A New Approach to Making I²C Bus Buffers

SCOPE: The information in this paper helps electronic engineers and engineering managers stay informed of developments with the I²C (Inter-Integrated-Circuit) bus system.

All existing I²C bus buffers use the signal amplitude to set the signal flow direction, and use voltage offset techniques to avoid bus buffer latching. The no-offset buffer is different and unique in that it uses the signal timing as the direction flow control, thus eliminating many prior design headaches.

The I²C Bus – From Humble Beginnings to New Uses

The simplicity and low cost of the I²C bus has enticed electronic engineers to find more uses for the two-wire Inter-Integrated-Circuit protocol, and its many derivatives.

While the technology has been around for decades, and is still a viable solution for many maintenance and control applications, additional components have been invented to make it better and more useful. In particular, to overcome some of the basic limitations due to adding more devices to the bus or extending the bus length beyond its original maximum of just a few meters.

The most popular addition to the family has been the bus buffer, which relays the signals, but there have also been a variety of other devices invented that can switch the I²C-bus to different branches and greatly enhance its usefulness.

Revisions to the I²C Specification have introduced the 400 kHz Fast Mode and 1 MHz Fast-mode Plus (Fm+) I²C-bus clock speeds, increasing the data throughput and flexibility, but also creating a little more confusion for the design engineer.

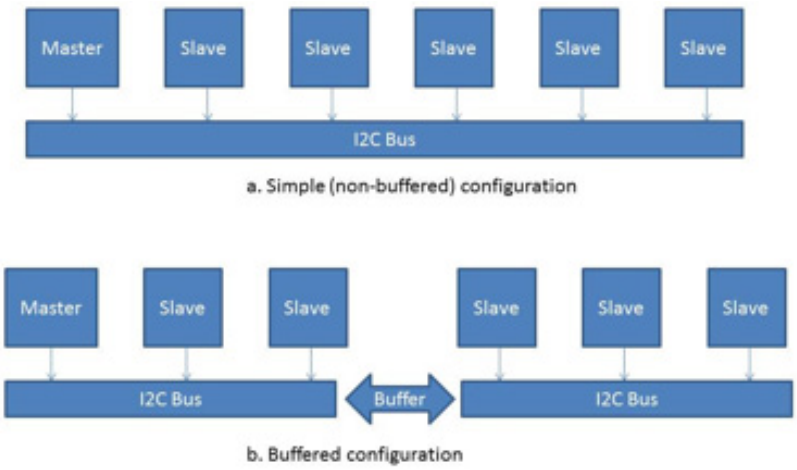


Figure 1. Dividing the I²C-Bus Using Bus Buffers

I²C bus buffers divide up a larger system into manageable sections. They pass bus traffic across boundaries, allowing an I²C-bus to function as a single communication scheme, Figure 1.

Bus buffers offer a range of benefits to circuit designers. Bus segments can have different operating voltages without impacting each other. Each section can support the maximum permitted capacitive loading, allowing a large or complex bus system to run at full speed. Also, sections can be isolated simply by disabling one or more bus buffers, which permits mixing of fast and slow slaves, and serves as a useful troubleshooting tool.

I²C Signal Flows

The I²C-bus employs handshaking, where messages from the master must be acknowledged by the target slave, and ignored by all other slaves on that section of the bus. A slave can send data to the master when requested making the simple two wire system bidirectional.

In a simple non-buffered implementation, all devices are hardwired to the signal lines, which by definition transmit signals in both directions, Figure 1a.

I²C-bus Buffer Basics

Introducing a buffer complicates the picture, since conventional buffer elements only allow signals to travel in one direction, Figure 1b. Signals don't flow in both directions at the same time, of course, but any bus buffer must instantly switch flow direction to avoid collisions.

Constructing an I²C-bus buffer with two unity-gain non-inverting amplifiers connected in an anti-parallel configuration, with one for each direction might at first appear to work, but it doesn't, Figure 2.

