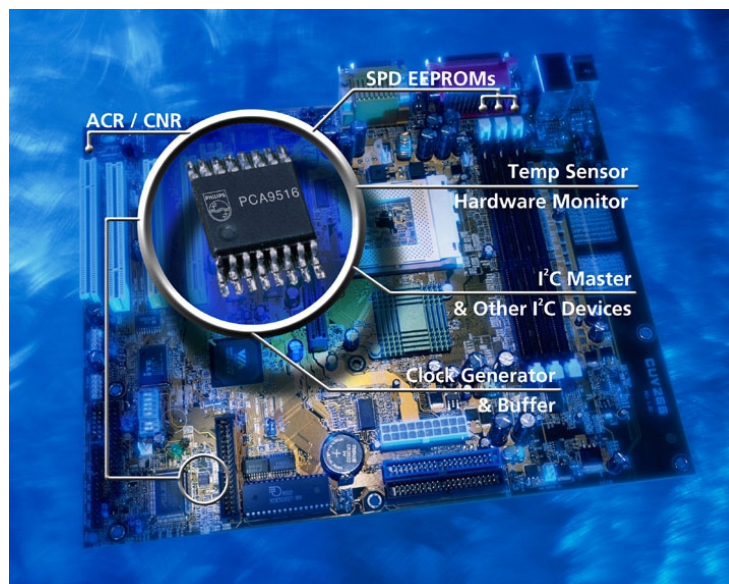


APPLICATION NOTE



AN255-02

I²C / SMBus REPEATERS, HUBS AND EXPANDERS

PCA9511, PCA9512, PCA9513, PCA9514, PCA9515, PCA9516, PCA9518, P82B96, P82B715

Abstract - Philips Semiconductors family of Repeaters, Hubs and Expanders are detailed in this application note that discusses device operation, maximum cable length and frequency calculations and typical applications.

Phil Tracy – Senior Field Applications Engineer

Alma Anderson – Senior Design Engineer

Jean-Marc Irazabal - Technical Marketing Manager

Steve Blozis - International Product Manager

Specialty Logic Product Line
Logic Product Group



Purchase of Philips I²C components conveys a license under the Philips' patent to use the components in the I²C system provided the system conforms to the I²C specification defined by Philips.

Philips Semiconductors

December , 19 2002

TABLE OF CONTENTS

TABLE OF CONTENTS	2
OVERVIEW.....	3
DESCRIPTION	3
APPLICATIONS	4
DEVICE FEATURES AND CHARACTERISTICS	5
DEVICE PINOUT	7
ORDERING INFORMATION	7
DATA SHEETS AND IBIS MODELS	7
PCA9511/12/13/14 TECHNICAL INFORMATION	7
BLOCK DIAGRAM	7
PCA9515/16/18 TECHNICAL INFORMATION	9
BLOCK DIAGRAM	9
VOLTAGE LEVEL TECHNICAL INFORMATION	9
DESIGN CONSTRAINTS	10
P82B715 TECHNICAL INFORMATION.....	11
BLOCK DIAGRAM	11
VOLTAGE LEVEL TECHNICAL INFORMATION	11
DESIGN CONSTRAINTS	12
P82B96 TECHNICAL INFORMATION.....	13
BLOCK DIAGRAM	13
VOLTAGE LEVEL TECHNICAL INFORMATION	14
DESIGN CONSTRAINTS	14
USE OF MULTIPLE DEVICES IN THE SAME I²C SYSTEM.....	15
DETERMING PULL-UP RESISTORS VALUE	16
APPLICATIONS	16
BUS EXPANSION BEYOND THE 400 pF LIMITATION	16
SPLIT BUS 100 KHz AND 400 KHz APPLICATION	17
VOLTAGE LEVEL SHIFTING.....	17
I ² C MULTIPLEXING.....	18
I ² C MULTI-POINT HOT SWAP APPLICATIONS	19
I ² C BUS EXTENSION OVER LONG CABLES.....	19
I ² C OPTO-ELECTRICAL ISOLATION	21
MULTIPLE SEGMENT APPLICATION	22
FREQUENTLY ASKED QUESTIONS.....	23
DEVICE USAGE AND DIFFERENCES	23
LIVE INSERTION	25
I ² C OVER LONG DISTANCES	27
ELECTRICAL PARAMETERS.....	30
I ² C PROTOCOL	35
ARCHITECTURE / APPLICATIONS.....	36
ADDITIONAL INFORMATION.....	46
APPENDIX 1 – Long Bus Calculations	47
CALCULATION OF MAXIMUM CLOCK FREQUENCY FOR LONG BUSES	47
THE ‘SHORTCUT’ APPROACH TO LONG BUS APPLICATIONS.....	49
SOME CALCULATION EXAMPLES	50
APPENDIX 2 – Using P82B715 to increase the total allowable bus capacitance.....	52
APPENDIX 3 - Operating P82B715 on lower supply voltages	53
APPENDIX 4 - The special logic levels and noise margins of buffer logic	54
APPENDIX 5 - Using P82B96 to create a multi-drop “distribution” bus using 12 V logic levels.....	55
APPENDIX 6 - Differences Between the LTC1694 and LTC1694-1.....	57
APPENDIX 7 - Typical response times to 'data valid' on SDA, measured from the falling edge of SCL.....	58

OVERVIEW

Description

The PCA9511, PCA9512, PCA9513, PCA9514, PCA9515, PCA9516, PCA9518, P82B715 and P82B96 allow a larger or longer I²C or SMBus system where the following constraints previously restricted its size:

- a) More I²C devices or larger wiring capacitance than the 400 pF maximum allowed in the I²C specification.
- b) Long bus wiring or backplanes
- c) Different operating supply voltages or logic voltage levels within one system
- d) Opto-isolation for safety or due to difference in ground plane
- e) Isolating a section of a system that has lost its power supply
- f) Insertion of unpowered cards into an active I²C bus

With only one repeater delay between any two segments, the:

- PCA9511/12/13/14 Hot Swappable I²C Buffers supports two segments of 400 pF and hot swapping with pin precharge and idle detect features.
- PCA9515 I²C Bus Repeater supports two segments of 400 pF with channel enable.
- PCA9516 5-channel I²C Hub supports five segments of 400 pF for a total I²C or SMBus capacitance of 2000 pF with 4 channel enables.
- PCA9518 Expandable 5-channel I²C Hub supports multiple groupings of five segments of 400 pF each for an unlimited I²C or SMBus capacitance with 4 channel enables per device.

The PCA9511/12/13/14 are designed to work with clock frequencies up to 400 kHz and are suitable for utilization in a multi-master I²C bus or SMBus environment. They are specifically designed for backplane multi-point applications and hot swap. The devices support bus arbitration and contention with bus masters located on any segment. They are multi-directional and require no direction control. Operational voltage supply is 2.7 V to 5.5 V with 5.5 V tolerant I/Os. Operational temperature range is -40 °C to 85 °C. See Philips application note AN10160-01 for more detailed technical information on these devices.

The PCA9515/16/18 are designed to work with clock frequencies up to 400 kHz and are suitable for utilization in a multi-master I²C bus or SMBus environment. They are specifically designed for larger buses on single card applications with the PCA9516 or PCA9518 for star configuration point to point connections. The devices support bus arbitration and contention with bus masters located on any segment. They are multi-directional and require no direction control. Operational voltage supply is 3.0 V to 3.6 V with I/Os that are 6.0 V tolerant allowing easy interfacing of 3.3 V and 5 V devices. Operational temperature range is -40 °C to 85 °C.

The P82B715 allows bi-directional signaling without an external direction control over wiring capacitance loads up to around 3000 pF. At normal I²C logic voltage levels, it transforms the impedance of the wiring by a factor 10, so 3000 pF of wiring appears as 300 pF loading on the I²C bus. The P82B715 does not afford any isolation or support logic voltage level translation. The P82B96 is recommended for new designs due to its superior features.

The P82B96 allows bi-directional signaling without an external direction control and provides a true buffer functionality supporting logic levels from 2 V to 15 V and bus impedance 10 times lower than I²C. It is designed for long distance signaling, multi-point applications and opto-isolation of bus segments in situations like power supplies with different ground planes.

Normally, **only one** PCA9515 Repeater or PCA9516 Hub should be used per I²C or SMBus system due to the way they operate to provide multi-directional signally without an external direction control. The PCA951x series of Hot Swappable Bus Buffers, the PCA9518 Expandable Hub or the P82B715/96 Bus Buffers should be used for any application requiring multiple devices in the same system.

This Application Note covers the use of these devices in a generic multiple master environment.. Information on the permitted interfaces and bus propagation delays between P82B715/96 Buffers and PCA951x/15/16/18 are also included.

Applications

These devices can be used for a wide variety of applications:

I²C or SMBus Extension – All devices allow extension of the I²C or SMBus on systems requiring capacitance loads larger than the 400 pF max specified by the I²C protocol. Designers can extend the use of the I²C or SMBus in systems with more devices and / or longer bus lengths with only one repeater delay between any device. The devices are multiple master capable and support arbitration and bus contention on any segment.

- The PCA9511/12/13/14/15 I²C Bus Repeater supports two I²C or SMBus branches of 400 pF
- The PCA9516 5-channel I²C Hub enables separation of the I²C or SMBus into five 400 pF segments
- The PCA9518 Expandable 5-channel I²C Hub enables separation of the I²C or SMBus into multiple groupings of five 400-pF segments for star configurations.
- The P82B715 and P82B96 provide the multi-point bus alternative. Working in the conventional I²C ‘wired and’ mode, but with an impedance level that may be 10 times lower than I²C thus allowing up to 4000 pF bus capacitance. P82B96 allows selection of any convenient logic voltage level up to 15 V with no restriction on the number of ‘drops’.

I²C or SMBus Isolation - Each PCA9511/13/14/15/16/18 individual repeater channel has an enable/disable feature that is used to electrically isolate that segment of the I²C or SMBus. In addition, all devices except the P82B715 release their I/O pins should their supply voltage fail. When an enable input is required with P82B96 the Picogate circuit in Figure 1 may be used.

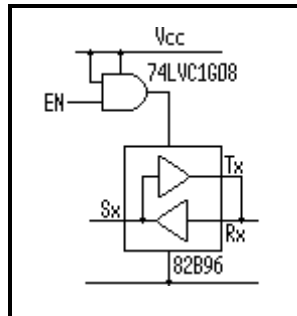


Figure 1. P82B96 Enable Input Configuration

Examples where bus isolation could be useful include:

1. Allow mixed operation of 100 and 400 kHz devices on a combined bus by isolating the segment with the 100 kHz devices from the rest of the 400 kHz devices so that the 400 kHz devices can operate at their maximum speed.
2. Supporting the PCI management bus with 8 or more PCI slots
3. Isolating the I²C or SMBus to support hot plug PCI slots
4. Allowing the main I²C or SMBus to drive multiple system boards
5. Addressing selected devices if there are multiple devices in the system with the same I²C address.
6. Sensing multiple power supplies on different bus segments and providing automatic isolation of failed segments or segments suffering ‘brown-out’ of their power supplies. An example of such sensing using the PCA9515 is shown in Figure 2. In this example a failure of EITHER Supply 1 OR Supply 2 will cause I²C 1/Bus0 to be disconnected from I²C 2/Bus1. The PCA9512 with its two V_{CC} supplies and ACC pin can also be used in this application by tying ACC to V_{CC}.

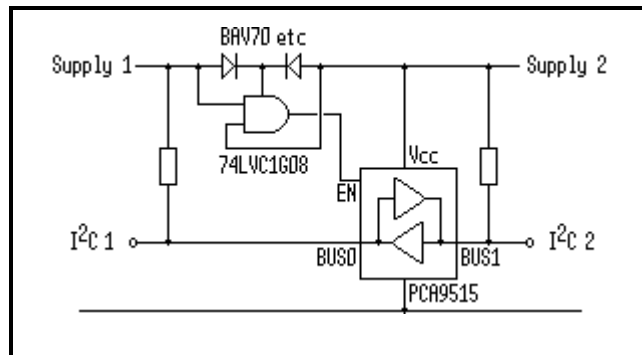


Figure 2. Multiple Voltage Sensing Example

Voltage Translation - The PCA9515/16/18 operate on 3.3 V V_{CC} but since they have 6.0 V tolerant I/Os, they support different I²C or SMBus segments at 3.3 V or 5 V, thus extending the life of older 5 V circuit blocks while allowing use of newer 3.3 V devices for improved system performance.

PCA9511/13/14 do **NOT** support logic level translations. The PCA9512 has two V_{CC} supply inputs specifically designed for voltage translation applications.

P82B715 does **NOT** support logic level translations. Its I/O pins simply track each other within 100mV, independent of the power supply (V_{CC}). Its logic levels may not exceed V_{CC} and will be diode clamped at $V_{CC} + V_{diode}$.

P82B96 is intended for operation with 5 V logic on its Sx side but can operate down to 3.3 V. The logic threshold on the Sx side is independent of the device's V_{CC} . The logic levels on the higher drive Rx/Tx side are half-rail CMOS switching levels related to the device's supply (V_{CC}) that is specified from 2 V to 15 V. All I/Os are 18 V tolerant.

The GTL2000/02/10 allow I²C voltage level translation at any voltage between 1.0 V and 5.0 V. See application note AN10145-01 Bi-Directional Voltage Translators for more information.

Hot Swapping Line Cards - The PCA9511/12 are featured with circuitry that precharges the SDA and SCL pins to prevent glitching data on an active I²C bus when the unpowered slave card is inserted into the backplane system. They also feature an idle detect circuit that will not connect the active bus to the line card devices until the bus is idle. The PCA9513/14 do not have the precharge feature since it is not needed or desired in some applications. The PCA9513 has a 92 μ A current source for PICMG applications.

