

Choosing the Correct I²C Device for New Designs

Siby James

ABSTRACT

The I²C bus is a very popular and powerful bus used for communication between a single (or multiple) master and a single (or multiple) slave device. In many applications there is a potential need for more slave devices on the bus, isolation between similar addressed slaves, or a need for more I/Os. These needs can be solved with an I²C buffer, switch, and I/O expander.

The application note helps users understand the use-cases of buffers and repeaters, switches, and I/O expanders and how to select the appropriate device for an application.

Contents

1	Introduction	2
2	I/O Expander Applications	3
	2.1 RESET Pins	4
	2.2 Internal Pullups.....	4
	2.3 Address Pins	4
	2.4 Level Shifting	4
3	I ² C Switches	5
	3.1 Control Register.....	7
	3.2 Hot Insertion Supported.....	7
	3.3 Address Pins	7
	3.4 Level Shifting	7
4	I ² C Buffers	8
	4.1 Pull-Up Resistors	9
5	I ² C Voltage Translation	11

List of Figures

1	I ² C Bus With Peripheral Devices Attached Example	2
2	Microcontroller or Processor Using I/O Expander to Control Peripherals	3
3	Selection Chart for TI I/O Expanders and their Key Features	5
4	I ² C bus with multiple slave devices with the same address.....	6
5	Eight-Channel I ² C Switch Example.....	6
6	Selection Chart for TI I ² C Switches and Their Key Features	8
7	Heavily-Loaded Bus Requiring I ² C Buffer	9
8	I ² C bus utilizing I ² C buffer, which allows more slave devices	9
9	Example of B-B side mismatch configuration for I ² C buffers	10
10	Selection Chart for TI I/O Expanders and Their Key Features.....	10
11	Operation of an I ² C Voltage Translator/Level Shifter	11

Trademarks

1 Introduction

The I²C bus is a standard bidirectional interface that uses a controller, known as the master, to communicate with slave devices. A slave may not transmit data unless it has been addressed by the master. Each device on the I²C bus has a specific device address to differentiate between other devices that are on the same I²C bus. Many slave devices require configuration upon startup to set the behavior of the device. This is typically done when the master accesses the internal register maps of the slave, which have unique register addresses. A device can have one or multiple registers where data is stored, written, or read. The physical I²C interface consists of the serial clock (SCL) and serial data (SDA) lines. Both SDA and SCL lines must be connected to VCC through a pull-up resistor. The size of the pull-up resistor is determined by the amount of capacitance on the I²C lines (for further details, refer to *I²C Pull-up Resistor Calculation* (SLVA689). Data transfer may be initiated only when the bus is idle. A bus is considered idle if both SDA and SCL lines are high after a STOP condition.

For more information on I²C bus operation, refer to *Understanding the I²C Bus* (SLVA704).

Figure 1 illustrates a typical I²C bus for an embedded system where multiple slave devices are used. The microcontroller represents the I²C master, and controls the IO expanders, various sensors, EEPROM, ADCs/DACs, and much more, all of which are controlled with only two pins from the master.