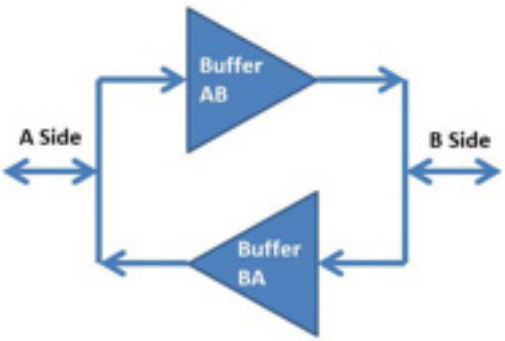


Figure 2. Basic Bus Buffer, Simple Block Diagram

Input signals appearing on the A side will propagate to the B side after the propagation delay of buffer AB, and a similar argument holds for signals appearing on the B side, the input to buffer BA.

However, in practice this design contains a serious flaw. To see why, let's investigate the device operation in a typical circuit, Figure 3.

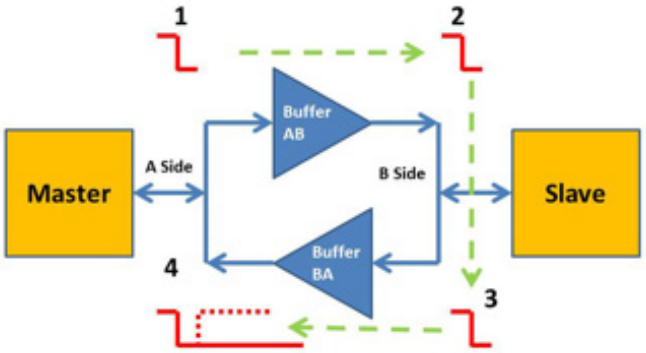


Figure 3. Buffered I²C Circuit Showing Latch-up Problem

The sequence of events is:

1. The Master device drives one side LOW, let's make it the A side.
2. The LOW signal propagates through Buffer AB and appears at side B.
3. Because side B is connected directly to the input of Buffer BA, the LOW signal now propagates through Buffer BA and drives side A LOW.
4. The Master releases side A, which would normally allow A to be pulled HIGH (dotted line) by the pull-up resistor. However, side A is being driven by Buffer BA, so it remains LOW (solid line).

The bus line has latched at a logic **LOW** (very close to zero volts, or logic zero) in a fatal error. To avoid this logic latch-up condition caused by the circuit configuration, the industry to date has developed techniques that manipulate the signal in some fashion.

Making I²C-bus Buffers that Work

Historically, there have been three developments that allowed I²C-bus buffers to avoid this latching. One type uses an analog amplifier to boost the drive current. The other two manipulate the amplitude of signals flowing through the bus buffer, using small offset voltages, and are classified as Incremental Offset or Static Offset types. The offset voltages are small enough to go undetected by other I²C-bus components, resulting in a practical I²C-bus system that extends the original I²C-bus concept.

All of these buffer techniques involve some performance compromise that generally goes unnoticed. For most applications, a solid design can be created with the fewest number of bus buffers and additional components.

Innovation in I²C-bus Buffer Techniques — The No-offset Type

The newest bus buffer technique uses timing as the direction flow control criteria.

This device is called the no-offset bus buffer. This frees the bus signals from those added voltage offsets used in other bus buffer designs. It also allows the no-offset buffer to be used in ways that are not permitted for the previous bus buffer types, such as placing multiple bus buffers in parallel or series configurations.

To understand how the no-offset bus buffer operates, think of a sporting event with a score board where competing teams race to win. Signals on either side of the no-offset bus buffer also "race" each other to determine which one will gain control of the buffer direction. When one side wins, the buffer control logic disables the other side until the winning side is released.

The no-offset bus buffer has two anti-parallel buffers, which can be turned off by the control logic, Figure 4.

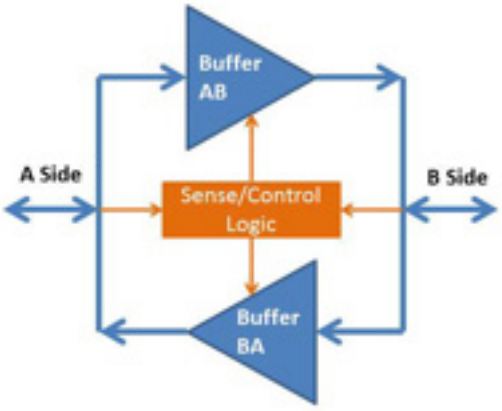


Figure 4. No-offset Bus Buffer, Simple Block Diagram

The control logic monitors the inputs on both side A and B and controls which buffer is active. By ensuring that only one buffer is active at a time during bus operation, the no-offset bus buffer provides bi-direction data transfer without a direction pin and avoids the fatal latch-up condition.