Device Features and Characteristics

PCA9511/12/13/14:

- Bi-directional buffer for SDA and SCL lines
- Prevents SDA and SCL corruption during live board insertion and removal from backplane
 - Idle detect circuit connects board only when bus is idle
 - 1 V precharge on all SDA and SCL lines prevents data glitch (PCA9511/12 only)
- Compatible with I²C, I²C fast mode, and SMBus standards
- DV/Dt rise time accelerators on all SDA and SCL lines
- Rise time accelerator disable (PCA9512 only)
- Low I_{CC} device disable: <1 mA (PCA9511/13/14 only)
- READY open-drain output (PCA9511/13/14 only)
- 92 μ A current source for PICMG applications (PCA9513 only)
- Operating voltage range is 2.7 V to 5.5 V
- 5.5 V tolerant I²C and enable pins
- Split V_{CC} for 3.3 V and 5 V level translation (PCA9512 only)
- Operating temperature range is -40 °C to 85 °C
- Accommodate 100 kHz and 400 kHz devices and multiple masters
- ESD protection exceeds:
 - 2000 V HBM per JESD22-A114
 - 200 V MM per JESD22-A115
 - 1000 V CDM per JESD22-C101
- Latch-up testing is done to JEDEC Standard JESD78 which exceeds 100 mA
- PCA9511/12/13/14 offered in 8-pin SO (D) and TSSOP (DP)

PCA9515/16/18:

- Accommodate more I²C devices or a longer bus length, up to 400 pF per segment
- Multi-directional I²C drivers that require no directional control isolate the I²C bus capacitance of each segment.
- Powered-off high impedance I²C pins support hot insertion and tolerant up to 6.0 V.
- Open drain outputs and fail safe (will not inhibit attached devices) operation on loss of power
- Designed for I²C-bus operation but compatible with SMBus
- Lock-up free operation
- Multi-master capable repeaters support bus arbitration and clock stretching with only one repeater delay between segments

- Operating voltage range is 3.0 V to 3.6 V
- 6.0 V tolerant I²C and enable pins - pull-up resistors to either 3.3 V or 5 V voltage levels are allowed on the individual segments since the I²C bus and enable pins are over voltage tolerant to 6.0 V.
- Individual channel enables allow segments to be individually isolated. Hardware enable pins operate at LVTTL voltage levels and are over voltage tolerant to 6.0 V.
- Operating frequency range from 0 kHz to 400 kHz
- Operating temperature range is -40 °C to 85 °C
- ESD protection exceeds:
 - 2000 V HBM per JESD22-A114
 - 200 V MM per JESD22-A115
 - 1000 V CDM per JESD22-C101
- Latch-up testing is done to JEDEC Standard JESD78 which exceeds 100 mA
- PCA9515 offered in 8-pin SO (D) and TSSOP (DP)
- PCA9516 offered in 16-pin SO narrow (D) and TSSOP (PW)
- PCA9518 offered in 20-pin SO (D) and TSSOP (PW)

P82B715:

- Accommodate more I²C devices or a longer, lower impedance bus wiring. Up to 3000 pF wiring capacitance.
- Bi-directional unity voltage gain buffering with no external directional control required.
- 10x impedance transformation but does not change logic voltage levels.
- Wide supply voltage range of 3.0 V to 12.5 V.
- Logic signal levels independent of device supply and may include (but not exceed) supply and ground.
- I²C-bus compatible but not compatible with SMBus.
- Manufactured in rugged bipolar process (includes diodes to V_{CC}).
- Operating frequency range from 0 kHz to 100 kHz
- Operating temperature range is -40 °C to 85 °C
- ESD protection exceeds:
 - 2500 V HBM per Mil. Std. 883C – 3015.7
- Latch-up free (bipolar process with no latching structures).
- P82B715 offered in 8-pin DIP (N) and SO (D)

P82B96:

- Can split bi-directional I²C signals into two unidirectional Tx and Rx data signals, or recombine these.
- 10x bus impedance transformation option, unlimited bus drive capability with external transistor.
- Multi-drop extension bus configuration supports full I²C protocol.
- Open collector outputs and fail safe operation (will not inhibit attached devices) on loss of power.
- Designed for I²C-bus operation and compatible with SMBus
- Powered-off high impedance and 18 V rated I/O pins support hot insertion or cable plug-in.
- Low supply current (1 mA)
- Manufactured in rugged bipolar process.
- Operating voltage range is 2.0 V to 15.0 V
- 18 V tolerant I/O pins - pull-up resistors to different voltage levels are allowed on each side of the device.
- Usual operating frequency range from 0 kHz to 400 kHz but will clock above 600 kHz
- Operating temperature range is -40 °C to 85 °C
- ESD protection exceeds:
 - 3500 V HBM per JESD22-A114
 - 1000 V CDM per JESD22-C101
- Latch-up free (bipolar process with no latching structures).
- P82B96 offered in 8-pin DIP (N), SO (D) and TSSOP (DP)

Device Pinout

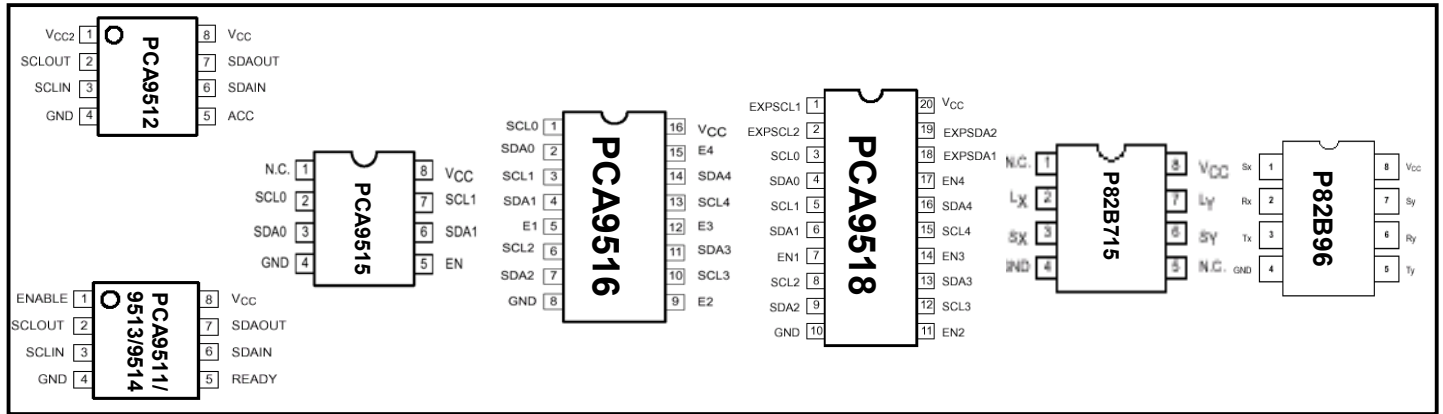


Figure 3. Device Pinout

Ordering Information

Package	Container	PCA9511	PCA9512	PCA9513	PCA9514
SO	Tube	PCA9511D	PCA9512D	PCA9513D	PCA9514D
	T & R	PCA9511D-T	PCA9512D-T	PCA9513D-T	PCA9514D-T
TSSOP	T & R	PCA9511DP-T	PCA9512DP-T	PCA9513DP-T	PCA9514DP-T

Package	Container	PCA9515	PCA9516	PCA9518	P82B715	P82B96
DIP	Tube	-	-	-	P82B715PN	P82B96PN
SO	Tube	PCA9515D	PCA9516D	PCA9518D	P82B715TD	P82B96TD
	T & R	PCA9515D-T	PCA9516D-T	PCA9518D-T	P82B715TD-T	P82B96TD-T
TSSOP	Tube	-	PCA9516PW	PCA9518PW	-	P82B96DP
	T & R	PCA9515DP-T	PCA9516PW-T	PCA9518PW-T	-	P82B96DP-T

Table 1. Ordering Information

Data Sheets and IBIS Models

Data sheets and IBIS models can be downloaded from www.philipslogic.com

PCA9511/12/13/14 TECHNICAL INFORMATION

Block Diagram

The PCA9511/12/13/14 are multi-directional translators that require no external directional control and are specifically designed to support a multi-master I²C/SMBus environment where masters can be located on both sides of the PCA9511/13/14 is shown in Figure 4A and PCA9512 is shown in Figure 4B.

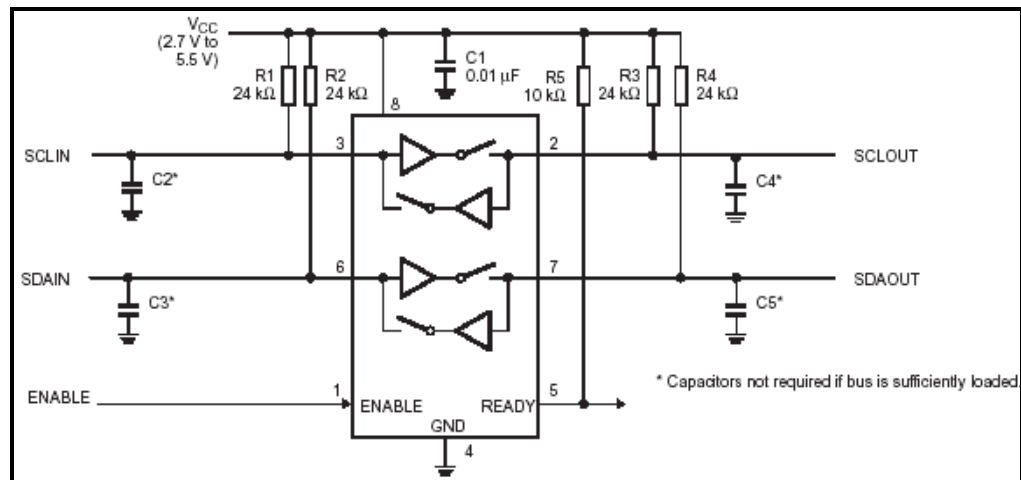


Figure 4A. PCA9511/13/14 Block Diagram

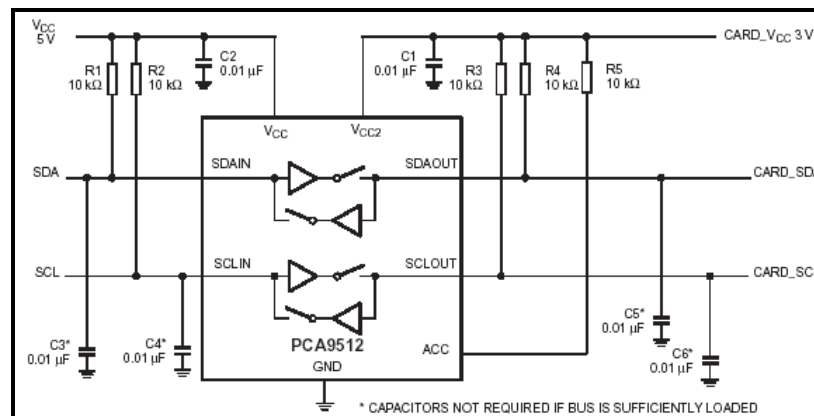


Figure 4B. PCA9512 Block Diagram

There are no external hardware address pins since this device does not have an I²C address. It merely buffers the I²C signals from segment to segment and does not respond to any I²C commands.

The PCA9511/12/13/14 operates at a supply voltage from 2.7 V to 5.5 V with 5.5 V tolerant I/Os. Additional technical information is presented in application note AN10160-01.

Figure 5 shows the PCA9511/12/13/14 used in a typical backplane hot swap application.

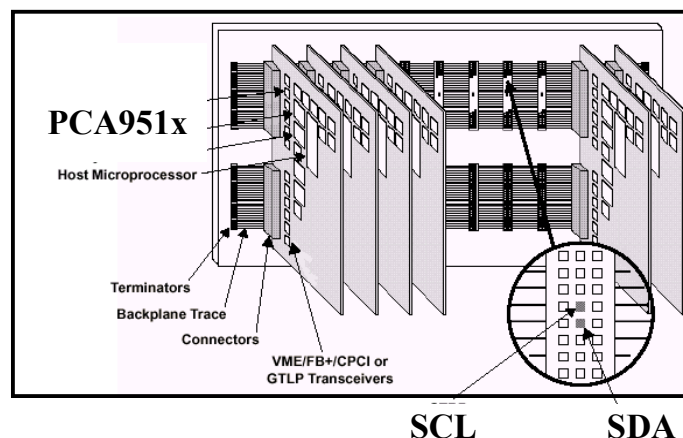


Figure 5. PCA951x Typical Multi-Point Hot Swap Application

PCA9515/16/18 TECHNICAL INFORMATION

Block Diagram

The PCA9515/16/18 are multi-directional translators that require no external directional control. They are specifically designed to support a multi-master I²C/SMBus environment where masters can be located on both sides of the PCA9515 as shown in Figure 4 or any or all of the segments of the PCA9516/18 hubs.

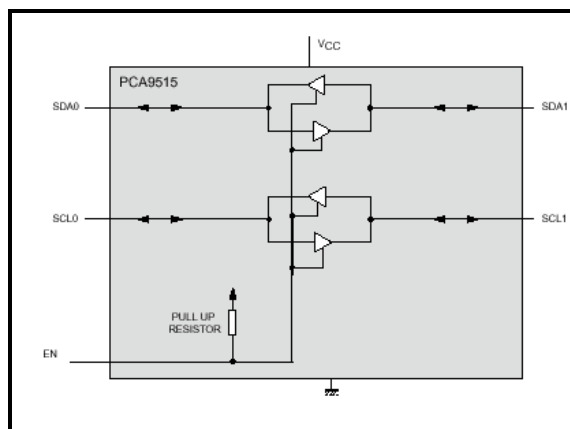


Figure 6. PCA9515 Block Diagram

There are no external hardware address pins since these devices do not have an I²C address. They merely buffer the I²C signals from segment to segment and do not respond to any I²C commands.

The PCA9515/16/18 devices operate at a supply voltage from 3.0 V to 3.6 V, with SDx/SCx and enable pins all tolerant to voltages up to 6.0 V. The pass gates of these devices are constructed such that the V_{DD} pin can be used to limit the maximum high voltage, which will be passed by the devices. This allows the use of different bus voltages on each SCx/SDx pair, so that, for example, 3.3 V devices on one segment can communicate with 5 V devices on a different segment without any additional protection. External pull-up resistors pull the bus up to the desired voltage level for this channel.

The repeater hardware enable pin allows the downstream segments to be isolated and includes an internal pull-up resistor. The enable is held low to isolate the downstream segment to communicate with the upstream channel.

Voltage Level Technical Information

The low-level input threshold is set just above the 0.4 V V_{OL} specifications limit for standard-mode devices. This limits the bus operation to a maximum current of 3 mA.

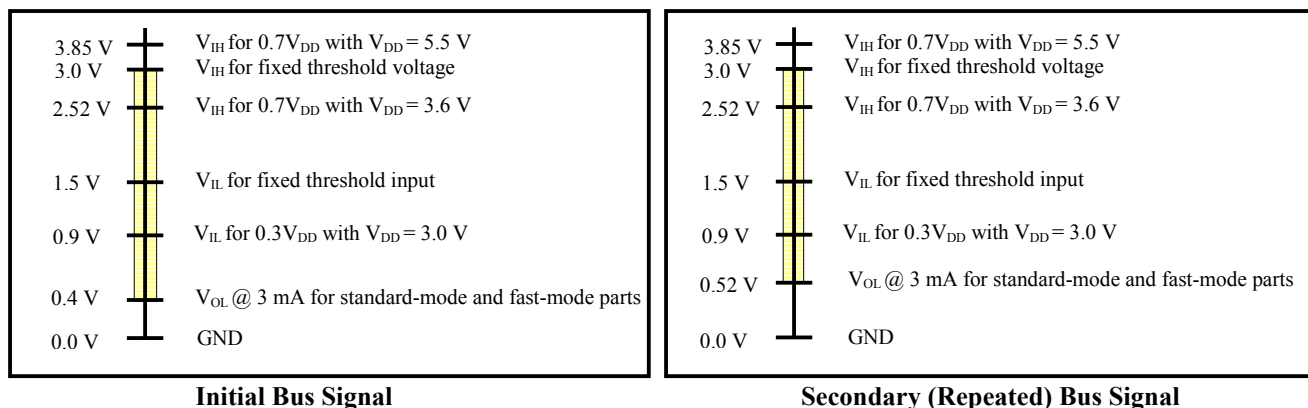


Figure 7. Voltage Level Comparison

Considering a $3.3\text{ V} \pm 0.3\text{ V}$ device with power supply referenced input thresholds, the minimum noise margin for 3 mA operation is 500 mV. In order to both drive the bus low and detect a contention condition or clock stretching event, the device V_{OL} is set at $0.52\text{ V} @ 3\text{ mA}$ as shown in Figure 7. Thus a minimum 380 mV noise margin is preserved when the device is driving low. The low-level detector used to recognize contention conditions has a threshold set at about 450 mV and includes an integrator filter for high frequency noise rejection.