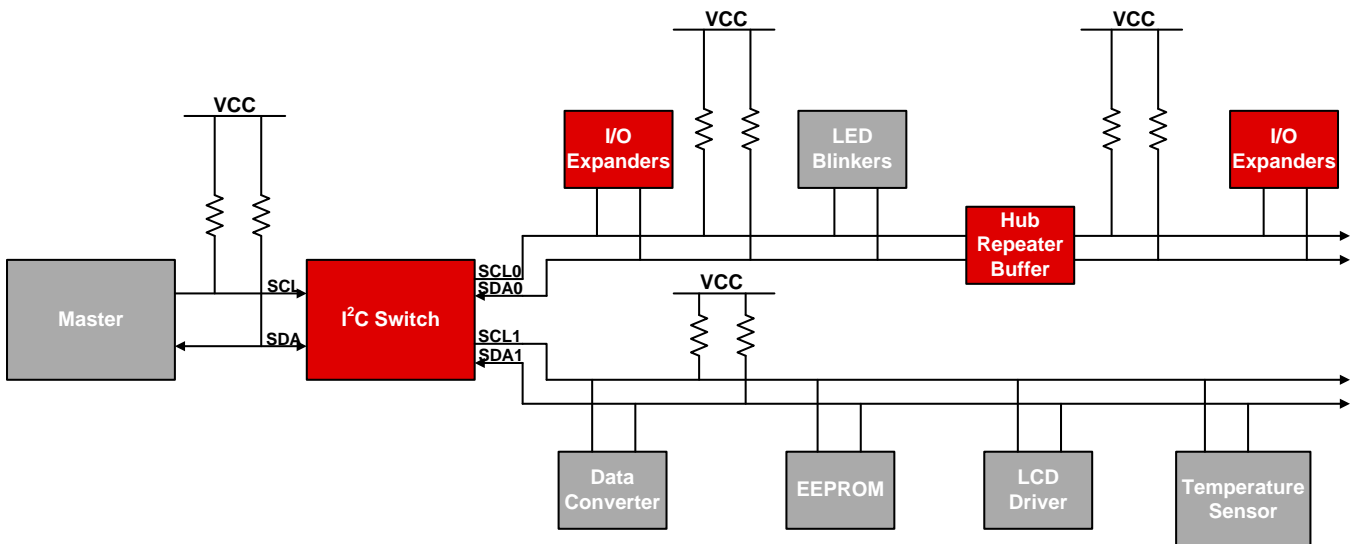
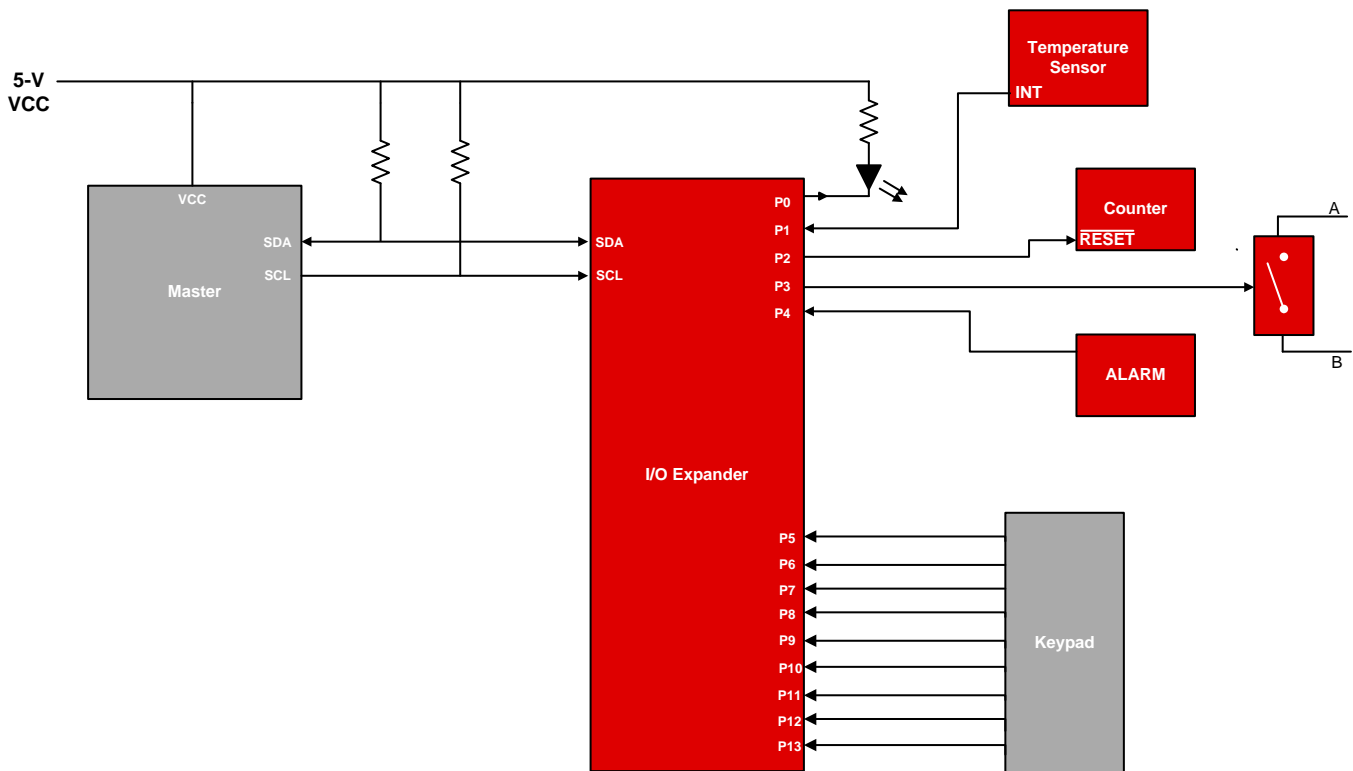


Figure 1. I²C Bus With Peripheral Devices Attached Example

2 I/O Expander Applications

I²C I/O expanders are devices used when there is a lack of inputs and outputs on a processor or controller. TI offers a wide variety of I²C-controlled I/O expanders ranging from 4-bit devices to 24-bit devices.

A common issue in applications involving microcontrollers and microprocessors involves the need for additional inputs and outputs. As inputs and outputs are needed more in applications to enable devices, monitor interrupts, and toggle resets, I/Os are beginning to be seen as a premium in master devices. However, with the development of I/O expanders, TI is creating the potential for cost-savings in applications that do not need higher end processors with extra GPIOs. For example, an application using the MSP430G22230, a microcontroller with 4 available GPIOs, could be paired with the TCA9555 GPIO expander in order to have 16 extra GPIOs available to support all application requirements.



NOTE: Arrows distinguish if I/O is an input or output.

Figure 2. Microcontroller or Processor Using I/O Expander to Control Peripherals

The use of I/O expanders in an application can range from controlling or enabling switches, enabling other devices within a system (that is, DC/DC with an EN pin), resetting other devices via RESET pin, or even monitoring status outputs (see Figure 2). TI offers a wide variety of I/O expanders in its portfolio; however, narrowing down the I/O expander that best fits an application need can be confusing, based on the features that each I/O expander has.

Designing systems with I/O expanders allows for more control when there is a lack of input and output pins provided on a processor. TI offers I/O expanders with added features, such as extra address pins for several unique slave addresses, reset pins, and internal pull-ups.

2.1 RESET Pins

A common fix for slave devices on the I²C bus that are experiencing faults is to power-cycle their power supplies to reset them. Power-cycling the power supply for the I/O expander can negatively affect other devices that are shared on a power rail. For example, it is possible for an I/O expander to share a power rail with an ambient light sensor that must be powered for the duration of device operation. Some I/O expanders feature a RESET pin, which offers the ability to manually reset the device using a microcontroller or microprocessor to toggle the reset input. The RESET pin functionality allows the power supply to remain powered while resetting the I/O expander.