Key Features of No-offset Bus Buffers

No-offset bus buffers are easy to use and meet the I²C specification for voltage and current compliance.

Output Drive

The output stage is an open-drain field effect transistor (FET), or an open collector bipolar junction transistor (BJT), Figure 5. When the I²C-bus is idle the output of the bus buffer is high impedance, allowing the bus line to be pulled high by the external bus pull up resistor. An active bus buffer sinks current through the FET or BJT, pulling the bus line low.

Isolation

No-offset bus buffers regenerate the signals, effectively isolating the capacitance loading on either side of the bus buffer. These devices are sometimes referred to as repeaters.

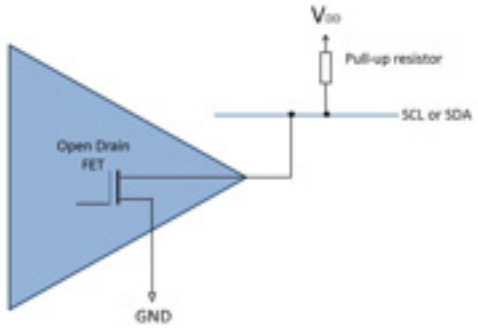
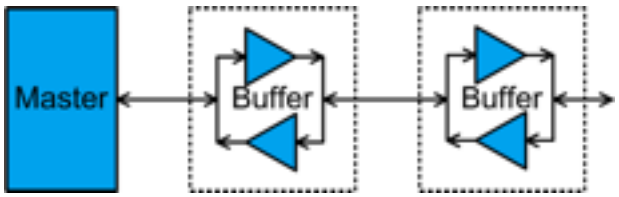


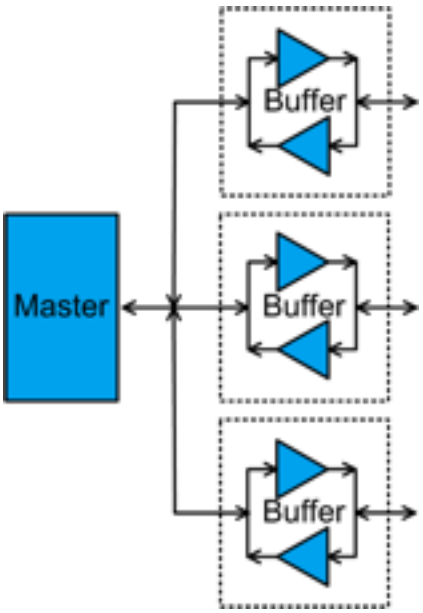
Figure 5. FET Output Circuit Used in No-offset Bus Buffer

Interconnection to Other Bus Buffers

Because the no-offset bus buffer regenerates the signals, it removes any offset voltage coming in from another bus component, allowing many no-offset bus buffers to be placed in cascade, something that can't be done with bus buffers that employ voltage offset, Figure 6.



Series connected scheme



Star (parallel) connected scheme

Figure 6. Connecting No-offset Bus Buffers Together

Drive Strength

With versions that provide 4mA and 30mA drive strengths (see Figure 11), the new no-offset bus buffers easily provide enough current to meet the I²C specification requirements, shown in Figure 7, while providing flexibility in price to performance.

Symbol	Parameter	Conditions	Std-mode	Fast-mode	Fast-mode +	Unit
			Min	Min	Min	
IOL	LOW-level output current	VOL = 0.4 V	3	3	20	mA
IOL	LOW-level output current	VOL = 0.6 V	n/a	6	n/a	mA

Figure 7. I²C Specification for SDA and SCL I/O Stage Current

Design Trade-offs with No-offset Bus Buffers

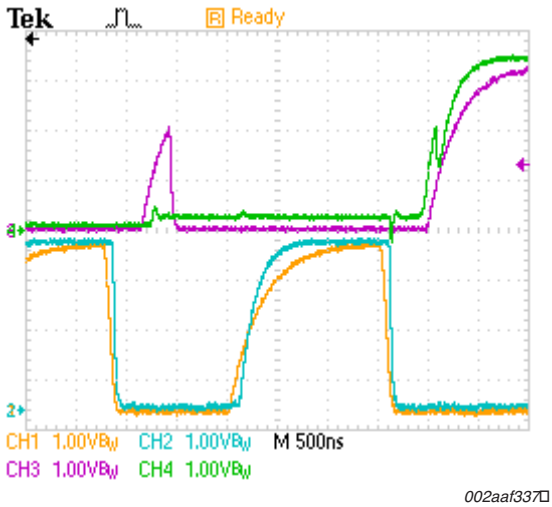
No bus buffer is perfect; glitch pulses are generated by any I²C bus buffer when the Master and Slave exchange control, Figure 8.