Design Constraints

Maximum Current - The bus can be operated at no more than 3 mA and the bus IR voltage drops must be controlled such that the device sees 0.4 V or less V_{IL} during a contention condition or clock-stretching event. Normally the bus pull-up resistor would need to be at the opposite end of the bus segment from the device. The device bus pins are over voltage tolerant to 5.5 V so that the I²C bus on one or all sides of the device can be operated at 5 V while the device is operated from a 3.3 V supply.

5 V Bus Operation - The bus pull-up resistors need to be tied to the 5 V supply if any 5 V devices are on the bus to guarantee that the V_{IH} of the 5 V device is reached. If no 5 V devices are used, a 3.3 V bus pull-up resistor supply should be used as this will result in a smaller RC time constant for the same 3 mA current.

Pull-Up Resistor Sizing - The device edge rate control on the open drain outputs means falling edge rate is almost independent of capacitive loading up to the 400 pF maximum allowed by the I²C bus specification. Both standard mode and fast mode allow the same 400 pF maximum bus capacitance. The rise time is determined by the RC product of the bus capacitance and the pull-up resistor; for standard-mode the limit is 1000 ns and for fast-mode the limit is 300 ns.

Consider the V_{DD} released input threshold of $V_{IH} = 0.7 V_{DD}$ and $V_{IL} = 0.3 V_{DD}$ for the purposes of RC time constant calculation. Then $V_{(t)} = V_{DD} (1 - 1/e^{-t/RC})$ where t is the time since the charging started and RC is the time constant.

$$\begin{aligned} V_{(t1)} &= 0.3 \times V_{DD} = V_{DD} (1 - 1/e^{-t1/RC}); \text{ then } t1 = 0.3566749 \times RC \\ V_{(t2)} &= 0.7 \times V_{DD} = V_{DD} (1 - 1/e^{-t2/RC}); \text{ then } t2 = 1.2039729 \times RC \\ T &= t2 - t1 = 0.8473 \times RC \end{aligned}$$

For standard-mode I²C bus: t = rise time = 1000 ns so $RC = 1180.2\text{ ns}$.

At a bus load of 400 pF: $R_{\text{max}} = 2.95\text{ k}\Omega$

However, for fast-mode the maximum rise time is 300 ns, so at a bus load of 400 pF, $R_{\text{max}} = 885\ \Omega$.

For a 3 mA bus pull-up at $V_{DD} = 5.0 \pm 0.5\text{ V}$; $R_{\text{min}} = 1.700\text{ k}\Omega$ and at $V_{DD} = 3.3 \pm 0.3\text{ V}$; $R_{\text{min}} = 1.0667\text{ k}\Omega$

Operation at 400 kHz – Fast- mode devices can only be operated at 400 kHz clock frequency if no standard-mode devices are on the bus. The maximum acceptable capacitive load for 400 kHz operation with a 3 mA current is 331 pF at $V_{\text{pull-up}} = 3.3\text{ V} \pm 0.3\text{ V}$, and 208 pF at $V_{\text{pull-up}} = 5.0\text{ V} \pm 0.5\text{ V}$, in order to satisfy the fast-mode rise time requirement.

Rise Time – Rise time must be less than or equal to 1000 ns because the device regenerates the I²C signal, the pull-up voltages on either side need not be equal, i.e., one side could be $5\text{ V} \pm 0.5\text{ V}$ and the other side could be $3.3\text{ V} \pm 0.3\text{ V}$, see figure 4.

Device Segment Enable – Only enable or disable the devices when all buses are in an idle state, to prevent system failures.

Enable = 0: Repeater or Hub is disabled

Enable = 1: Repeater or Hub is enabled

P82B715 TECHNICAL INFORMATION

Block Diagram

The P82B715 is a bi-directional logic driver that increases the allowable total I²C system wiring capacitance and requires no external directional control. It uses unidirectional analog current amplification to increase the current sink capability of I²C devices by a factor 10 and the theoretical load capacitance to 4000 pF. Half of the P82B715 block diagram is shown in Figure 8.

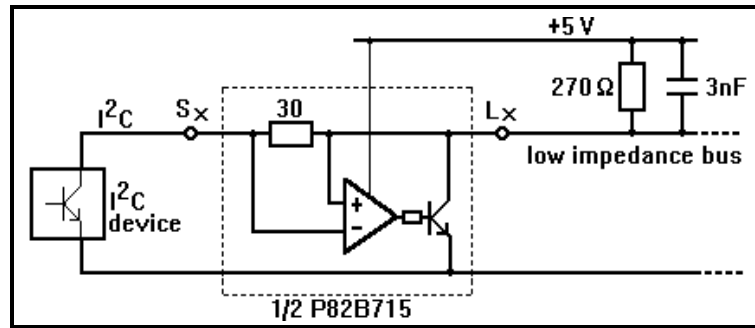


Figure 8. P82B715 Block Diagram

The P82B715 device operates with a supply voltage from 3 V to 12.5 V but the logic signal levels at Sx/Lx are independent of the device's supply. They remain at the levels presented to the device by the attached ICs. The P82B715 cannot modify the bus voltage levels but it does contain internal diodes connected to V_{CC} that will conduct if logic levels exceed the supply voltage by more than 0.5 V. External pull-up resistors pull the connected buses up to the desired voltage level, normally the device supply V_{CC}.

Voltage Level Technical Information

The internal current amplifier is activated whenever the input Sx has a lower voltage level than the output pin Lx. It has a small 'input offset voltage' built-in to ensure system stability. In normal application the static voltage difference between Lx and Sx will not exceed 100 mV. The current sunk on the 'buffered' bus Lx is controlled so it does not exceed 10 times the current being sunk at (i.e. flowing out of) the Sx pin and the maximum 3 mA sink current at Sx causes a difference in bus logic 'low' levels (Lx-Sx) of less than 100 mV. This makes P82B715 fully compatible with all logic signal drives, including TTL. The current amplifier also includes slew rate limiting to ensure system stability the selected slew rate makes P82B715 suitable for Standard-mode systems up to 100 kHz. Signals pass from the 'buffered' Lx side back to the Sx (I²C bus) side simply via a passive 30 Ω path. For the specified maximum 3 mA static sink current of I²C devices this resistance results in a logic voltage drop (Sx-Lx) of typically 90 mV.

When logic level shifting is required, the impedance transforming action allows a single transistor to be used as shown in Figure 9.

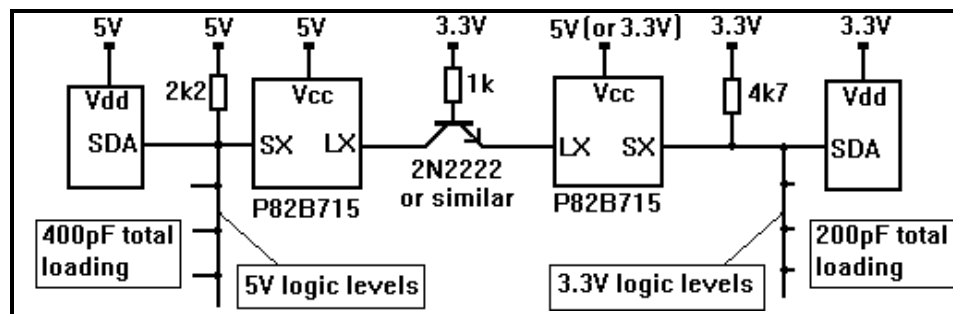


Figure 9. Logic level shifting with P82B715

Design Constraints

5 V Bus Operation – The P82B715 was originally designed and specified for 5 V bus operation. The specification deratings that apply when its supply voltage is reduced below the data sheet value of 4.5 V are given in Appendix 3. The absolute minimum supply is 3 V.

Pull-Up Resistor Sizing - The pull-down current amplification provided by P82B715 allows reducing the pull-up resistors on its Lx bus side by a factor up to 10 times. Any capacitance connected on the Lx bus side is also “scaled” by a factor 1/10, so 1000 pF of loading at Lx only appears as 100 pF loading of the I²C bus at Sx whenever the buffer is active. It is active whenever its Sx pin voltage is lower than its Lx pin voltage. While it is not mandatory to fit pull-up resistors on both sides of P82B715, it is usually best practice to do so. If no pull-up was fitted at Sx, and 200 pF was connected at Sx, then for a rising edge of Lx the amplification will be active and the charging current of the 200 pF will be amplified by a factor 10. This means its ‘effective’ loading on the bus at Lx is 2000 pF! If a pull-up resistor was fitted at Sx, such that the natural rise-time of Sx becomes faster than the rise-time of the Lx side, then the amplifier in P82B715 will not be activated and there will be no slowing of the bus rise-time at Lx. In fact, there will be a tiny improvement because any pull-up at Sx is always connected via 30 Ω to Lx.

A simple rule saves complex calculations:

Use pull-ups on both sides of P82B715 and ensure that the rise-times calculated for each side meets the I²C requirement of 1 μ s. This is met whenever their RC product is less than 1.18 μ s (see paragraph “Pull-Up Resistor Sizing” page 10)

For the Lx side, all the normal I²C impedances may be reduced by a factor up to 10 because its static sink current is 30 mA and the device provides a current gain of 10. The minimum pull-up resistor to 5 V $\pm$ 10% becomes $(5.5 \text{ V} - 0.4 \text{ V} / 30 \text{ mA}) = 170 \text{ ohms}$. But that assumes this is the only pull-up in the system. That is allowed, but we recommend using pull-ups on Sx as well. Because P82B715 does not isolate Lx from Sx, the ‘effective’ pull-up, referenced to any I²C bus (i.e., at Sx) must be calculated to ensure it does not exceed 3 mA.

Example:

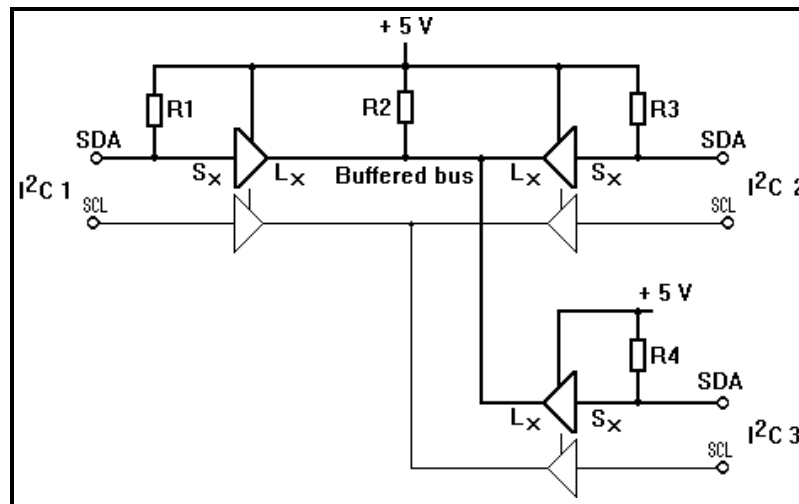


Figure 10. Calculating pull-ups and sink currents
(Note: Capacitors & SCL pull-ups omitted for clarity)

Figure 10 shows three P82B715s connected to a common ‘buffered’ bus. The resistors have been sized to give an RC product of less than 1.18 μ s so the bus rise time requirement will be satisfied.

An I²C device connected at I²C 1 holding the bus low has to sink the current flowing in its local pull-up R1 PLUS, with assistance from P82B715, the currents in R2, R3 and R4. Because the resistors R3 and R4 act to pull I²C 2 and I²C 3 to a voltage higher than the voltage at Lx their buffers will be inactive but these busses will be pulled low via the internal 30 Ω resistors that link Lx to Sx. So the effective current that must be sunk by the buffer on I²C 1 is the sum of the currents in R2, R3 and R4. The current that must be sunk at I²C 1, due to buffer action, is 1/10 of that current.

So the ‘effective’ pull-up, determining the current to be sunk by an I²C device at I²C 1 is R1 in parallel with resistors 10 times the values of R2, R3 and R4. So, if R1 = R3 = R4 = 10 kΩ, and R2 = 1 kΩ, the pull-up load at I²C 1 is $10\text{k}\Omega // 10\text{k}\Omega // 100\text{k}\Omega // 100\text{k}\Omega = 4.55\text{k}\Omega$

The same calculation applies for I²C 2 and I²C 3.

To calculate the current being sunk by the Lx pin of the buffer at I²C 1, note that the current in R1 is being sunk directly by the I²C device at I²C 1. The buffer therefore sinks the currents flowing in R2, R3, and R4. The effective pull-up equals $R2 // R3 // R4$. In this example, it is $1\text{k}\Omega // 10\text{k}\Omega // 10\text{k}\Omega = 833\ \Omega$. For a 5.5 V supply and 0.4 V low, that means the buffer is sinking 16.3 mA. Its rated static limit is 30 mA. So the requirement is that the pull-up on the buffered bus, in parallel with all other pull-ups at Sx pins that Lx is pulling low must be a net resistance greater than the allowed minimum of 170 Ω (for a 5 V system).

P82B96 TECHNICAL INFORMATION

Block Diagram

The P82B96 (shown in Figure 11) is a bi-directional buffer device that requires no external directional control and is intended for application in buses such as I²C, SMBus, DDB and similar derivatives. It provides full isolation of the buses on its two sides, allowing different voltage levels or impedances to be used.

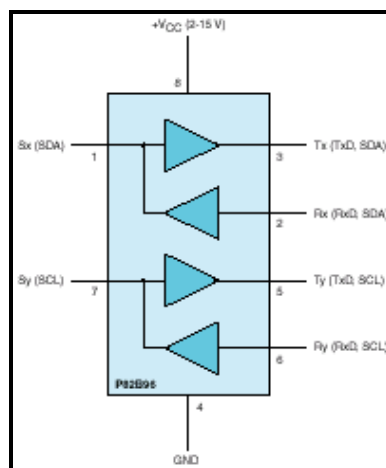


Figure 11. P82B96 Block Diagram

In normal applications, the Sx side will be connected into a standard I²C or SMBus and the logic voltage levels and impedances will conform to the requirements of those systems.