2.2 Internal Pullups

Certain devices within the I/O expander portfolio include internal pull-up resistors as an added feature. Internal pull-up resistors have the benefit of saving both board space and placement costs for external resistors. Internal pull-ups are more useful for applications where several or all of the ports are set as an input and require pull-up resistors. Since the pull-ups are connected to VCC, increased power consumption will occur when the inputs are held at a low state.

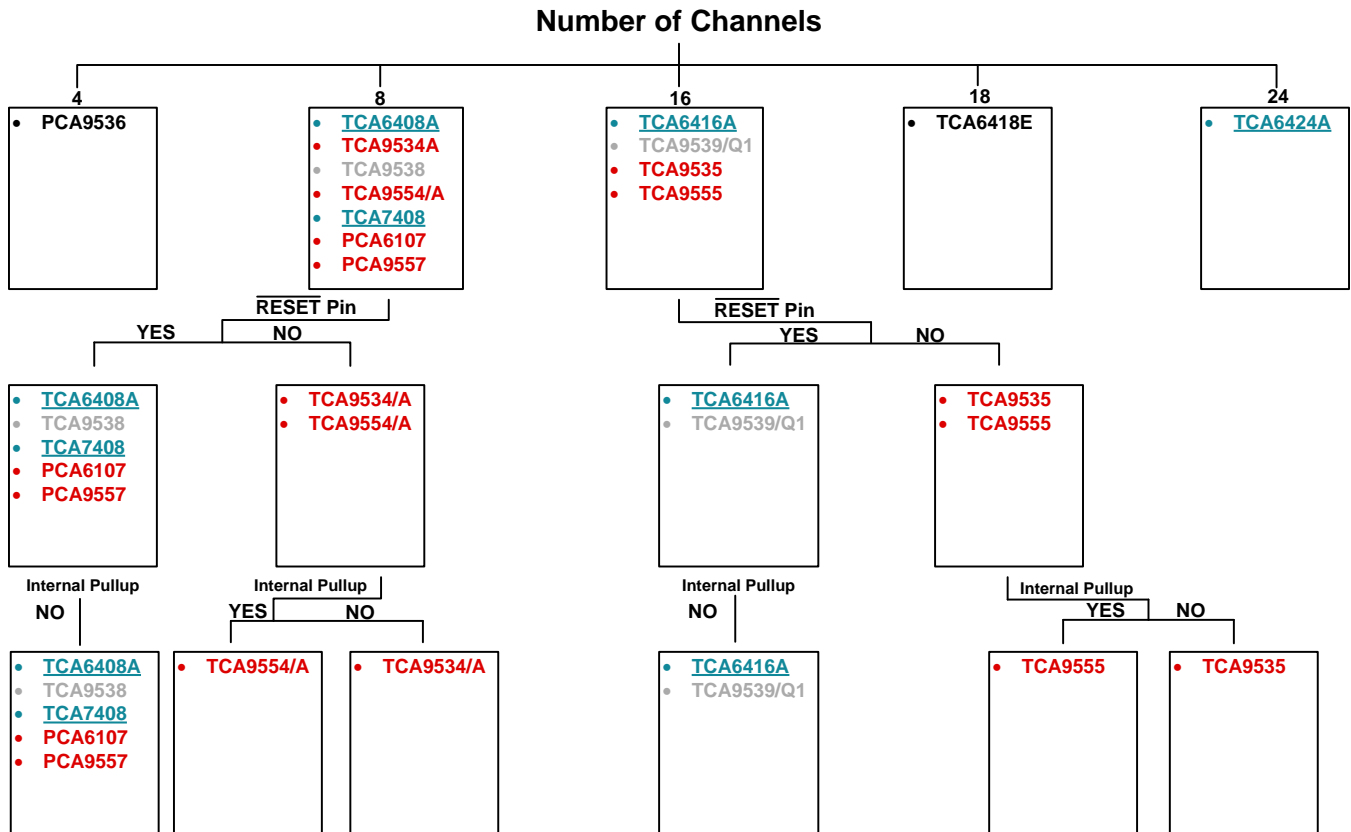
2.3 Address Pins

To ensure successful communication on the I²C bus, each slave device must have a unique address to prevent address conflicts. Most of the devices in the I/O expander portfolio have 1–3 address pins which offer 2–8 unique addresses for each device. Ensuring each slave has a unique address prevents address conflicts, which can result in unexpected behavior or corrupted data during communication.

2.4 Level Shifting

Some devices in the I/O expander family feature level shifting/voltage translation, which allows the I²C bus voltage to be at a different voltage level than the GPIO ports. A common example of I/O level shifting is for situations where the output of the I/O expander must be at 5 V to drive logic devices or LEDs, but the I²C bus is at 3.3 V.

When choosing from the I/O expander portfolio, the selection chart shown in [Figure 3](#) allows quick selection of the appropriate device, based on specific parameters/features. For example, an I/O expander with 8 channels, no RESET pin, and internal pull-up resistors would narrow the search down to TCA9554/A.



Note: Underlined devices indicate level-shifting ability.

- No Address Pin
- 1 Address Pin
- 2 Address Pins
- 3 Address Pins

Figure 3. Selection Chart for TI I/O Expanders and their Key Features

3 I²C Switches

I²C switches are slave devices controlled by the I²C bus that fans out multiple I²C channels to enable more control over your bus.

Applications with slave devices that share the same address can use an I²C switch to prevent slave address conflicts. By having the same slave address, the I²C master will be unable to tell which slave device it is speaking to and most importantly, may communicate commands to the wrong slave device. A solution is to incorporate an I²C switch (Figure 4 and Figure 5). An I²C switch has the capability of isolating slave devices with the same addresses to prevent corruption.