Fortunately the glitch on the data line (SDA IN, purple in the figure) does not fall on a clock edge, and is therefore ignored by the Slave devices. Many I²C devices have digital noise filters on their inputs, further reducing the risk from false pulses.

Clock Buffering (SCL Line)

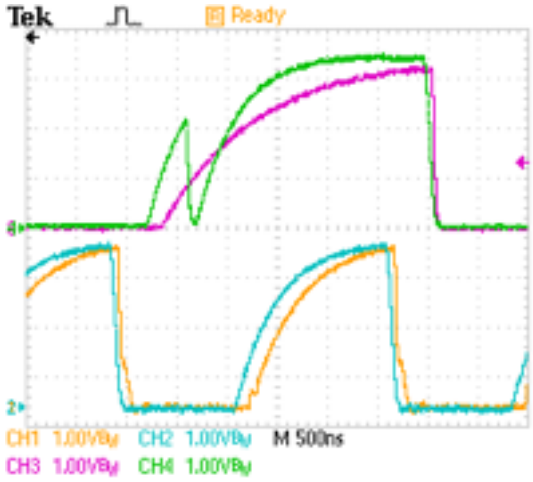
The majority of I²C bus schemes have only one bus Master, which always generates the clock signal. The I²C specification allows for multi-master or Slave-master devices which may take control of the clock signal. Any bus buffers on the bus between two masters would have to pass the data (SDA) and the clock (SCL) in both directions when control is handed off.

For this reason, the no-offset bus buffer passes the clock signal in one direction only, but allows direction reversal on the data signal. Applying the no-offset bus buffer to multi-master schemes requires care.



Ch1 (orange): SCL_IN Ch2 (blue): SCL_OUT
Ch3 (purple): SDA_IN Ch4 (green): SDA_OUT

Figure 8. Glitch Waveform Generated by Direction Reversal



Ch1 (orange): SCL_IN Ch2 (blue): SCL_OUT
Ch3 (purple): SDA_IN Ch4 (green): SDA_OUT

Figure 9. Glitch Waveform Generated by Bus Rise Time Unbalance

Data Buffering (SDA Line)

Referring to Figure 9, we see a glitch pulse on the SDA OUT (green trace) due to mismatched bus rise times, a unique condition with the no-offset bus buffer technique.

An easy fix is correct selection of bus pull up resistors on either side of the no-offset bus buffer. This technique eliminates the glitch by making the rise time of both sides approximately the same.

Introducing the Industry's First No-offset Bus Buffers

NXP pioneering no-offset bus buffer technology is currently available in three products: NXP PCA9525 for Fast-Mode I²C applications, NXP PCA9605 with higher current drive capability for Fast-Mode Plus or heavily loaded buses, and NXP PCA9646 that integrates a four channel multiplexer/switch with no-offset bus buffers, Figure 10. These devices are monolithic CMOS integrated circuits that feature:

- No-offset voltage
- Support for I²C Fast Mode and Fast-mode Plus
- Good noise immunity due to hysteresis on the inputs
- Operating voltages from 2.7 V to 5.5 V with low supply current
- Enable pin to allow bus segments to be isolated
- Parallel connection permitted on all buffer I/Os, increased design flexibility

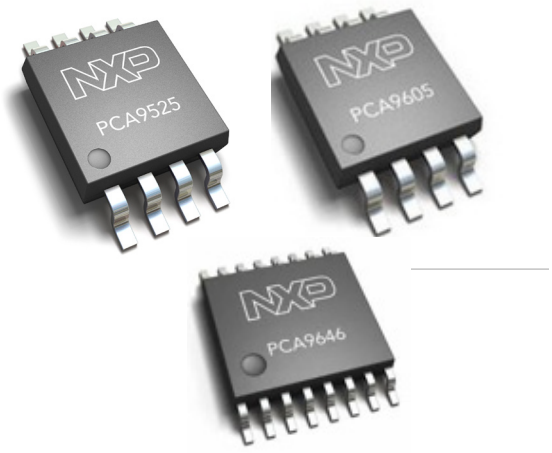


Figure 10. The NXP Family of No-offset Bus Buffers

Application Overview

Figure 11 illustrates how to use the no-offset bus buffers in systems to take maximum advantage of the design capabilities and achieve the best price to performance flexibility.