The Tx/Rx side provides the following options:

- Normal I²C bus, with standard CMOS logic $0.5V_{DD}$ threshold levels, with V_{CC} anywhere in the range 2 – 15 V. In this case, it must comply with bus capacitance limits, rise/fall times, and pull-ups must be selected for $I_{OL} < 3\text{ mA}$. In this option, Tx and Rx pins are externally connected.
- Bus with all the special multi-drop, multi-master characteristics of an I²C bus yet it need not conform to the bus impedance, capacitance or rise/fall limits. The designer may set these. In this case the I_{OL} capability of 30 mA at Tx (10x normal I²C) may be used to decrease the bus impedance and allow at least 10 times increase in this bus capacitance (e.g. $> 4\text{ nF}$). The 2 V– 15 V capability may be used to provide improved noise margins when working in a noisy environment. In this option, Tx and Rx pins are externally connected.
- The Tx and Rx signals may be used as individual uni-directional logic interfaces to convert the I²C signals into any bus hardware format, for example to differential signaling over CAN or RS485 buses. These uni-directional signal transmission paths may include opto-isolators or even radio links. Propagation delays in low speed opto-isolators like 4N36 can limit the bus speed to around 5 - 10 kHz but higher speed couplers like 6N137 permit full 100 kHz operation and even higher frequencies are possible.

- d) The Tx open collector output may be amplified (e.g. by adding a PNP transistor) to further increase the I_{OL} to 300 mA or more. This allows working over very long buses. See Appendix 1 for more details.

Voltage Level Technical Information

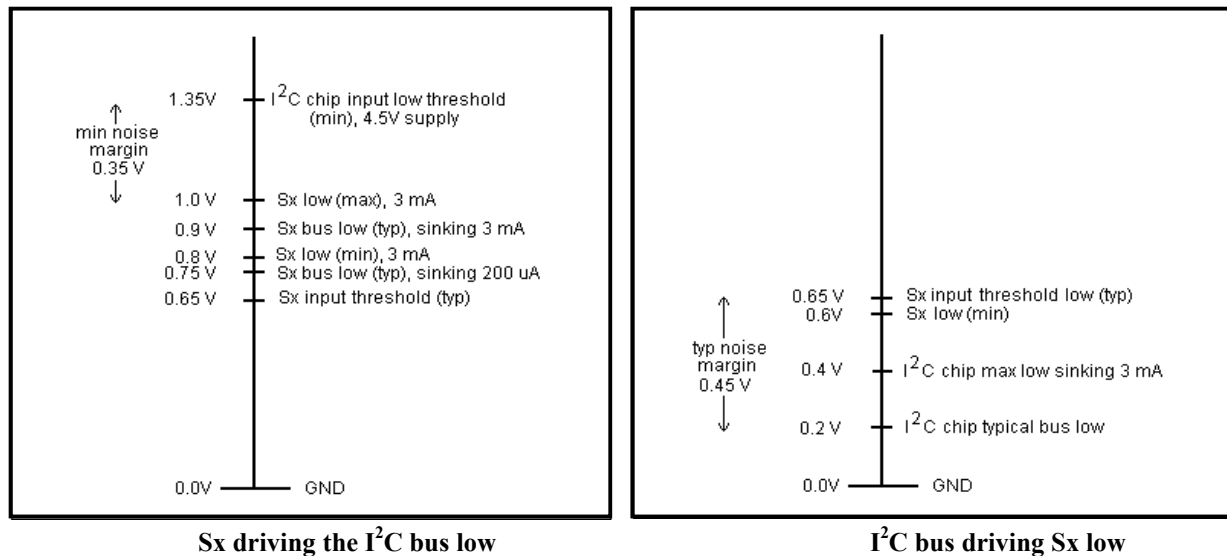


Figure 12. Signal Voltage Levels Compared (at 25 °C)

The low-level input at Sx is set at 0.65 V to give a 0.25 V noise margin above the 0.4 V V_{OL} specifications limit for standard-mode devices sinking the maximum allowed 3 mA. In order to both drive an I²C bus low and detect a contention condition or clock stretching event, the device V_{OL} at Sx is set at 0.9 V (typical) at 3 mA as shown in Figure 12. This provides a minimum noise margin of 0.35 V below the $0.3V_{DD}$ minimum bus input low specification when working on $5\text{ V} \pm 10\%$ supply. The typical noise margin, based on a typical 50% switching threshold, is $2.25\text{ V} - 0.9\text{ V} = 1.35\text{ V}$.

Care is needed when connecting a 3.3 V bus at Sx because the typical noise margin with Sx sinking 3 mA is reduced to $1.5\text{ V} - 0.9\text{ V} = 0.6\text{ V}$ while the guaranteed margin is 0 V. Where possible, 3.3 V (or lower) devices should be connected on the Tx/Rx side that uses standard CMOS logic levels. If connected to Sx, then the pull-up resistors should be set to the maximum allowed by the rise time, or to provide the Sx minimum of 200 μA , whichever needs the lower resistance. Working at 200 μA provides a typical noise margin of $1.5\text{ V} - 0.75\text{ V} = 0.75\text{ V}$ and even 0.25 V of margin compared to a minimum $0.3 V_{DD}$ at 1 V.

The temperature coefficient of the bus low created by Sx and the input logic threshold at Sx are equal, so these levels track each for temperature changes. Their coefficient is -2 mV/K .

Design Constraints

Maximum Current - The I²C bus can be operated at no more than 3 mA and the bus IR voltage drops must be controlled such that a Sx I/O sees 0.65 V or less for a valid V_{IL} . So the Sx input is intended for connection only to devices on the same PCB with relatively low total bus capacitance. Select the pull-up resistor for the lowest pull-up current consistent with bus speed requirements. The minimum pull-up current at Sx is 200 μA . The Tx pin can sink a static current of 30 mA for $V_{OL} = 0.4\text{ V}$ maximum, and typically 100 mA at $V_{OL} = 1\text{ V}$. Both Tx and Sx drives are internally current controlled (limited) at about 5 times their static sink ratings.

Variable Bus Voltage Operation - The bus pull-up resistors at Sx will normally be tied to the supply for the I²C devices connected to it. This I/O has fixed input and output logic levels. The V_{IL} is 0.65 V and V_{OL} is 0.9 V, independent of both the device's V_{CC} and the levels used on the connected bus. Sx may be pulled-up to any required voltage, the limit is 18 V. The logic switching threshold at Rx is related to the devices V_{CC} as for CMOS logic. It switches at 50% of V_{CC} but has tighter tolerances than CMOS. Limits are 42% and 58% of V_{CC} . It is not necessary to use pull-ups to V_{CC} on the

Tx/Rx interface provided it is recognized that V_{CC} sets the logic threshold. So operating with $V_{CC} = 5\text{ V}$ sets the logic switching level at 2.5 V but it is still allowed to connect to a bus operating with 12 V logic as long as the 1.25 V noise margin (low) is acceptable.

Pull-Up Resistor Sizing – The calculations are the same as shown on paragraph “Pull-up Resistor Sizing” page 10 when either side of P82B96 is operated as a standard I²C bus. When the 10x drive capability is used then all resistances can be scaled down by a factor 10 and bus capacitance can be at least 10x greater.

Operation at 400 kHz – Although P82B96 will clock well above 600 kHz it was designed for operation up to 100 kHz because its propagation delays, and those of connected devices and long buses will rarely permit operation to 400 kHz. If timing calculations for a system indicate 400 kHz is within the necessary limits it can of course be used at the higher speeds.

Rise Times – Bus rise and fall times, and capacitance limits, apply to any bus connected to I²C devices. When the Tx/Rx side of P82B96 is being used at non-standard impedance or voltage levels (so only connected to other P82B96 or P82B715 devices) then no such limits apply and the designer is free to set these parameters to suit the application.

USE OF MULTIPLE DEVICES IN THE SAME I²C SYSTEM

Series connection of PCA9515 and/or PCA9516 should not be attempted because the output of one device will not be recognized as a low for arbitration purposes by a second device. For the same reason, series connection between two Sx I/O pins of the P82B96, or the interconnection of a Sx I/O of a P82B96 with a PCA9515 or PCA9516 is not permitted. Multiple PCA9511/12/13/14 Hot Swap Buffers can be used in series.

However, it is only the P82B96 Sx pins using special logic voltage levels that should not be interconnected. In the normal application of both P82B96 and P82B715 it is intended that two devices will USUALLY be connected in series. The P82B715 uses normal I²C voltage levels, and only one side of P82B96 uses special levels. All sides of the PCA9515/16/18 use special levels.

There are no fundamental restrictions on making series connections between any of the following groups:

- A) The Tx/Rx side of P82B96
- B) Either side of the PCA9511/12/13/14
- C) Either side of P82B715
- D) Either side of just one PCA9515/16, OR just one Sx connection of a single P82B96

The selected system impedance levels, voltages, and propagation delays of course need to be considered.

Multiple PCA9515/16 devices are allowed in a system where the devices are not positioned in series and I²C components on separate PCA9515/16 devices do not need to talk to each other. In the example in Figure 13, the bus controller can talk to the power supply, cards or module, one at a time or it could talk to all of them at the same time if the power supply, cards and module didn't ever need to talk with each other or hear different device acknowledgements. The power supply, cards and module can never talk to each other since a low will not be transmitted between two PCA9515/16 devices in series. All cards on the PCA9516 can talk to each other and the bus controller since they are on the same hub.

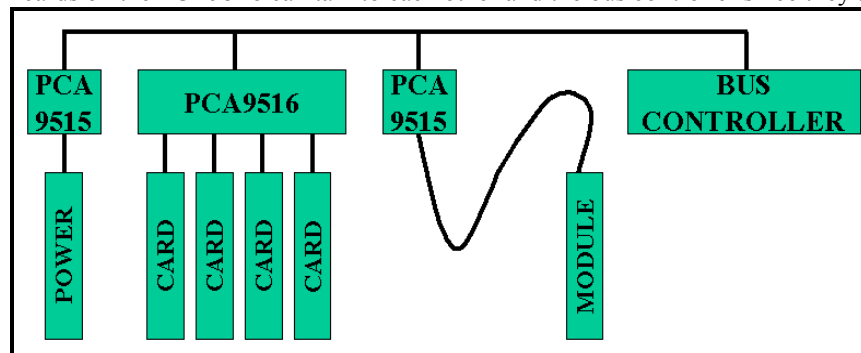


Figure 13. Multiple Parallel PCA9515/16 Devices

The PCA9518 was designed to allow larger systems where all segments need to be able to talk with each other.

Some allowed (blue) and prohibited (red) combinations are shown in Figure 14.

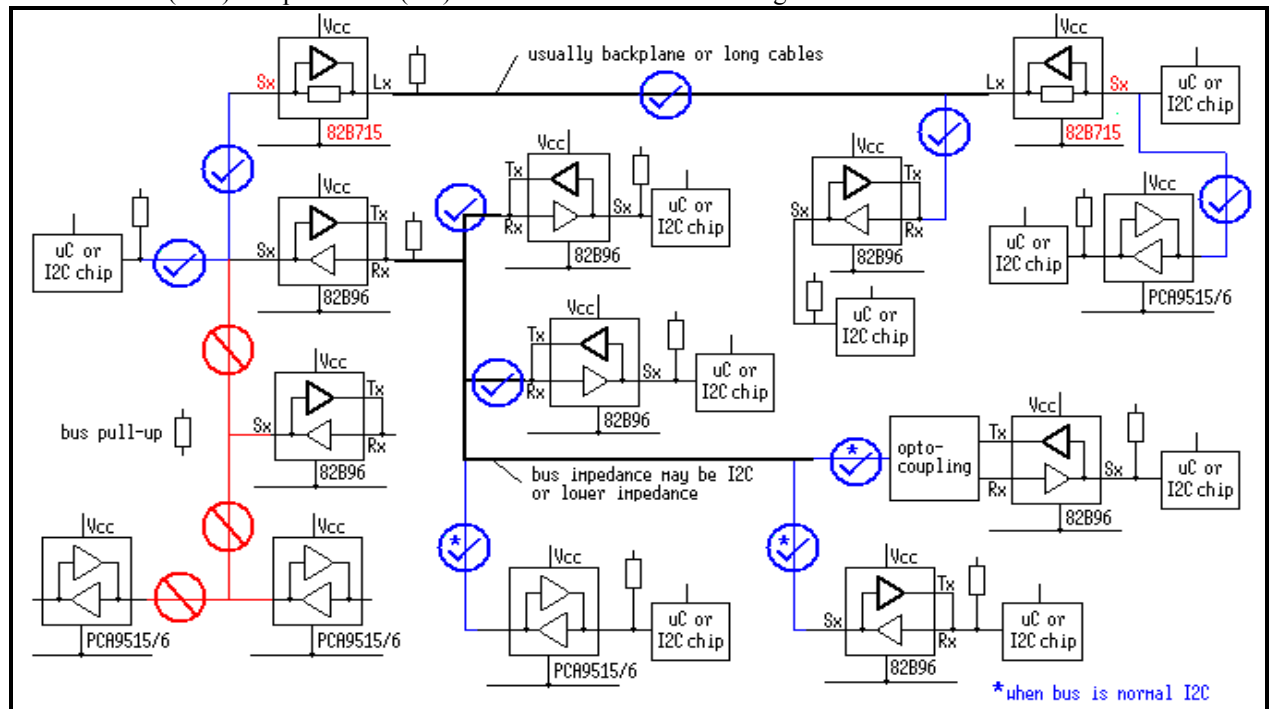


Figure 14. Various Ways to Interconnect Repeaters, Hubs and Extenders

APPLICATIONS

Bus Expansion beyond the 400 pF Limitation

Adding more I²C and SMBus devices on the bus may exceed the 400 pF limitation. The PCA951x/15/16/18 can isolate device and trace capacitance to different segments that are 400 pF or less capacitive loading so that the I²C and SMBus devices can operate properly. A typical PCA9515 application is shown in Figure 16.

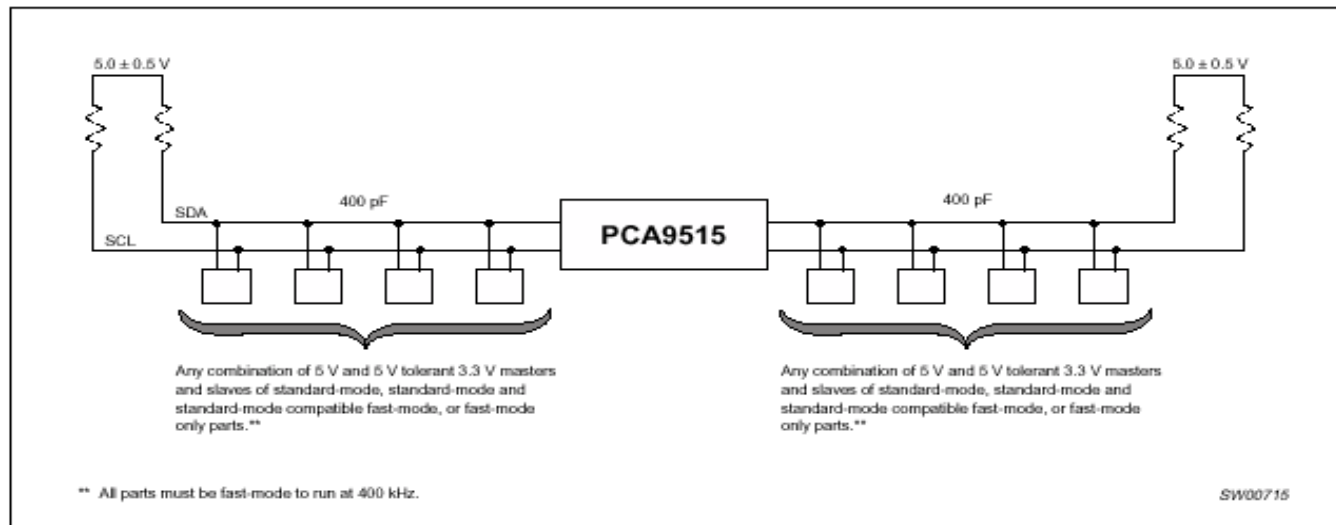


Figure 16. Typical Bus Expansion Application

Split Bus 100 kHz and 400 kHz Application

If all masters are on one side of the PCA9515 repeater and none of the slaves on the other side of the repeater use clock stretching and if all of these slaves are capable of 6 mA I_{OL} , as shown in Figure 17, then the slave side pull-up can be sized for 6 mA because there will be no contention nor clock stretching to be communicated by the PCA9515.