Switches are also used to isolate portions of an I²C bus which have legacy slave devices that cannot handle higher speed bus transactions. Also switches can be used to turn off portions of the bus which are not powered constantly, or are shut down temporarily, to save power.

A common misconception about I²C switches is that they are often confused as a multiplexer. The difference between a switch and multiplexer is that a multiplexer utilizes a selection bit to enable a specific channel while a switch can enable one or many channels.

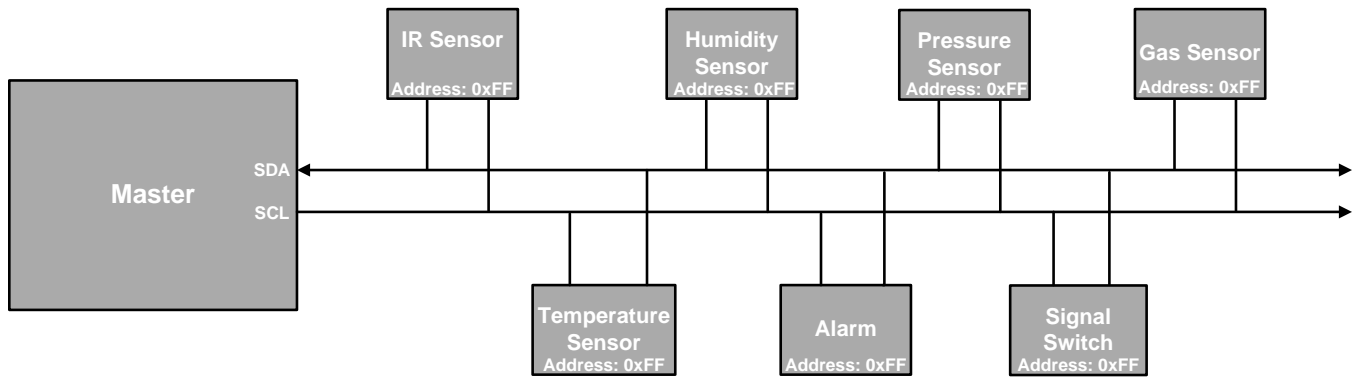


Figure 4. I²C bus with multiple slave devices with the same address

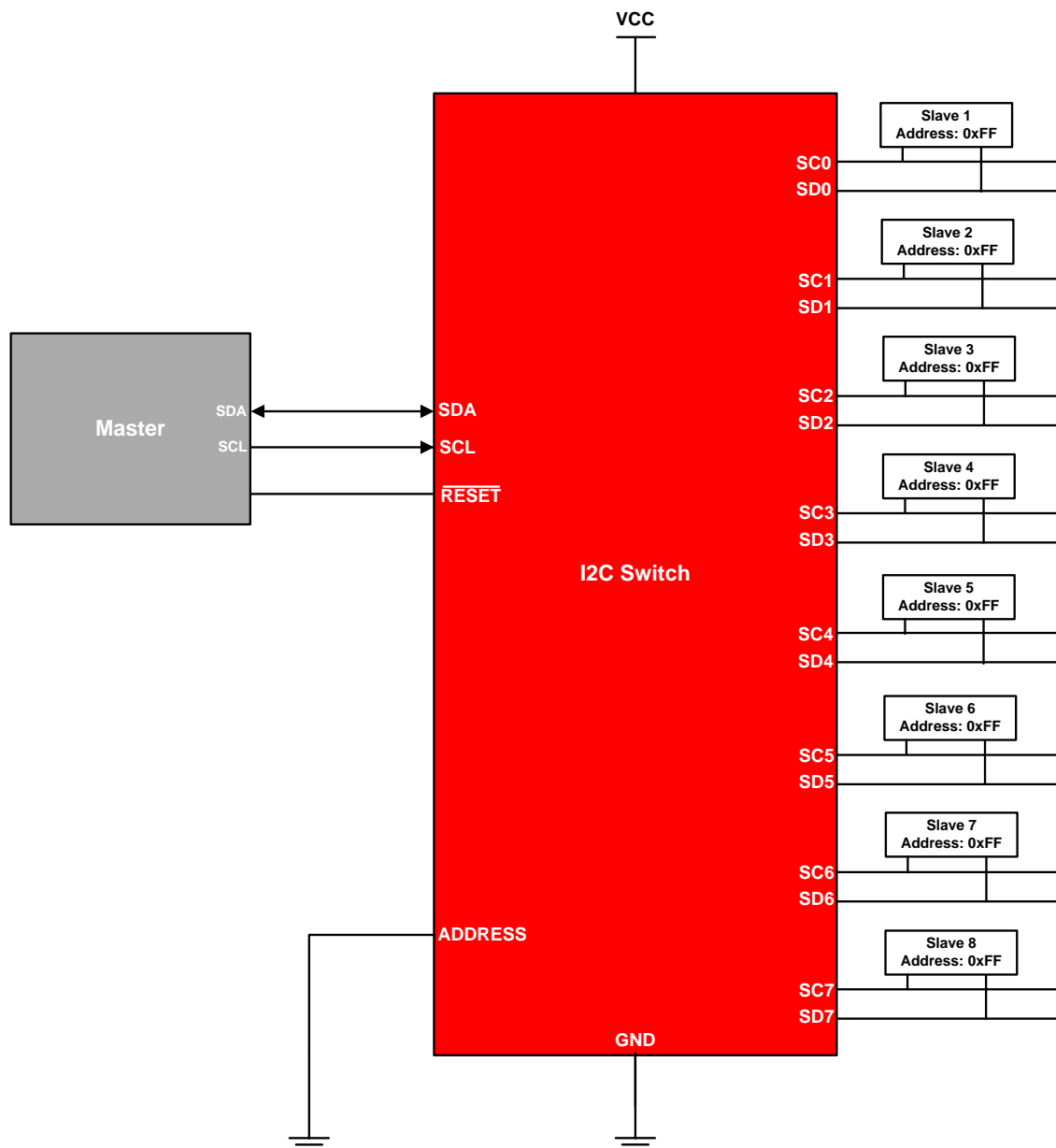


Figure 5. Eight-Channel I²C Switch Example

3.1 Control Register

I²C switches determine which channels are connected internally through their control register. Following the successful acknowledgement of the address byte, the bus master sends a command byte that is stored in the control register of the I²C switch. This register can be written and read via the I²C bus. Each bit in the command byte corresponds to an SC_n or SD_n (SC_n/SD_n) channel and a high (or 1) selects this channel. Multiple SC_n/SD_n channels may be selected at the same time. When a channel is selected, the channel becomes active after a stop condition has been placed on the I²C bus. This ensures that all SC_n/SD_n lines are in a high state when the channel is made active, so that no false conditions are generated at the time of connection. A stop condition always must occur immediately after the acknowledge cycle. If multiple bytes are received by the I²C switch, it saves the last byte received