The PCA9525 provides designers with a lower cost option for systems that do not require the higher output drive strength. It works well for I²C-bus applications that meet the 3mA specification.

With drive strength of 30 mA at 0.4V, the PCA9605 bus buffer is compatible with any of the I²C operating modes. Use this buffer to extend a bus operating at 3mA on one side, for example, and up to 30mA on the other, to drive the higher capacitance of a long cable or many slaves.

The PCA9646 bus switch combines the new no-offset buffers with a four channel switch, replacing a popular non-buffered predecessor (PCA9546). Each channel can drive up to 30mA without additional buffers, even if multiple channels are connected together at the same time, something that could not be done with earlier (voltage offset) buffer technology.

	PCA9525 Bus Buffer *4 mA output drive at 0.4V*	PCA9605 Bus Buffer *30 mA output drive at 0.4V*	PCA9646 Bus Switch *30 mA output drive at 0.4V*
Standard-mode 100 kHz **Fast-mode 400 kHz**	Yes	Yes	Yes
Fast-mode Plus 1 MHz		Yes	Yes

Figure. 11 Overview of No-offset Buffer Price to Performance Applicability

NXP PCA9525: I²C-bus Buffer for Cost-Sensitive Applications

The PCA9525 is the simplest and most economical buffer in the family, with a maximum output sink current of 4 mA. It incorporates a bidirectional auto-reversing data buffer for the SDA line and a unidirectional clock buffer for the SCL line, Figure 12.

The PCA9525 may be used to buffer the actual bus load on the bus Master, typically a CPU or microcontroller device. The PCA9525 can operate at speeds over 1MHz (with appropriate loading limitations, see Figure 11).

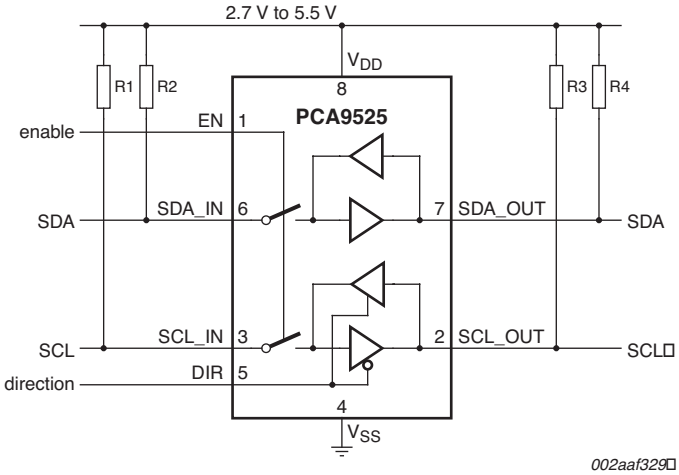


Figure 12. PCA9525 Block Diagram

A large number of PCA9525s can be connected in parallel or series to extend the bus distance and number of slave devices, Figure 13. PCA9525 removes any offsets and regenerates a fresh LOW output signal of just 80 mV (typical) at a sink current of 4 mA—regardless of the input LOW voltage.

