in implementing isolation in a manner that avoids bus glitches or lock-up. Figure 2 shows a circuit based on optocoupler technology. Since optocouplers are inherently unidirectional devices, each bidirectional I2C line must be broken out into two unidirectional lines to support communication through the optocouplers. In Figure 2, only the SDA lines are shown for simplicity. Isolating a complete I2C interface requires four optocouplers. The resulting increase in cost, board space, and complexity diminishes the inherent value (that of providing a simple, low cost, 2-wire interface) of I2C.

An additional problem with a circuit such as this is that it results in undesirable bus glitches (see Figure 3).

- A high-to-low transition occurs at SDA. In this situation, $IC_1$ is turned on by the current flowing through $R_2$. This, in turn, pulls Node 1 low and consequently pulls SDA’ low via $D_2$.
- SDA’ is taken low. Nothing changes except $D_2$ no longer conducts.
- SDA is released and goes high as there is nothing holding it low. The LED in $IC_1$ turns off. After a delay, the transistors of $IC_1$ turn off. Node 1 goes high and turns on the LED in $IC_2$. After another delay, the transistors of $IC_2$ turn on and SDA is pulled low to the desired state.

Notice that SDA is high for the duration it takes for $IC_1$ to turn off and $IC_2$ to turn on. This is an unwanted glitch on the bus in which SDA is high while SDA’ is trying to bring the bus low.
A solution offered by NXP Semiconductors in the P82B96 data sheet consists of their dual bidirectional bus buffer and four optocouplers. This solution is free of glitches or lock-up but it still requires multiple components bringing board space and cost penalties that undermine the benefits of the I²C interface.

**iCOUPLER I²C ISOLATION SOLUTIONS**

In contrast, Analog Devices, Inc's ADuM1250/ADuM1251 iCoupler products are single-component I²C isolators free of any glitch or lock-up issues and without the size, cost, and complexity of optocoupler-based solutions. The ADuM1250 supports interfaces with bidirectional data and clock lines while the ADuM1251 supports interfaces with a bidirectional data line and a unidirectional clock line. Both products carry UL approved 2.5 kV rms isolation ratings and are offered in an 8-lead SOIC package.

---

**Figure 4. ADuM1250 Functional Block Diagram**

**Figure 5. ADuM1251 Functional Block Diagram**
An isolated I2C interface using the ADuM1250/ADuM1251 is extremely simple (see Figure 6). It consists of just one component and a bypass capacitor for each supply. The pull-up resistors are those associated with any I2C interface. Guidelines for the selection of resistor values are provided within the I2C specification.

The ADuM1250/ADuM1251 isolators support two bidirectional communication lines by internally configuring four unidirectional isolation channels in the wire-AND open-drain configuration of the I2C interface. Special logic voltage levels at Side 1 of the devices are used to avoid bus glitches or latch-up. This prevents a logic low asserted by the Side 1 receiver from being interpreted as being input low by the Side 1 transmitter, thereby breaking the loop between Side 1 and Side 2 (see Figure 7).

The logic low output from the Side 1 receiver is 0.9 V (maximum). This is low enough to be read as an input low by other standard CMOS devices' Side 1, but high enough to avoid being interpreted as a logic low by the Side 1 transmitter, which has its logic low threshold at 0.7 V (maximum). Therefore, an output low from the Side 1 receiver is properly detected by devices connected to the bus, but is not fed back to Side 2 by the Side 1 transmitter. This prevents the bus problems commonly associated with optocoupler solutions while still supporting clock frequencies up to 1 MHz. Because the feedback loop is broken at Side 1, there is no need to do the same at Side 2. On that side, standard logic voltage levels are used.

**ADuM1250/ADuM1251 USAGE NOTES**

The recommended method of using the ADuM1250/ADuM1251 is to connect Side 2 to the I2C bus. This side of these isolators is fully compliant with the I2C specification for standard and fast mode operation. Side 1 (while fully compatible with I2C devices) is not strictly compliant to the I2C specification due to the special logic low voltage levels used to prevent bus latch-up.
Figure 8 shows an I\(^2\)C bus with multiple devices connected via ADuM1250 isolators. Each device has its own power supply and is connect to Side 1 of an ADuM1250. Side 2 of each ADuM1250 is connected to the bus and powered from a common V\(_{DD2}\) supply.

In any I\(^2\)C interface, care should be taken to make sure that design is compliant with the timing requirements in the I\(^2\)C specification. Specifically, the propagation delays of the isolation channels as well as the propagation delay mismatches between isolation channels need to be considered to make sure the resultant interface meets I\(^2\)C requirements.

There are two timing parameters, which isolators in an I\(^2\)C interface can affect:
- SDA setup time (250 ns for fast mode)
- SDA hold time (300 ns for fast mode)
Two situations need to be considered when analyzing the impact of I2C isolators. The first is when the master device is writing to the slave (see Figure 9). In this situation, the channel-to-channel mismatch of the isolator can decrease the setup or hold time of the SDA signal as received by the slave. To protect against this possibility, the master setup and hold times should be increased by at least the amount of channel mismatch to ensure that required setup and hold times are met at the receiving slave device.

The second scenario is that when the slave device is writing to the master (see Figure 10). In this situation, the master provides the SCL clock to the slave, which in turn writes the SDA signal to the master. The master pulls the SCL low and must receive the SDA edge from the slave before the SCL returns to a logic high state (less the required setup time). This means that the round-trip propagation delay through the isolator plus the slave’s response time must be less than that of the SCL logic low duration less its setup time:

\[ t_0 + t_{\text{SCL}} + t_{\text{RESPONSE}} + t_{\text{SDA}} < T_{\text{LOW}} - T_{\text{SETUP}} \]

This, in turn, places a constraint on the response time of the slave.

\[ t_{\text{RESPONSE}} < T_{\text{LOW}} - T_{\text{SETUP}} - t_0 - t_{\text{SCL}} \]

In both scenarios, an ideal I2C isolator has short propagation delays and tight, channel-to-channel matching. In this regard (in addition to their size, cost, and simplicity benefits), the ADuM1250/ADuM1251 also offer superior performance characteristics because they impact the timing of I2C interfaces to a lesser degree.