3.2 Hot Insertion Supported

A feature that is supported by all I²C switches by TI is *Hot Insertion*. The I²C switches can be added into a system without connecting any of the downstream channels during power up. For example, the 8-channel TCA9548A will not connect pins SD/SC[0–7] when hot inserting the device until explicitly commanded to by the I²C master. This means that when the TCA9548A is inserted into an active bus, it will not connect any of the channels, eliminating the chance of a device on those channels interfering with the main SDA/SCL line since the I²C master will have to command the TCA9548A to connect the SD2 and SC2 lines.

3.3 Address Pins

To ensure successful communication on the I²C bus, each slave device must have a unique address to prevent address conflicts. Most of the devices in the switch portfolio have 1–3 address pins which offer 2–8 unique addresses for each device. Ensuring each slave has a unique address helps prevent address conflicts, which can result in unexpected behavior or corrupted data during communication.

3.4 Level Shifting

I²C switches allow level shifting between the master side and slave side to ensure compatibility between I²C master and I²C slave as well as between multiple slave devices operating at different voltages. Mismatched voltages are a concern with voltages of master and slave devices moving towards lower voltages. For example, it is possible for a 1.8-V master I²C bus to communicate with a 3.3-V slave device on another channel. All channels may be at different voltages, with the only requirement being that the VCC pin must be connected to the lowest bus supply voltage that the switch will see.