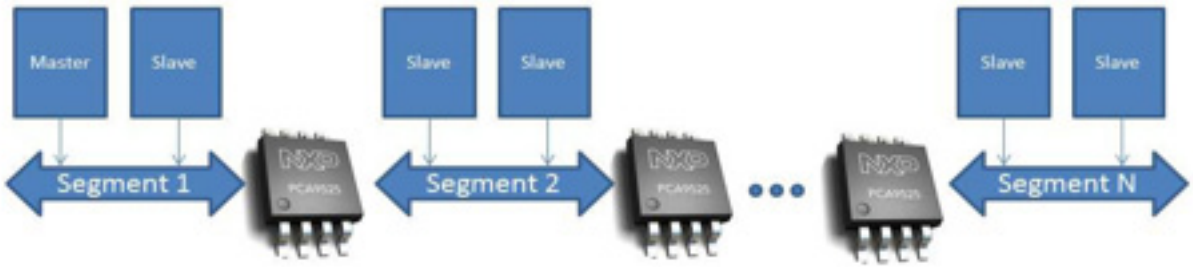


Figure 13. Multiple I²C Bus Segments Buffered with NXP PCA9525s

NXP PCA9605: I²C Bus Buffer with Higher Drive Capability

The PCA9605 has the same internal structure and is pin-compatible with the PCA9525 operating up to 1 MHz but with stronger output drivers that can sink up to 30 mA output current. The PCA9605 is suitable for standard, Fast Mode, and Fast-mode Plus (Fm+) implementations.

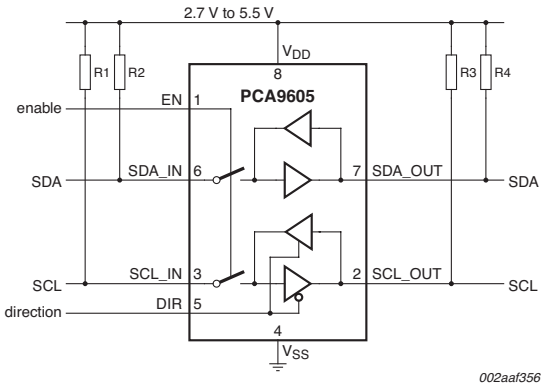


Figure 14. NXP PCA9605 Block Diagram

Like the PCA9525, the PCA9605 regenerates the logic signals and therefore can be used to extend the I²C bus for modest distances, up to as much as 20 meters using twisted pair cable, Figure 15.

Another application for the PCA9605 is to extend an I²C-bus much farther by breaking the long cable into shorter segments and placing a buffer between each segment, Figure 16.

For more information, please refer to NXP Application Note AN11075: Driving I²C-bus signals over twisted pair cables with PCA9605

The PCA9646 takes advantage of the no-offset bus buffer's ability to be placed in parallel with other no-offset bus buffers, which is not possible with static or incremental offset or amplifier bus buffer types. Often it is necessary to divide an I²C-bus scheme into sections, to avoid overlap of slave addresses or to mix different speed slave devices together, for example.

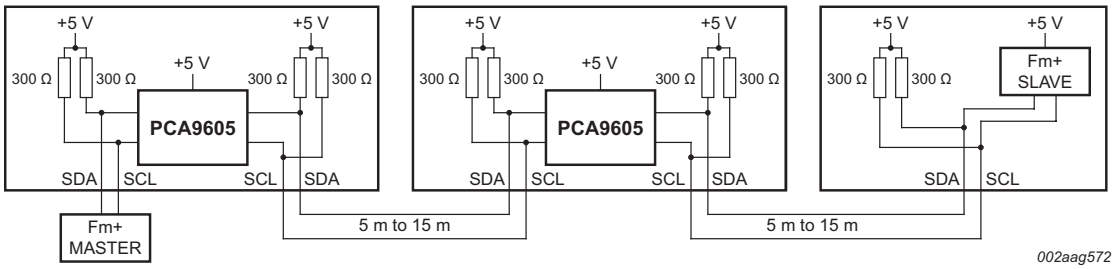


Figure 15. Designing an Extended I²C Bus Using NXP PCA9605s

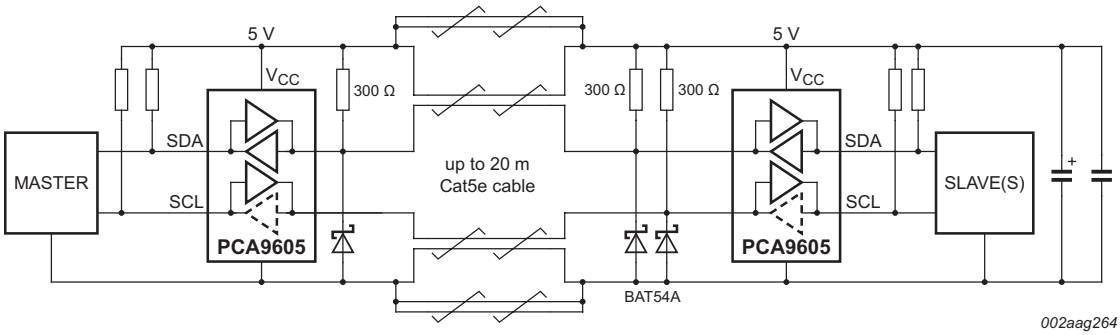


Figure 16. NXP PCA9605s Used to Drive Twisted Pair Cables

Each of the five ports on the PCA9646 is connected to internal no-offset bus buffers, Figure 16. One of these is connected upstream to the bus Master, and configuration of the other four ports is then by an internal control register similar to the PCA9546A. The PCA9646 has three address lines, allowing multiple PCA9646 devices to share the same bus.