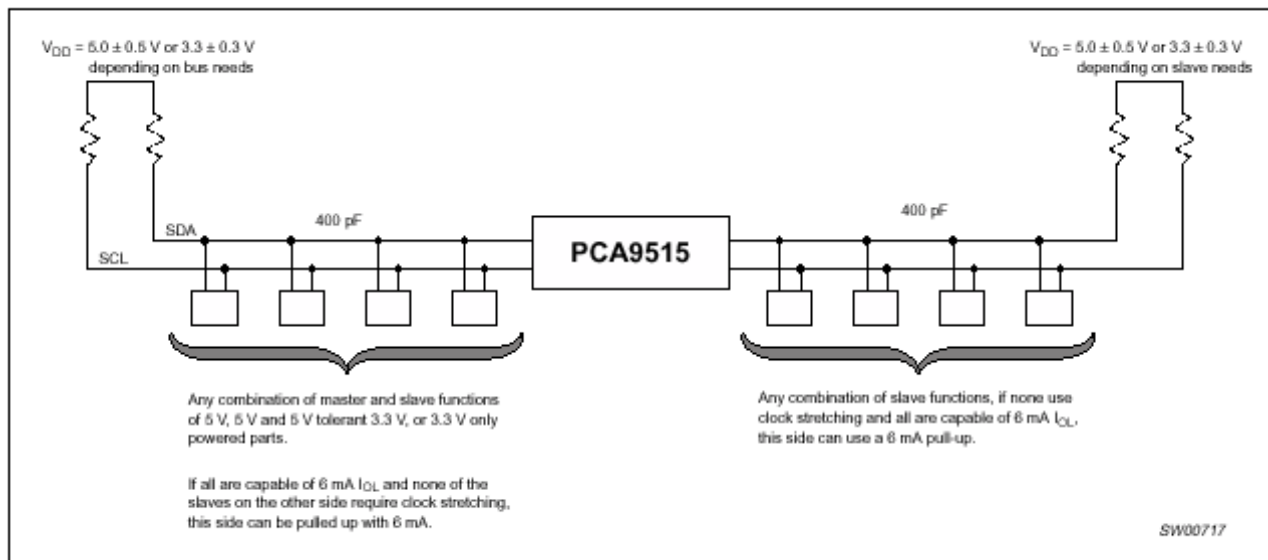


Figure 17. PCA9515 in Split Bus 100 kHz and 400 kHz Application

If all of the masters and slaves on the master side of the PCA9515 are capable of fast-mode operation, none of the slaves on the opposite side of the repeater require clock stretching and the master(s) are able to generate enable and disable changes during the bus idle time. Then the enable feature of the PCA9515 could be used to isolate the master side to allow the master to run at 400 kHz rather than the standard-mode 100 kHz necessary when any standard-mode slave are present on the slave side of the repeater.

If an individual slave is particularly badly behaved during power-up but performs correctly after power-up, or power-up and system reset on power-up, then the PCA9515 enable function could be used to isolate the offending device until it is well behaved and both sides of the repeater are in an idle state. The enable should never be changed, enabled nor disabled, except when both sides of the repeater are in an idle state. If the device being isolated is a slave that does not use clock stretching, then the isolation technique and the standard bus expansion technique can be combined.

Voltage Level Shifting

Many I²C and SMBus devices operate at different voltage levels but need to operate on a common bus. The PCA9515/16/18 allows translations between 3.3 V and 5 V. So, for example, a 5 V I²C master on one segment can communicate with a 3.3 V (non 5 V tolerant) SMBus device on different segment. The channel pass gates are constructed such that the V_{DD} pin can be used to limit the maximum high voltage that will be passed by the device. This allows the use of different bus voltages on each pair, so 3.3 V devices can communicate with 5 V devices without any additional protection as shown in Figure 15. All I/O pins are tolerant to 6.0 V. The PCA9512 with two V_{CCS} can also be used to translate between 3.3 V and 5 V buses. Refer to Philips application note AN10160-01 for more information. The GTL2000/02/10 can also be used to translate between any voltage between 1.0 V and 5.5 V but these devices do not buffer any capacitance. Refer to Philips application note AN10145-01 for more information. P82B96 allows logic level shifting in the range 2V to 15V and can be useful to improve noise immunity on long buses.

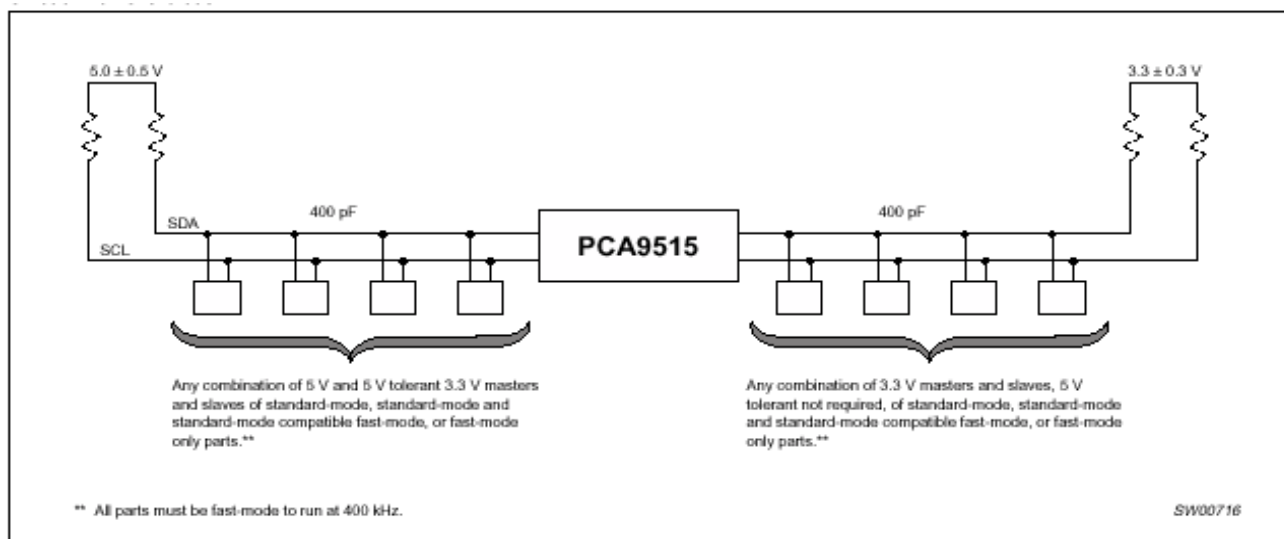


Figure 18. Typical Voltage Translation Application

I²C Multiplexing

Some specialized devices only have one I²C address and sometimes several identical devices are needed in the same system. The PCA9516 and PCA9518 could be used to both isolate capacitive load and to connect one downstream segment at a time to the bus master, where each downstream segment contains identically addressed devices. This would eliminate address conflicts. The PCA954X family of devices can also be used for multiplexing but **they do not regenerate the I²C signal and do not isolate the bus capacitance** to each segment. Refer to Philips application note AN262 for additional information on the Multiplexer and Switch family. In Figure 19, the PCA9516 is used to multiplex between Slaves 1 through 4 where it is assumed that there are identically addressed devices on each slave module.

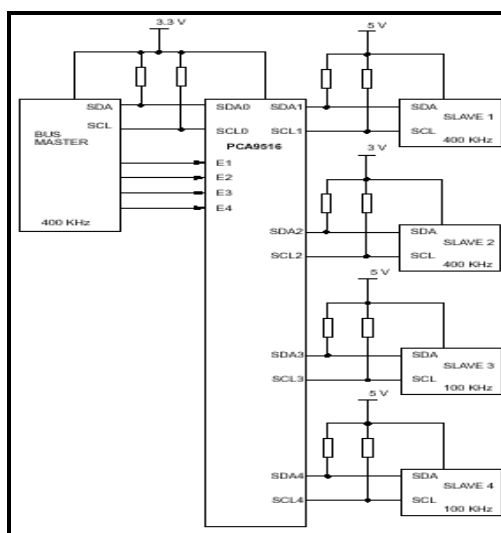


Figure 19. Typical Multiplexing Application

Multiple PCA9516s can be attached in parallel to a common master bus if only one hub's segments will be active at any one time or the hub segments do not need to talk with each other. PCA9518s should be used if more than one hubs segment must be active at the same time and any hub segment to any other hub segment communications is required.

I²C Multi-Point Hot Swap Applications

Use of the I²C bus in backplane systems is increasing and either a point to point arrangement with multiple individual buses to each line card from the master or a single multi-point bus to every line card is used. The PCA9511/12/13/14 are designed for this multi-point application. They have:

- A precharge feature for the SDA and SCL pins to prevent glitching the bus (PCA9511/12)
- An idle detect circuit that will not connect the multi-point bus to the slave devices on the line card until the bus is idle (PCA9511/12/13/14).

In Figure 20, the PCA9511 is used in this multi-point application. Refer to Philips application note AN10160-01 for more information on the PCA9511/12/13/14.

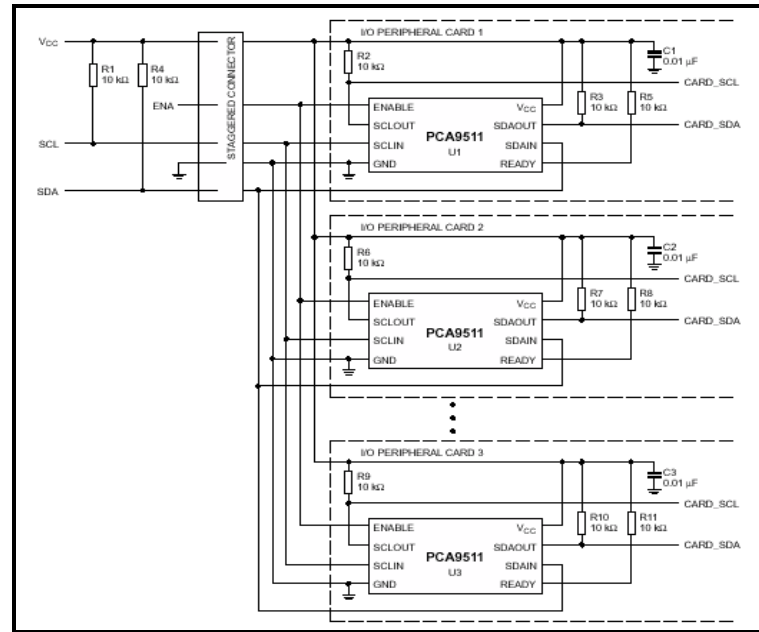


Figure 20. Typical Multi-Point Application

I²C Bus Extension Over Long Cables

The I²C bus is normally used in small localized areas but some applications may require extending the I²C bus further. There are several devices that can be used to extend the I²C bus over cables for a significant distance. The PCA951x, P82B715 and the P82B96. The PCA9511/12/13/14, P82B715/96 can all be used in series. The P82B96 is recommended for this application (see Appendix 1). Figure 21 shows the PCA9511 which can also be used in this application.

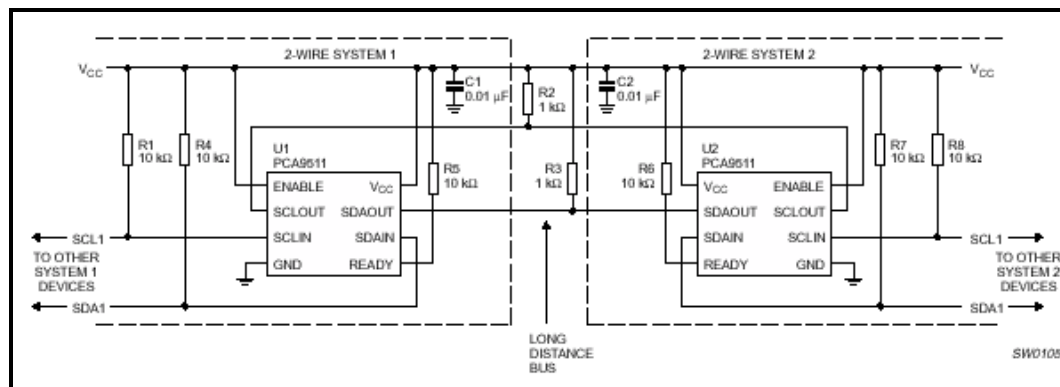


Figure 21. Typical Cable Bus Extension Application using the PCA9511

The P82B715 can be used to extend I²C over a cable and is shown in Figure 22A.

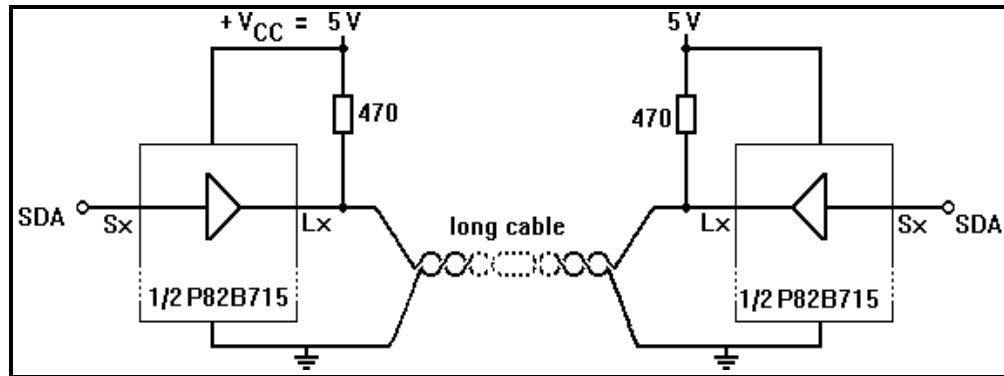


Figure 22A. Typical Cable Bus Extension Application using the P82B715

The P82B96 can be used to level shift and extend I²C over a cable and is shown in Figure 22B. The Schottky diodes help suppress ringing on cables that are not correctly terminated, see Appendix 5, and the Zeners may be required to protect against excessive voltages coupled into very long wiring.