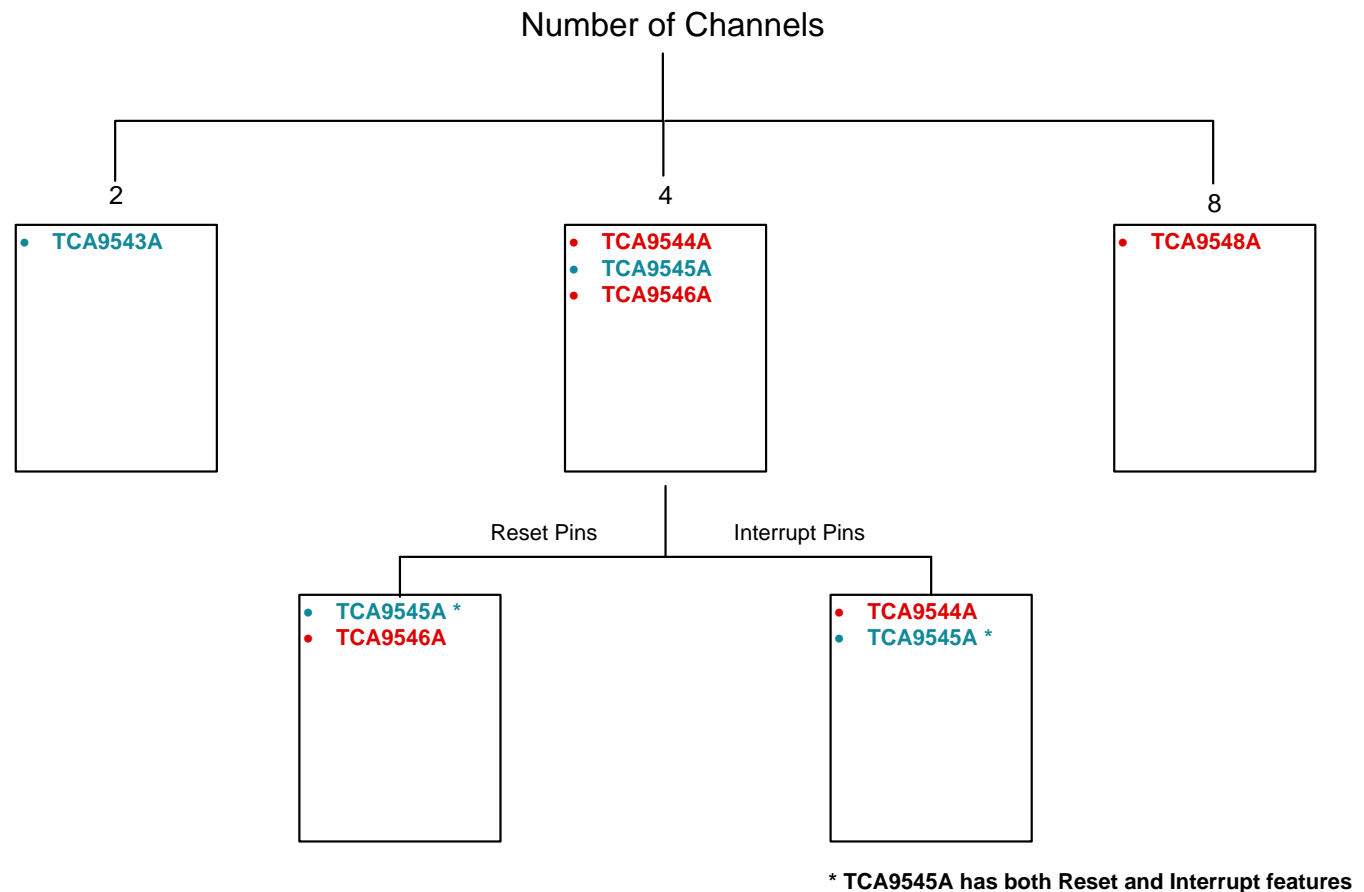


Figure 6. Selection Chart for TI I²C Switches and Their Key Features

4 I²C Buffers

In compliance with the I²C specification, the maximum capacitive load of the bus is limited to 400 pF. Once the capacitance is exceeded, the rise times on the I²C bus may violate the timing requirements. With an I²C buffer and repeater (buffer/repeater), slave device's capacitances can be isolated on each side of the buffer to have two I²C segments, each capable of up to 400 pF each.

Buffers/repeaters allow for more slave devices to be added to the I²C bus in applications that have a heavily-loaded I²C bus. In applications utilizing the I²C bus, slave devices and trace length are major contributors to the total capacitance of the I²C bus. A quick way to estimate the amount of bus capacitance is to account about 10 pF per slave device. If the total capacitance nears the maximum load of 400 pF, there could be issues with the rise time falling out of the I²C specification.

The end of this section (Figure 10) shows a selection chart for narrowing down buffer selection in I²C applications that exceed capacitance loads.

4.1 Pull-Up Resistors

Pull-up resistors need to be connected from the I²C lines to the supply voltage to enable communication. In regards to the capacitive load on the bus, there is a max resistance value associated with the bus capacitance, while the minimum value is determined from the maximum allowed sink current (per I²C slave devices on the bus), offset voltage, and supply voltage. ⁽¹⁾Once the thresholds are determined, the system designer must determine if a stronger pullup is required to account for the capacitance. With a stronger pullup (lower resistance value), more current will be used to make rise times faster for SCL and SDA and ensure that I²C rise time specifications are met. For more information on pull-up resistors on the I²C bus, refer to *I²C Bus Pullup Resistor Calculation (SLVA689)*.