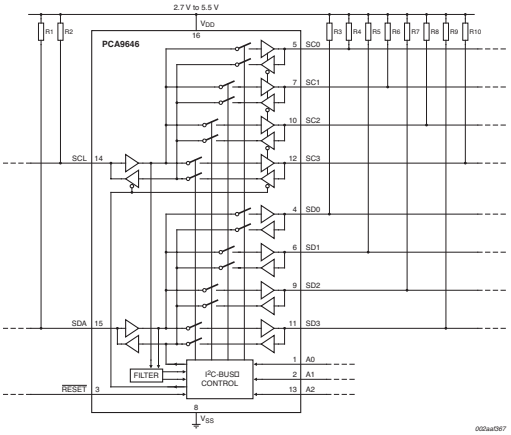


Figure 17. NXP PCA9646 Block Diagram

The PCA9646 is well suited for many applications:

- Large LED displays for signage and other uses
- Control systems for solar panel arrays
- Power management systems in high-rise buildings and campus environments

Figure 17 shows the PCA9646 used as a line driver. One device can drive four such lines (only one shown). The receiving end may then again be used as a four-way bus switch, radiating out into another four lines. Using the address pins, this entire structure may be repeated. Thus a total of eight PCA9646 line drivers can be connected to a single bus master (U1), allowing a single I²C bus port to drive 32 (8x4) long-distance bus pairs.

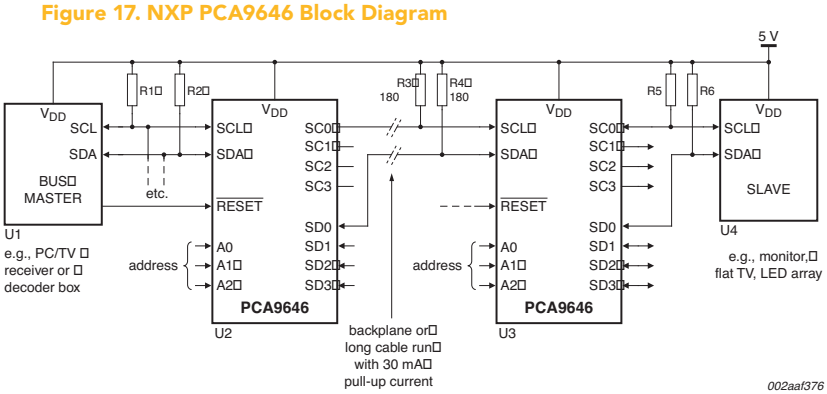


Figure 18. 30mA Line Driver Design Using the NXP PCA9646

The PCA9646 can support very large systems including many hundreds of slaves and hundreds of meters of wiring, applications where EIA485 (formerly RS485) interface devices have been used.

For more information, please refer to NXP Application Note AN11084: Very large I²C-bus systems and long buses

Summary

As designers continue to find new and challenging applications for the I²C-bus, they often push against the limits of the bus for distance and number of devices. NXP's development of the no-offset bus buffer removes these restrictions. Unlike static offset buffers, the NXP no-offset bus buffers regenerate clean logic LOW signals on *both* sides of each buffer, allowing them to be placed in parallel as well as in series. This approach divides the bus into smaller segments, extending the possible reach to hundreds of meters and effectively eliminating load restrictions.

NXP offers three different no-offset bus buffer devices: the PCA9525 for cost-sensitive designs up to 1 MHz but less than 400 pF, the PCA9605 for 1 MHz high-current and Fast-mode Plus applications and the PCA9646 for 1 MHz high-current and Fast-mode Plus applications which require a four channel multiplexer/switch for very large systems or address conflict resolution. For more information on these innovative and powerful products, visit: http://busbuffers.interfacechips.com.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com
Date of release: January 2012