

Using I²C in Systems with Slow Clock Edges

Author: Edward Grivna

Associated Part Family: CYxxx

This application note discusses how to ensure full compliance with the I²C-bus specification when the bus has high capacitive loading.

1 I²C Signaling Rates

Cypress devices with I²C ports are optimized to work in common electrical environments and bus conditions. They are typically configured as master or slave to communicate through the I²C-bus. The current version of the I²C-bus specification requires devices to comply with different waveform timings across multiple signaling rates:

- 100 kbps in Standard mode
- 400 kbps in Fast mode
- 1 Mbps in Fast-mode Plus

2 Possible I²C Issues

While Cypress devices are designed for compliance with the [I²C-bus specification](#), if your system operates at or near the extreme limits of the I²C specification, parts of the timing should be verified to ensure it performs correctly.

One issue is a system with a clock line (SCL) with fall time (t_f) close to the I²C maximum of 300 ns. The [I²C-bus specification](#) defines the data hold time parameter ($t_{HD,DAT}$) as shown in [Figure 1](#).

The key things that must be understood about $t_{HD,DAT}$ are:

- The parameter is defined as a property of signals on the *external* bus, not as a property of signals inside a device.
- The parameter is defined from the low-threshold end of the falling edge of SCL ($V_{IL} = 30\%$ of V_{DD}), to the start of the falling or rising edge of SDA (70% or 30% of V_{DD}).
- Any device on the bus must guarantee that $t_{HD,DAT} > 0$ ns across the entire range of allowable rise and fall times for SCL and SDA.

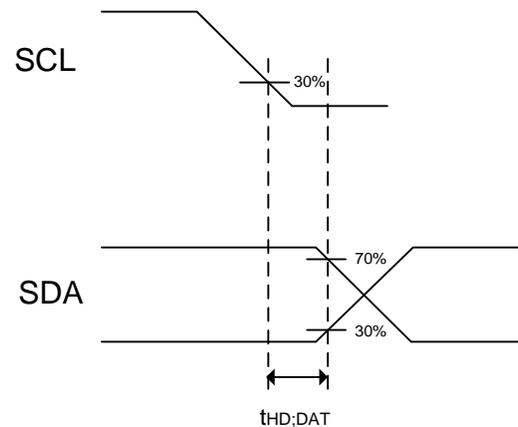
This final statement is important and the I²C-bus specification contains a footnote (on p.48) specifically addressing the topic:

“A device must internally provide a hold time of at least 300 ns for the SDA signal (with respect to the $V_{IH(min)}$ of the SCL signal) to bridge the undefined region of the falling edge of SCL.”

Let us clarify what this means and why it must be checked in your system to ensure there are no issues.

The footnote in the specification means you must guarantee that $t_{HD,DAT}$ is always greater than 0 ns when the SCL fall time (t_f) is near its maximum value. Furthermore, it requires the device that drives SDA to begin driving at least $t_{f(max)}$ after SCL began falling. Since $t_{f(max)}$ is 300 ns for Standard and Fast mode, the footnote quotes that particular time. A smaller time is required for Fast-mode Plus. The question you need to answer for your system is whether the $t_{HD,DAT} > 0$ ns condition is violated when SCL falls slowly.

Figure 1. I²C Data Hold Time ($t_{HD,DAT}$)



3 I²C Logic Levels and Timing Corner Cases

To answer this question, let us look at how the logic levels are defined in the I²C-bus specification. Per that specification, an external signal is at logic '1' when it is at or above 70% of the supply (V_{DD}) level, and logic '0' when it is at or below 30% of the supply level. These definitions leave the region between 30% and 70% undefined.

In reality, an input buffer switches from a '1' to a '0' somewhere in this undefined region which varies with signal slew rate and hysteresis in the buffer. This can create an issue in a very specific situation.

Consider that a slave responds (with acknowledge or data bits) on SDA to the master's SCL. If the SCL fall time is large (near maximum), it is possible that the slave recognizes SDA as being 'LOW' (logic-0) long before it reaches 30% of V_{DD} . If the SDA rise or fall time is very short, it can be asserted before SCL passes the 30% threshold and potentially corrupting the data transfer (see Figure 2).

Note that $t_{HD;DAT}$ is a bus overall timing and is determined by many factors such as input switching thresholds of the ICs on the bus, the number of devices on the bus, the capacitive loading of the devices, and the physical dimensions and length of the bus traces.

This condition is rare in most system for the following reasons:

- Portable devices often use Fast-mode Plus (1 Mbit/s) to save power, which has a minimum t_f of 120 ns.
- I²C-buses typically use a small number of ICs and short printed circuit traces. Therefore, the I²C bus is rarely loaded and the expected transition time t_f is often 25 ns or less.
- This situation can only occur when the I²C-bus is heavily loaded and there is a significant difference in drive strength between master (weak) and slave (strong).

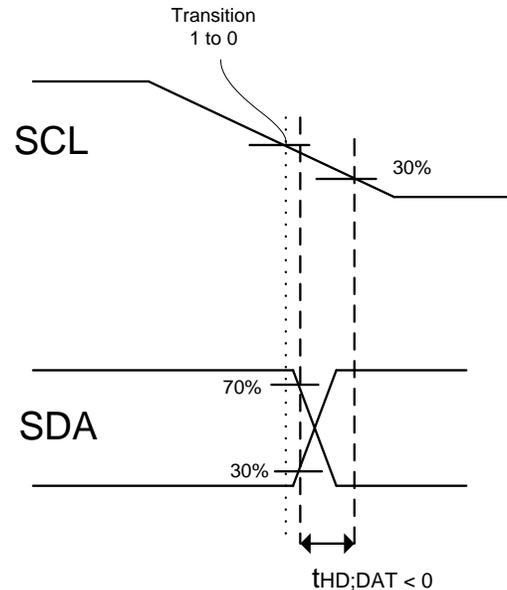
4 Solution

If your system uses I²C, it is good practice to measure the rise and fall times of SCL and SDA to determine if your system is in this corner condition. Measure the fall time of SCL from the master to ensure it is not close to the maximum allowed value. If the fall time is less than 50% of the specified limit, then there should be margin to avoid a violation of the $t_{HD;DAT} > 0$ ns condition. When measuring, be sure to follow established probing practices for fast edges to avoid getting erroneous results. If the SCL fall time is close to the allowed maximum, possible solutions are to

- increase the SDA rise and fall time by adding capacitance from the SDA line to ground
- Increase the SDA rise time by increasing the resistance of the buss pull-up resistor
- Increase the SDA fall time by changing the GPIO drive mode

When making any of these changes, also ensure that they do not negatively impact the timing compliance of the other devices on the I²C bus.

Figure 2. Data Corruption When $t_{HD;DAT} < 0$



Document History

Document Title: AN81514 - Using I2C in Systems with Slow Clock Edges

Document Number: 001-81514

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	3699247	MPU	08/03/2012	New application note
*A	5731154	ELG	05/09/2017	Converted to new template Changed from TrueTouch-specific to apply to any I ² C port

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at [Cypress Locations](#).

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6](#)

Cypress Developer Community

[Forums](#) | [WICED IOT Forums](#) | [Projects](#) | [Videos](#) | [Blogs](#) | [Training](#) | [Components](#)

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.



Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2012-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.