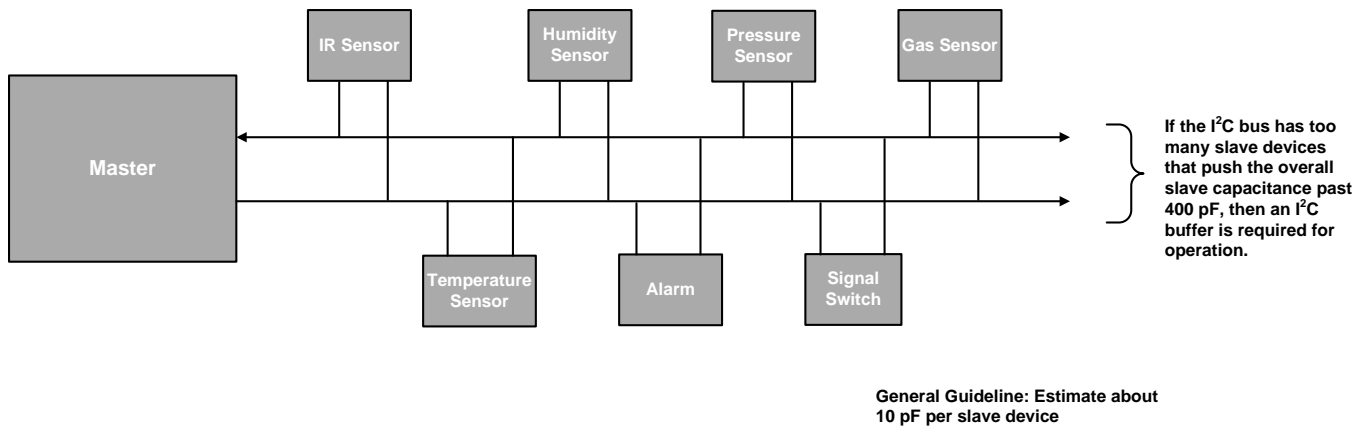


Figure 7. Heavily-Loaded Bus Requiring I²C Buffer

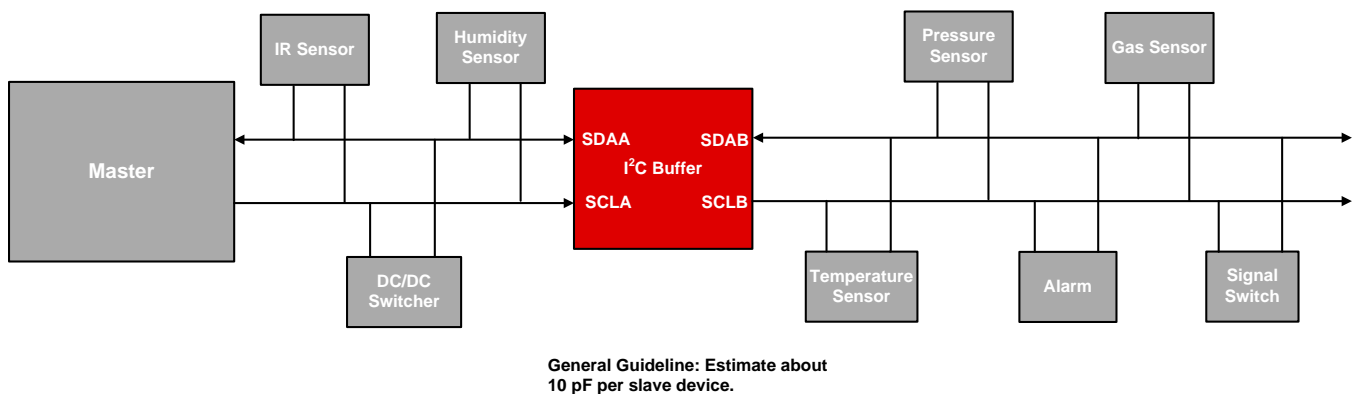


Figure 8. I²C bus utilizing I²C buffer, which allows more slave devices

4.1.1 Static Voltage Offset

Certain devices within the I²C buffer/repeater portfolio, such as TCA9517/A and TCA9617/A/B, contain a static voltage offset. The type of buffer design on the B-side prevents these devices from being used in series with devices that use a static voltage offset. When a static voltage offset is present on the B-side of these TCA devices, they will be unable to recognize buffered low signals as a valid low and do not propagate it as a buffered low again. An example of this would include, taking two TCA9517A devices and placing them in an A-B B-A configuration. By having both B-sides in series, the design runs the risk of miscommunicating low signals as the device will be unable to recognize this and result in transmitting a high.

⁽¹⁾ lost the footnote?

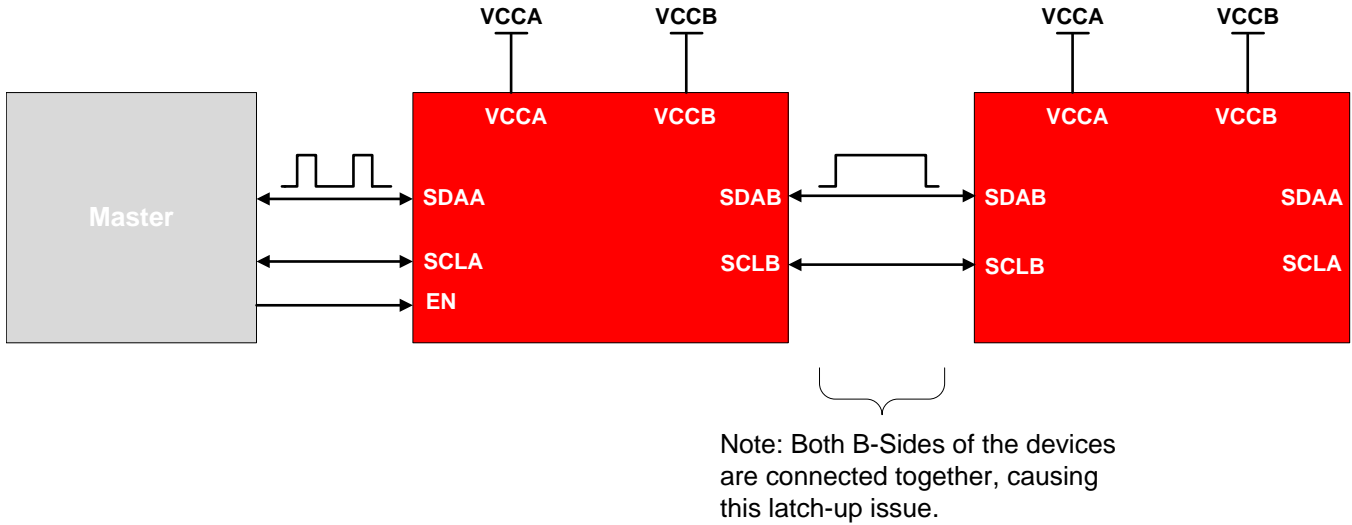


Figure 9. Example of B-B side mismatch configuration for I²C buffers

4.1.2 Level Shifting

Many of the devices in the I²C buffer/repeater portfolio have the ability to level-translate from master voltages to slave voltages to ensure communication between the voltage mismatch. For example, the TCA9517/A is an example of a level-translating buffer that has two voltage rails which allows for the mismatched master and slave voltages to appropriately communicate to each other through the I²C bus.