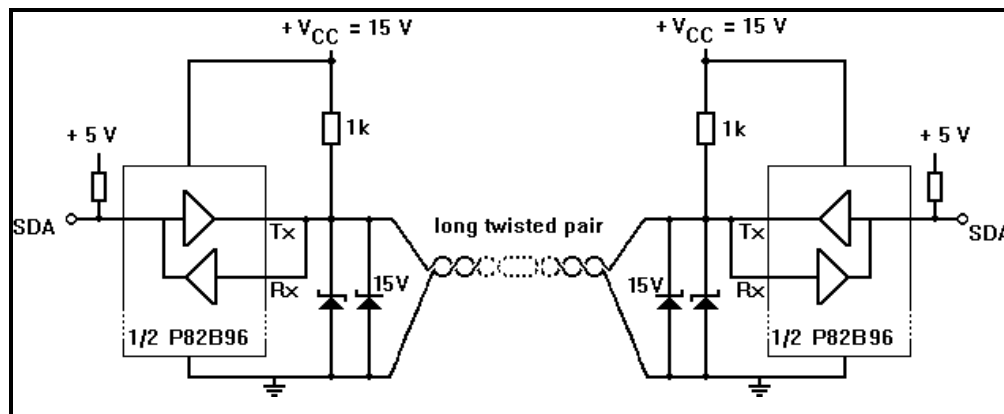


Figure 22B. Typical Cable Bus Extension Application using the P82B96

There are advantages and disadvantages of each buffer device, some of which are outlined in this selection table:

CABLE DRIVE APPLICATION	PCA9515	P82B96		P82B715	
		on Sx	on Tx/Rx	on Sx	on Lx
Max. cable capacitance 3nF approx	400 pF	400 pF	> 4 nF	400 pF	
Bus (cable) logic voltage	3.0 - 3.6 V	3.3 - 15 V	2 - 15 V	3 - 12 V	
Bus logic sink current max.	3 mA	3 mA	30 mA	3 mA	30 mA
Max I ² C clock speed	400 kHz	400 kHz		100 kHz	
Cable length recommended	very short	very short	unlimited	as I ² C to 30 m/100'	
Logic 'low' level	0.4 V	0.5 V	V _{cc} /2	as connected chips	
Typ noise margin, input driven 'low'	0.2 V	0.3V	V _{cc} /2	as connected chips	
Logic 'low' output voltage max	0.6 V	1 V	0.4 V	input level +0.1V	
Typ noise margin output low, 3.3V logic	1.1 V	0.65 V	1.5 V	1.4 V	
Typ noise margin output low, 5V logic	2 V	1.6 V	2.3 V	2.2 V	
Typ noise margin output low, TTL logic	0.8 V	0.4 V	1.2 V	1.1 V	
Logic levels to cables can be shifted	yes	yes		no	
Typ propagation delay	100 ns	300 ns		400 ns	

Table 2. Cable Bus Extension Selection Guide

I²C Opto-Electrical Isolation

The P82B96 can be used for opto-electrical isolation. In Figure 23, the P82B96 is used in the simplest low speed application. More information can be found in Philips application note AN460. A faster circuit, suitable for approximately 50kHz but still using low cost Optos, is shown in Figure 24. A very high speed circuit is shown in Figure 25.

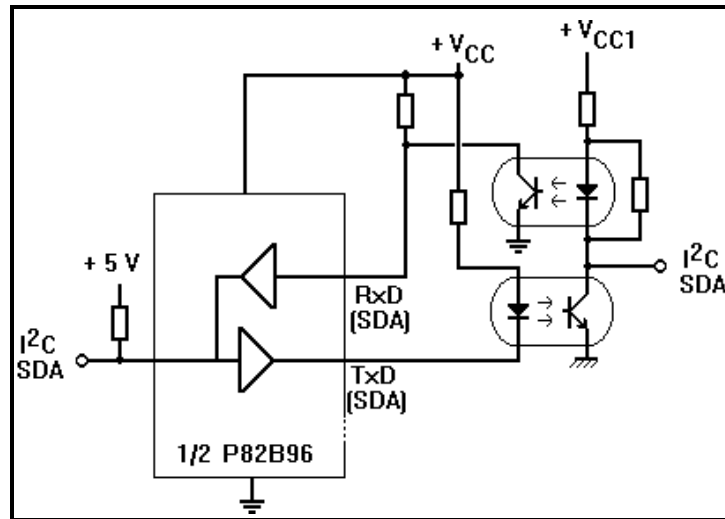


Figure 23. Simple 5 kHz Opto-Electrical Isolation Application

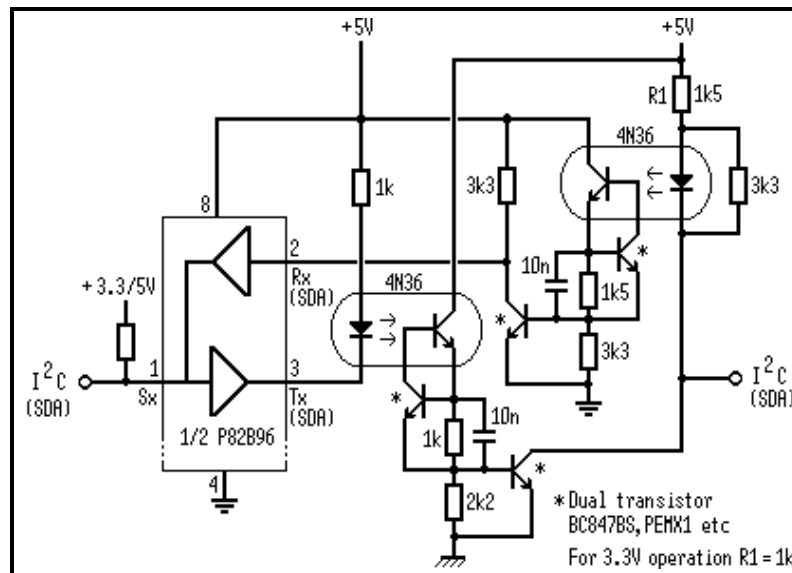


Figure 24. 50 kHz application with general purpose low speed Optos

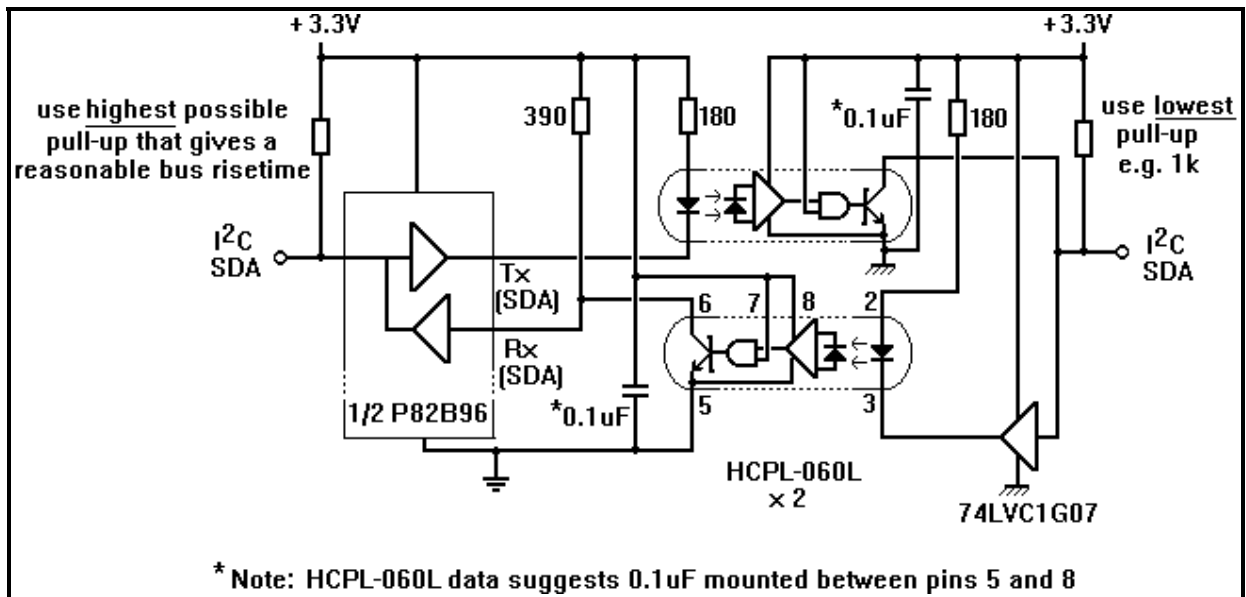


Figure 25. Suggest circuit for highest speed Opto-coupling

With the correct choice of master and slave devices this circuit can be tolerated for up to 400 kHz operation.

Multiple Segment Application

The PCA9518 is similar to the PCA9516 but with four extra open drain signal pins that allow the internal device logic to be interconnected into an unlimited number of segments with only one repeater delay between any two segments as shown in Figure 26. The PCA9518, like the PCA9515/16, is transparent to bus arbitration and contention protocols in a multi-master environment and a master can talk to any other master on any one of the I²C segments. The enable pins isolate four of the five PCA9518 segments. A pull-up resistor needs to be placed on unused segments that do not have a hardware disable pin.

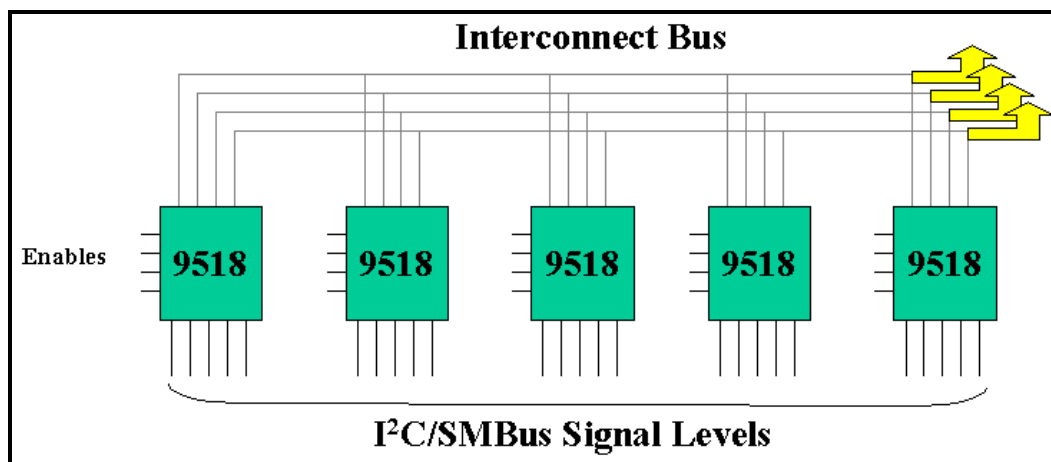


Figure 26. PCA9518 Interconnect Bus

Using the PCA9518 in the 20 card application shown in Figure 27.

- Any submaster can talk to any other submaster on any of the line cards and the main master can talk with any submaster with only one repeater delay.
- An I²C General Purpose I/O Expander with the capability of changing its output states at a STOP command only. This allows the master to enable and disable the PCA9518 segments without confusing devices on the segment becoming isolated or being returned to the main bus. If a normal GPIO is used (i.e., I/O' changes state after the command register is sent, during the Acknowledge sequence) then the devices on the bus would not see the stop

prior to being isolated or would only see the stop or a partial sequence of I²C commands if returned to the bus. The GPIO is controlled by the main system master and can isolate cards 1 through 4 using the enable pins on the PCA9518. Additional GPIO devices can be used to control I²C signals to all the submaster cards

- Another option is for the master to directly control the PCA9518 enable pins but that may require longer trace lengths and multiple GPIO on the microcontroller. This would however ensure that the segments could be individually isolated if for some reason the I²C bus locks up and the remote GPIO is no longer operational.
- Pull-up resistors are required for the 4-wire inter-device bus. Value depends upon the number of devices connected together. The RC constant of the node needs to be taken into account. Maximum current is 12 mA. 500 Ω on each node can then be used.

Note that channel 0 is not used for the card slots. The reason is that if cards are connected to channel 0, they cannot be isolated from the main bus since channel 0 does not have an Enable pin.

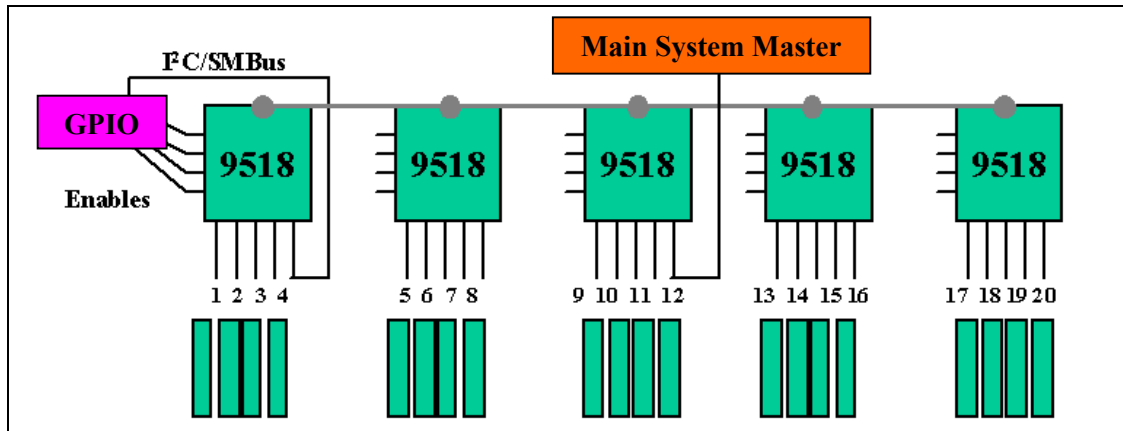


Figure 27. PCA9518 in 20 Slot Application

FREQUENTLY ASKED QUESTIONS

Device usage and differences

1. **Question:** What are the main differences between the PCA9515 and the P82B96?

Answer: The main 'buffering' differences are related to the logic operating voltage levels and drive current capability. In addition, the P82B96 has a capability to split the bi-directional I²C bus into two conventional uni-directional logic signals or to re-combine two such signals to form normal I²C signals again.

The P82B96 is intended for use in systems that will typically include significant propagation delays (long cables or opto-couplers) so it will normally be applied in standard-mode systems but can clock up to 600 kHz. PCA9515 offers much shorter propagation delays and will be the usual choice for 400 kHz systems.

PCA9515 is a symmetrical device with standard I²C sink-current capability while P82B96 has standard I²C drive capability on one side (Sx) but up to 30 mA static sinking (10 times normal level) on the other (Tx) side. The PCA9515 (both sides) and P82B96 (Sx side) use special logic thresholds to prevent bus latch-up. The P82B96 uses perfectly standard half-rail logic levels on its Rx side so that I/O is fully compatible with all logic families and there is no (theoretical) limit to the number of other P82B96 devices allowed to be interconnected in a multi-drop bus configuration.