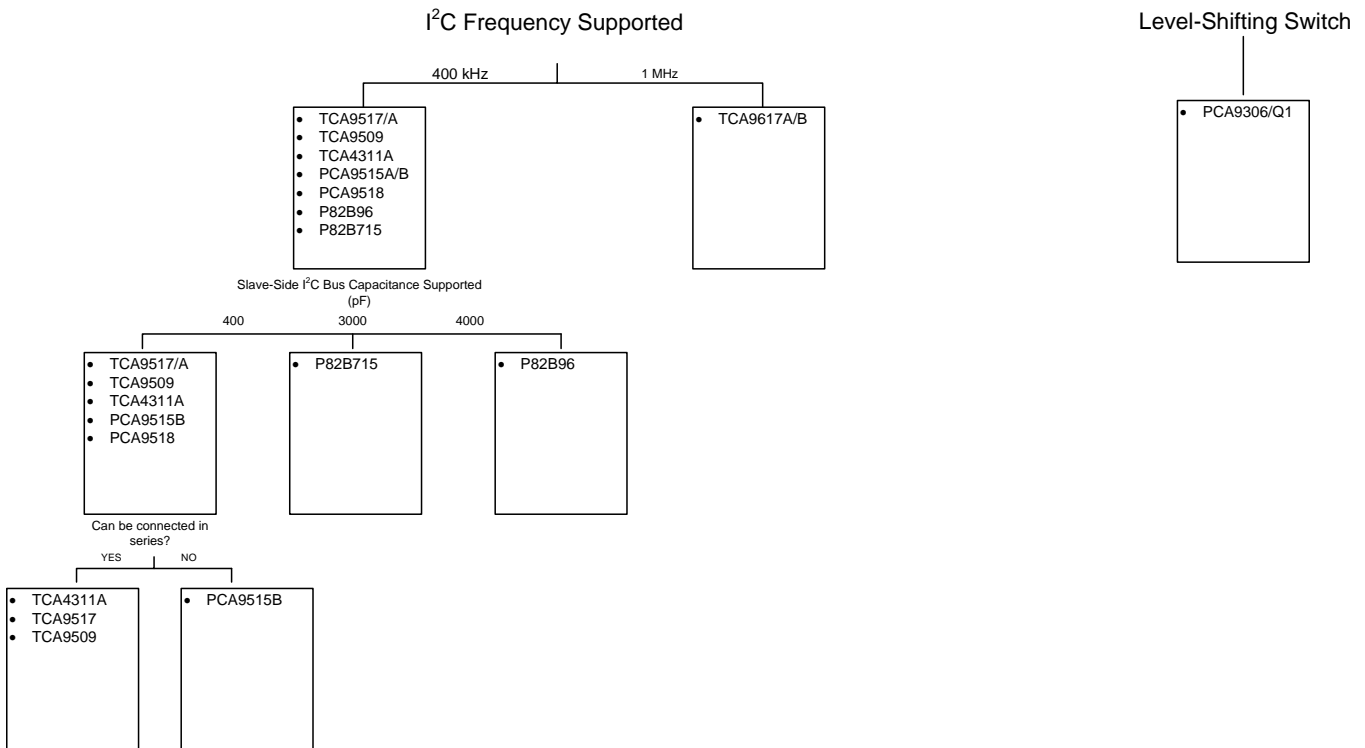


Figure 10. Selection Chart for TI I/O Expanders and Their Key Features

5 I²C Voltage Translation

While TI's I²C buffers/repeaters offer level shifting for master and slave voltage mismatches, TI also offers level shifting on the I²C bus if buffering is not needed. PCA9306 and TCA9406 are two devices in TI's portfolio that offer level shifting. The internal structure of both devices is similar to that of a switch which allows the bidirectional open-drain bus to pass through while level-shifting to ensure proper communication through I²C. The operation of the I²C level-shifting is the same as the voltage translation performed by I²C buffers/repeaters.

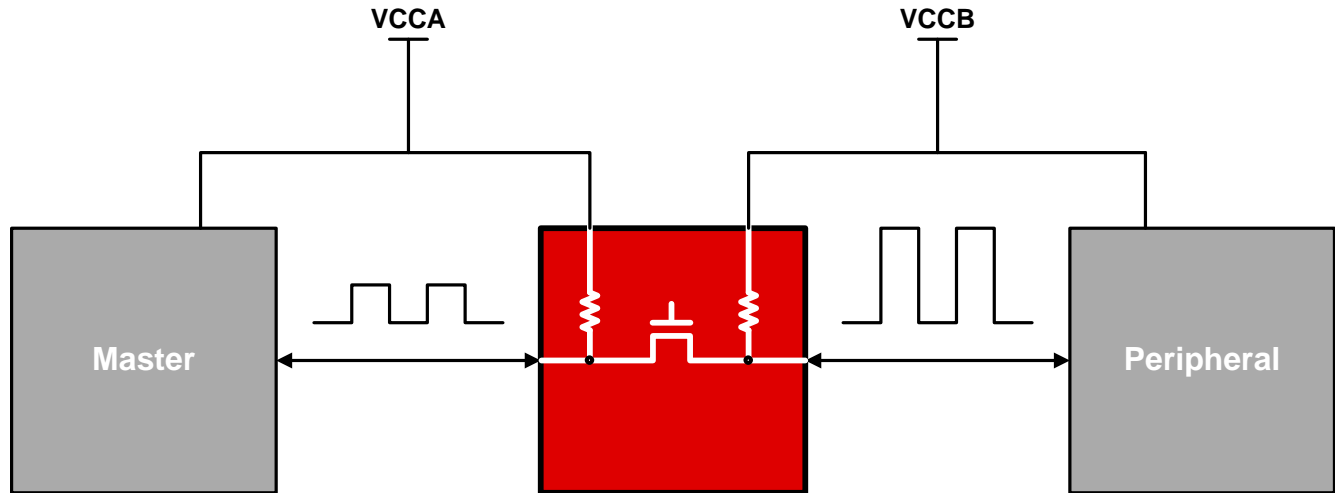


Figure 11. Operation of an I²C Voltage Translator/Level Shifter

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), [evaluation modules](#), and [samples](http://www.ti.com/sc/docs/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2018, Texas Instruments Incorporated