Repeater, Hub and Bus Extender Selection Summary

FEATURES	PCA9515	PCA9516	P82B96	P82B715
Data sheet supply voltage range V_{CC}	3.0 - 3.6 V	3.0 - 3.6 V	2.0 - 15.0 V	4.5 - 12.0 V
Nominal logic levels supported (range)	V_{CC} to 5.5 V	V_{CC} to 5.5 V	V_{CC} to 15.0 V	Equal/less than V_{CC}
Allows I ² C bus logic level shifting (range)	3.0 - 5.5 V	3.0 - 5.5 V	2.0 - 15.0 V	no level shifts
Allows interconnecting I ² C buses, each 400 pF	yes, 2	yes, 5	yes, 2	no
On-chip bus sink current capability	I ² C	I ² C	10x I ² C	10x I ² C
Drives lower impedance "I ² C like" buses	no	no	yes	yes
Max. (multimaster) bus capacitance supported	800 pF	2000 pF	unlimited	3000 pF approx
Allows inter-working of I ² C and SMBus	yes	yes	yes	no
Designed operating I ² C clock speed	400 kHz	400 kHz	400 kHz	100 kHz
Typ. propagation delay (excluding contention)	100 nsec	120 nsec	300 nsec	400 nsec
(Multimaster) system configuration	repeater	hub/star	multi-drop bus	multi-drop bus
Splits I ² C to Tx/Rx allowing opto-isolation	no	no	yes	no
Releases all I/Os if V_{CC} supply fails	yes	yes	yes	no
I/Os can be pulled above chip's V_{CC} level	yes, to 5.5 V	yes, to 5.5 V	yes, to 15.0 V	no
Logic "buffer enable" input(s)	yes	yes, 4	no	no
Supply current (typical)	2 mA	7 mA	1 mA	16 mA
Packages	TSSOP/SO8	TSSOP/SO16	DIL/SO8	DIL/SO8

Table 3. Repeater, Hub and Bus Extender Selection Summary

2. **Question:** Can the PCA9515/16 be used in series?

Answer: No, two of the PCA9515/16/18 cannot be used in series, or one of these connected to the Sx I/O pin of the P82B96. The reason is because of the way these devices sense whether a low on the line is being driven externally (e.g. by a device on the bus connected to that segment) to the device or internal (e.g. by a device on a bus connected to the opposite side of the device) to the device. A low on one end of the bus will not be seen as a low at the other end of the bus if it must travel through more than one repeater/hub/extender. The Philips PCA9511/12/13/ are being designed to be used in series. Normally two P82B715s can be used in series because they do not use special logic levels. It is allowable to connect the PCA9515/16 to the Tx/Rx side of the P82B96 or to either side of the P82B715.

3. **Question:** We have an application for 30 identically addresses I²C slaves, each 2 meters from the bus master and we need a bus wiring solution.

Answer: Possible solutions include:

- Use four of the 8-channel PCA9548 switch in parallel and activate only one downstream channel at a time. Some concerns would be the capacitive loading of the 2 meter wiring exceeding the 400 pF limitation, but at 80 pF/m, it should not happen.
The PCA9548 uses I²C commands to activate and deactivate the downstream channels. The channel does not shift until the STOP command is sent by the master and the switch acts only as a wire and does not regenerate the signal.
- Use the P82B715 or P82B96 in a multi-point arrangement that can help extend the bus longer distances (e.g., 30 meters depending on the transmission means). The P82B96 has a split output port bus buffer that can also be used for interfacing to other cable drivers allowing differential transmission of the I²C signal, optically isolating the I²C bus or just increasing the bus voltage to improve the noise margin.
- Use eight of the 5-channel PCA9516 in parallel. The bus master would connect to the SD0/SC0 segment of all eight PCA9516s and then the master would use its GPIO (30 GPIOs are needed) to enable and disable the required downstream segment one at a time. Each segment isolates capacitance from the other segments so 400 pF is allowed on each 2 meter trace. **The PCA9516 uses hardware enable pins to activate and deactivate the downstream channels.** More than one segment can be enabled at the same time but the limitations are that although a segment on hub A can talk multi-directionally with any other segment on hub A, it will not be able to send a low signal to downstream segments on hub B, C, D, etc. Since in this application only one segment can be enabled at a time, each segment has slave devices with the same I²C address, this will not be a problem. The 4-channel PCA9516 hub does the selection via hardware pins and the bus needs to be idle before adding or removing channels. The hub also allows 400 pF per channel since the signal is regenerate by the hub.

Bottom Line - The PCA9516 would be the better solution since all downstream segments can be physically isolated should the bus hang up and it isolates the capacitive loading to each segment. The PCA9548 however has a hardware reset pin that opens all channels if the bus should hang up so the master can regain control. The PCA9548 should be considered if reduced cost is important (only 1/2 as many devices are needed) and it is desired to use I²C commands to select channels since 30 extra GPIOs are not available on the bus master. You can't use standard I²C GPIO slaves to expand your master I/O to enable and disable the PCA9516 since GPIO changes state on acknowledge, before the slave would see the stop command, and the slave would become confused. A GPIO changing the I/O state at a STOP command is required.

4. **Question:** We are passing signals from a PCA9515 through the PCA9544. On the I²C bus lines, they are using a 1.3 k Ω pull-up to 3.3 V on the PCA9515 side of the pass transistor and a 20 k Ω pull up to 3.3 V on the other. We see a PCA9515 Vol = 0.5 V to 0.6 V and on the other side of the pass transistor a signal level of 0.7 V to 0.8 V. This makes the combination unsuitable for use with a Linear Tech ® LTC4300-1 (which has a max Vil spec. of 0.6 V and which is used on the other side of the pass transistor). Does Philips have a plot of Vol vs Iol for the PCA9515 - if we're at 0.6 V @ 2.7 mA, we will be out of data sheet specification at 6 mA. Using the 20 k Ω ohm resistors, they appear to be limiting the amount of current through the pass transistor in order to reduce the drop across it. From the calculations, they have about 165 μ A of pull-up current through the pass transistor. To get a 200 mV delta Ron is quite considerable. Do we see any Ron dependency with through current?

Answer: You are suffering from one of the unintended outcomes of using the LTC4300-1 device.

- The PCA9515 uses an output driver that is essentially a voltage source that only sinks current and its level is set at 0.52 V. Thus, from zero to more than 10 mA, the output voltage only changes by a few millivolts. Even with variations over process, temperature, and power supply, the voltage is within a few tens of millivolts of 0.52 V. The output voltage at even 10 mA is less than 0.6 V.
- The maximum Ron for the PCA9544 is 55 Ω . Assuming 135 μ A from the 20 k Ω resistor, the voltage drop across the PCA9544 should be less than 0.01 V. Even at 55 Ω Ron, which is the maximum limit, to get a 0.2 V drop across the PCA9544 would require 3.6 mA.

Therefore, the LTC4300-1 is the most likely source of the extra current. I believe a check of the system with the LTC4300-1 removed will confirm that the voltage driven at the output of the PCA9544 is less than 0.6 V. The problem is that the "pull-up" current one-shot in the LTC4300-1 is being triggered by noise and is contending with the low level driven by the PCA9515. The pull-up current source in the LTC4300-1 speeds the low to high transition by increasing the total pull-up current and is switched off when the pin is high so as not to contend on the next high to low transition. If the current source of the LTC4300-1 gets turned on when the PCA9515 is driving a low, it will remain on because the pin does not go high. The current supplied by the current source will be high enough to pull the voltage up to 0.7 V - 0.8 V.

Note: The LTC4300-1 is 100% equivalent to the Philips PCA9511 and the LTC4300-2 is 100% equivalent to the Philips PCA9512.

5. **Question:** We are having problems with the PCA9516 and want to replace it with an analog multiplexer solution. Can you share any information to justify the need for these hub devices and discuss in more detail how arbitration is used (beyond what is in the data sheet)?

Answer: The PCA9515/16 provide the principal benefit of allowing the system to be increased in size well beyond the 400 pF limit in the I²C specification by physically separating the system into branches. Each branch can be loaded up to 400 pF. The enables can be used to isolate faulty branches, or resolve address conflicts, however to prevent system problems the enable pins should only be switched during an idle state. Acknowledge and arbitration is fundamental aspects of I²C protocol. Arbitration only applies to multiple master systems however acknowledge is used between master and slave to confirm byte transactions. If you only need to multiplex the lines and loading is not a problem, we offer a family of I²C controlled multiplexers and switches under the PCA954X series.

Live insertion

1. **Question:** Can the PCA9515 be used in a Hot Insertion application? I'm worried about when the main system power is off and the PCA9515 is not energized. What happens when one of the cards is plugged in? Could the active I²C signal voltage damage the device?

Answer: When the PCA9515/16/18 are powered off, the SCL and SDA pins are in a high impedance state so a voltage being applied to them will not damage the devices. The PCA9511/12/13/14 are specially designed for hot swap applications.

2. **Question:** Is it possible to pull up some of the hub downstream I²C channels to a permanent supply line and some others to a switched supply lines?

Answer: When the hub is in operation and the channel is enabled, the channel termination voltage must be on to pull the I²C bus high. If the channel is disabled, then the channel termination voltage can be turned off.

3. **Question:** If the PCA9516 is powered up and a channel enabled what will happen if I remove the channel's pull-up voltage?
Answer: If the power supply is turned off, it will almost certainly pull low. If the power supply is removed, then the leakage current will determine what happens. If the leakage to ground dominates, it will go low and if the leakage to a high dominates it will stay high. (High is an unlikely outcome especially if no power is present).
4. **Question:** In the theory of operation you refer to a pull-down resistor - if I remove the pull-up voltage will I in fact pull down to GND and lock every channel up?
Answer: Most likely, if the channel is enabled and drifts low because the pull up supply is removed it is recognized as a low and will be repeated on all enabled channels for as long as the condition exists or until the channel is disabled, whichever comes first.
5. **Question:** What is the behavior of the PCA9516 hub when the switched supply is switched off?
Answer: When power is cut to the PCA9516 hub, the device outputs go to a high impedance state and no communication through the hub is possible. When in a high impedance state, the outputs will not affect operation of the segment and normal I²C communications are possible on that segment, however, none of the 5 segments will be able to talk to each other. However care must be taken if segments are conducting independent conversations when reconnecting or restoring power. Discussion of the software routines to cover all the possibilities during power failures or hot-plugging is beyond the scope of this application note and will be covered separately.
6. **Question:** Does the PCA9516 Hub power supply need to be put on the same supply that powers the microcontroller (permanent supply) or a switched supply?
Answer: It does not matter to the PCA9516 hub. If it is the permanent power supply then the device is always active and the downstream channel need to be pulled high or disabled. If switched off, all outputs go to a high impedance state and no communication is possible through the hub.
7. **Question:** What about Fault Tolerance? How do the devices fail? Would it be possible to tri-state the I²C bus interface in case there is a problem and makes the bus hanging up? This would make our system better and will give them a better Fault Tolerance.
Answer: Modern ICs have extremely low failure rates so failures will almost always be due to overstress. Overstress can be prevented by applying protection measures where the likelihood of overstress justifies this extra cost. Devices like P82B96 have very high typical breakdown voltages, over 25 V on the Sx and almost 50 V on Tx pins so it is relatively easy to apply protection clamps set at around the data sheet limit voltage (15 V – 18 V) that will not influence normal operation.

The P82B96 is relatively new and has no reported failures. It is closely related in technology to P82B715. In selling that device for 7 years I think the Quality and Failure Analysis lab has only ever seen about 4 defective products returned. All were at the same time and due to an unusual customer application that was not protected in our testing or data. We calculated the chance of other faulty devices at 1 in 10,000 and even that was restricted to production in a short time interval (due to a mask defect) it cannot happen again because we have added factory tests for it. I am not aware of any field returns for any cause in the last 4 years (since that test was added), or in the earlier period.

The PCA9511/12/13/14/15/16/18 and P82B96 are designed to release their I/Os if their power supply fails, so the bus is available for other devices to communicate. The PCA9515 Repeater and PCA9516/18 Hubs also features an Enable pin that provides the tri-state function. So several devices are offered that can be used to isolate failure problems PROVIDED the problems can be software-diagnosed -- for example by isolating sections of the bus in turn and testing for shorts until the faulty section is found.

It is also possible to include simple hardware circuits that check the state of the bus lines and cause disconnection of sections if the bus stays low for longer than a designed period. Some SMBus devices integrate this function. An example of such hardware is shown in Figure 28.

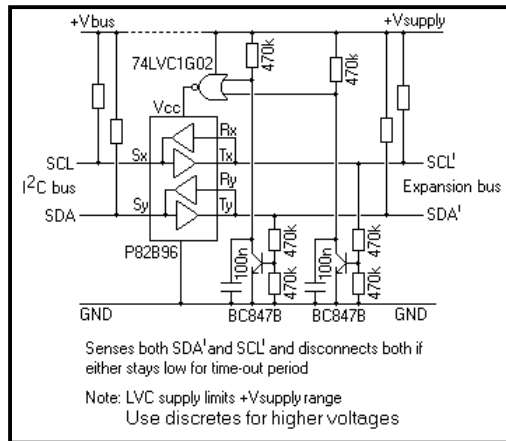


Figure 28. Bus “stuck low” sensor schematic

I²C over long distances

1. **Question:** The PCA9515 is being used in the network between the server and peripherals.

- Is there any rule for the distance of I²C Bus?
- Is there any method to improve signal quality in the I²C Bus in the cable length of 1m?

Answer: There is no maximum length specified in the I²C specification. The limit of the length is a function of several factors including capacitance (usual maximum limit is 400 pF), the minimum value for Rp, propagation delays along the cables, the type of cable used and the integrity of the logic signals in the presence of noise. There are bus rise and fall times that must be met with limits set by the static and dynamic sink capability of standard I²C devices (3 mA sink current etc). There are some ways to improve the rise time by using a switched pull up circuit. At 400 kHz data rates, there is not any problem with transit time for just 1 meter of cabling. This assumes the question refers to cable propagation delay, then there's no 'problem' with PCA9515 (100 ns each way) and no problem with 1 meter (10 ns round trip) and a bus 'low' time of 1.3 μs. Appendix 1 covers these complex calculations.

2. **Question:** There is an old P82B715 App note about a “mile long” I²C. Is that practical?

Answer: No, certainly not for P82B715. In real applications it is fine (conservatively and well proven) to about 100 feet at 5 V at 100 kHz. It was explained in the application note that the result was obtained only under perfect lab conditions, and with low clock speeds. Signal propagation delays over a mile are around 17 microseconds meaning the clock speed certainly cannot exceed about 25 kHz. If a mile is really needed then it could be achieved using the P82B96 with the circuit in Appendix 1 provided the necessary protection against things like lightning-induced currents are included.

3. **Question:** We really want to simplify the board design and wiring by using an I²C bus and would rather use the P82B715 than the P82B96 because the number of cables will be reduced from 6 to 4. We want to use USB cables and USB connectors as the physical connectors for the P82B715 because they are easy to obtain and the price is very low.

Answer: When distributing I²C signals using 'buffer' ICs like P82B96 or P82B715 the signals that are distributed are usually just the same ones as for I²C. We just use 4 wires. So we 'buffer' the SCL and the SDA signals, but we still only require one signal wire for each of these signals. It is helpful to send the two logic signals (SDA and SCL) with a return (ground) wire for each. So we would like to send (SCL plus Ground) and (SDA plus Ground). That takes 4 wires to send the I²C signals to each sub-sub board. It is also allowed to substitute a voltage supply wire for one of the ground signals, because the supply will also be bypassed to ground.

So then you would use two pairs of wires: (SCL with ground) and (SDA with the positive supply wire). That is 4 wires in total. This is usually exactly the same for both P82B96 and P82B715. The pins Tx and Rx will be connected together and pins Ty and Ry will be connected together. Then P82B96 buffer I/O connections become: Input at Sx, output is Tx linked to Rx. Input at Sy, output at Ty linked to Ry.

Figure 29 below shows how I²C signals would be distributed to all the sub/sub boards with the P82B96. Notice there are only 4 wires in the suggested "expansion bus" that distributes the I²C signals. In this example this "distribution bus" is using 12 V logic signals so that it will be very immune to noise. It is not necessary to use 12 V logic, it will also work with 5 V or any other voltage that is convenient to use.

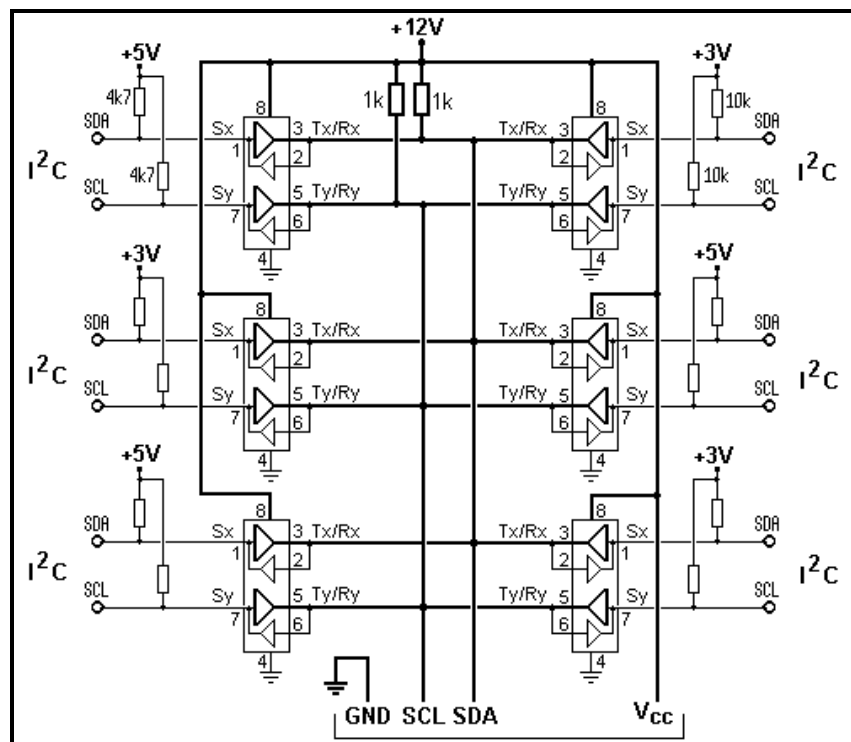


Figure 29. Distribute I²C using USB cables (1)

In this diagram, ALL the I²C connections have FULL multi-master status. Any one of them could be your sub-board input, with the control Micro, and all the others (operating at 5 V or 3.3 V logic levels) can be your sub-sub board connections. USB cables are very convenient for distributing I²C. Just take care to use the USB D+/D- pair for sending one of the I²C signals (for example SCL plus V_{CC}) and the USB power pair for sending the other signal (SDA plus Ground). Please do NOT use the conventional supply and ground connections as used for USB. That would mean you would send SDA and SCL over the D+ and D- wires and that is NOT ideal. It can work, if the customer insists to use it that wrong way, but it will cause coupling of the SCL signal into the SDA signal that is not a problem with the differential USB D+/D- signals.

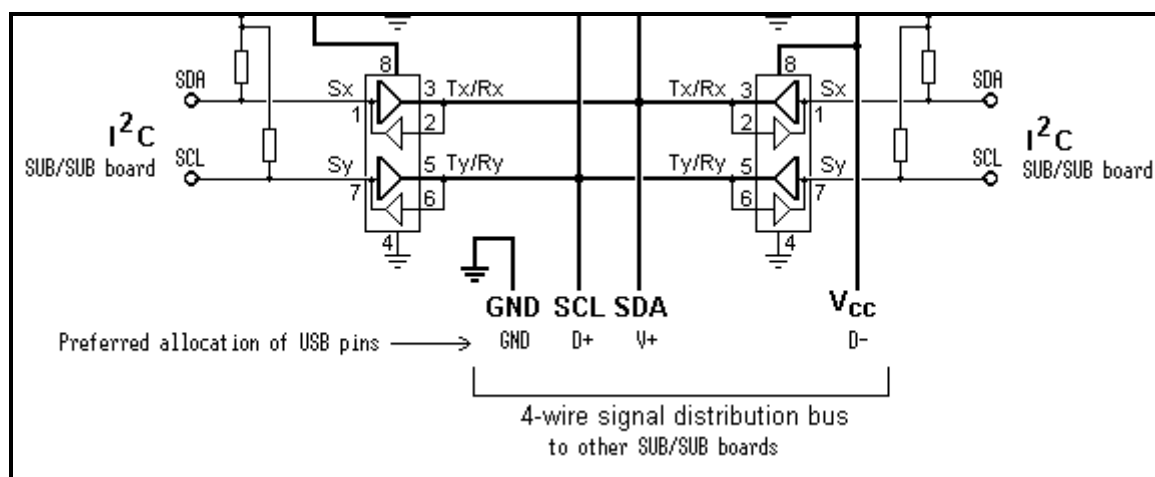


Figure 30. Distribute I²C using USB cables (2)

Figure 30 is another drawing showing one proposal for using USB cabling. Because P82B96 can change the logic signal levels to suit 3.3 V logic or 5 V logic on the sub/sub boards, and also use even higher voltage logic on the "USB" cables for noise immunity, it is preferred for this application.

4. **Question:** What is the maximum length of the cable that can be extended by the P82B715 or P82B96?
Answer: Details can be found in Appendix 1 but a simple answer is that the maximum length depends on the SPEED that the bus operates and the type of cable that is used. For P82B715, we say that the limit is about 50 meters and we recommend a clock speed slightly lower than 100 kHz is used.

In example 2) in Appendix 1, we show that 85 kHz is a very safe speed to operate with 50 meters of cable. For P82B96, there is no maximum cable length because the bus capacitance limit does not apply and it is possible to add extra transistors to drive very long cables that are correctly terminated with their characteristic impedance. In example 3) in Appendix 1, we calculate that a cable length of 1 kilometer can be operated with a clock speed of 31 kHz and a cable length of 100 meters can be safely operated at a speed of 71 kHz. Those calculations contain safety factors and the distances and speeds can be slightly higher. If the customer can tell us the actual I²C device numbers and the type of cables that will be used then we can calculate the limits that apply to their system.

5. **Question:** On a PCB, there are several I²C devices connected to one I²C bus. I want to extend the I²C bus by connecting two further I²C devices using a ribbon cable. Because of the capacity of the ribbon cable (about 50 pF/m) I want to separate it with a PCA9515. The pull-up resistors for the SDA and SCL lines are on the PCB.

PCB with I²C bus ↔ PCA9515 ↔ pull-up resistors ↔ connector ↔ ribbon cable ↔ first I²C device ↔ ribbon cable ↔ second I²C device

According to the I²C specs the low level output voltage V_{OL} for an I²C device is 0.4 V @ 3 mA sink current. The V_{OL} of the PCA9515 is typically 0.5 V, but to recognize an Acknowledge sent by an I²C device, the voltage level of SDA has to be 0.4 V max. This is OK for short distances between PCA9515 and I²C device. But what about the voltage drop caused by the connector and the ribbon cable? Further, the I²C specifications recommend a series resistor R_s for the SDA / SCL input of the I²C device to protect it against high voltage spikes. There's also a voltage drop (e.g. $100 \times 3 \text{ mA} = 300 \text{ mV} = 0.3 \text{ V}$) that is too much / not allowed for the PCA9515. Conclusion: I cannot use the PCA9515 in my application. Is this assumption correct?

Answer: You are correct when you conclude that it is not good practice to try to signal over 'long' cable with the special logic levels that we need to use to prevent bi-directional I²C buffer ICs from latching like what is used on the PCA9515/16/18. That is why the P82B96 has those special logic levels on only ONE side and the other side uses perfectly conventional (CMOS style) logic levels with that logic threshold set at half the device's supply (V_{CC}). You are free to choose any voltage you wish for V_{CC} on P82B96 (2 V to 15 V) and this does not change the logic interface signal levels on the special logic interface side. So you are able to interface to 5 V logic or, with a little care, even 3.3 V logic. As you say, those special levels are OK for short distances (e.g., on one circuit board). The P82B715 uses an 'analog' amplification approach to permit inclusion of more cabling. It effectively reduces the system capacitance contribution of the cable to 1/10 its actual capacitance. It does not change the logic voltages.

6. **Question:** For our application, we want to send 3.3V I²C from our microcontroller over long cables to several slave boards. How do we choose between P82B96 and P82B715?

Answer:

1) The main microcontroller has 3.3 V logic. If 3.3 V logic is sent over cables in a noisy environment, it is easy for the noise to cause spurious bus signals. The interference only needs to be about 1.0 V to 1.5 V **to cause wrong signals**. We propose to feed that 3.3 V logic into P82B96 and change it to 12 V logic signals. Then interfering noise signals must cause 6 V signals on the cables before they can cause any wrong logic signals. If P82B715 was used, the logic signals must remain at 3.3 V, because P82B715 cannot change them. So the performance for noise will be about 5 times worse than when using P82B96. When high immunity to noise is required, the P82B96 gives that performance. With the P82B96, it is allowed to have different logic levels on the slave boards. If some I²C devices use 5 V logic then P82B96 will change the 12 V logic on the cable to 5 V logic for those devices. With the P82B715, all devices in the system must use the same logic levels (or some extra circuits must be provided to change the logic levels).

2) If the P82B715 is used, then the total system capacitance (after a special calculation for the cables) cannot be more than the specification limit of 400 pF. The designer must add all the capacitance on every slave board and then add all the capacitance of all the cables. The total of all the capacitance (for all the boards), PLUS the total cable capacitance divided by 10, must be less than 400 pF. That will put limits on the number of slave boards that can be connected into one system. It also requires calculations. When the P82B96 is used, the only limitation is that each separate slave board (alone) must not exceed 400 pF. There is NO LIMIT on the capacitance of the wiring and NO LIMIT on the total system capacitance. It is allowed to connect any number of slave boards. The system calculations are much easier and changing the system is easy.

Electrical parameters

1. **Question:** What is the minimum and maximum frequency of the PCA9515?

Answer: Since the PCA9515/16/18 are not clocked devices, the data sheet does not mention a clock frequency, but they can operate up to 400 kHz depending on system timing requirements. An F-max for a buffer is hard to deal with and it makes more sense to discuss the maximum toggle rate of the output. At the system level the principle issue is that the PCA9515/16/18 adds delays (170ns T_{phl} and 78ns T_{plh}). These delays aggravate the set-up and hold time requirements and also improve the edge rates that may or may not cancel out the delay. Thus, although all of the components in a system may be capable of operating at 400 kHz, the delay through the PCA9515/16/18 may cause set-up or hold time violations that limit system operation to something less than 400 kHz. The PCA9515/16/18 are capable of running well above 400 kHz (input/ output toggle frequency and they are tested at a 1 MHz frequency in final test) but a system with a PCA9515/16/18 in it may not be capable of 400 kHz. The devices cannot be used to connect a bus running at 100 kHz to a bus running at 400 kHz since all the buses always need to be operating at the same speed unless they are isolated from each other and a bus master is located on each of the segments. Minimum speed is the minimum speed specified by the I²C protocol, which is 0 kHz.

2. **Question:** We want to use the I²C at 400 kHz. However if we use the P82B96, the data transfers will go down to less than 100 kHz right? We want to know the way to increase noise immunity without using the P82B96.

Answer: We updated the P82B96 data sheet and moved the oscilloscope traces from Application Note AN460 into the data sheet. They show P82B96 waveforms when operating at 624 kHz. We also added the feature "capable of 400 kHz when system delay times permit" to the data sheet. The reason we do not generally say P82B96 is suitable for 400 kHz is that it introduces some propagation delays and a system designer must check those delays are acceptable in a total system. I have just checked our PCA9564 parallel to I²C bus interface and we can claim the P82B96 can be used in systems using the PCA9564 and the delays of the P82B96 will NOT limit the system speed. PCA9564 can be run at its maximum speed of about 360kHz as a master, or 400kHz as a slave.

If you can tell us ALL the ICs to be used in your system, and we can check their timings, then we can tell you whether 400 kHz, including using the P82B96, is possible. Usually the problem is that the timings for ALL ICs are not available. That is the only reason we can't publish P82B96 as ALWAYS suitable for 400 kHz. Some typical timings of various ICs are given in Appendix 7.

3. **Question:** Is the PCA9516 SMBus compatible?

Answer: The PCA9515/16/18 and P82B96 are SMBus compatible with regard to the normal I²C component of the SMBus standard.

4. **Question:** Why can't the P82B715 be used with SMBus devices? I tried it and it works.

Answer: The P82B715 was designed well before SMBus existed and it can't work with the SMBus 'weak' pull-down current (350 μ A). It senses bus sink current with a comparator across a 30 Ω sense resistor and the sense threshold level is almost the sink current level specified for the SMBus. Of course, it is virtually impossible to design an SMBus logic driver with only 350 μ A sink current and it would be surprising if they couldn't sink 3 mA. The SMBus also allows relatively large series protection resistors in each device output. We have to recommend designing to published specs. Many SMBus devices (and all Philips SMBus devices) are specified to sink standard I²C currents (3 mA) so of course, those may be used provided series protection resistors comply with I²C.

5. **Question:** Can the PCA9516 V_{OL} be improved too less than 0.5 V? The goal is to be around 0.3 V or less.

Answer: The PCA9515/16/18 are specifically designed to drive a low of 0.5 V and the difference between the family 0.3 V low and the special 0.5 V low on the PCA9515/16/18 is used to prevent lockup and yet accommodate

arbitration and clock stretching. The 0.5 V low should still allow plenty of noise margin to the $0.3 \times V_{DD}$ input threshold of I²C functions even with $V_{DD} = 3.3$ V ($0.3 \times 3.3 = 0.9$ V).

6. **Question:** I've been looking through the PCA9516 data sheet and have some questions/clarifications. To avoid a buffer output from driving its own inputs low, you set the input low threshold 70 mV (typical) below Vol. From the data sheet how do I determine the max Vol my external master needs to ensure a logic low is seen on the buffer input. I would assume min Vol - max Vilc. Am I overlooking something? We cannot daisy chain these devices correct?

Answer: Yes, it is correct that buffer devices cannot be daisy-chained. Special logic levels are used to avoid latching when there is a need to buffer/extend an I²C bus. The valid low level is specified as following: $(V_{ol} - \max V_{ilc}) = 0.47 - 0.07 = 0.4$ V.

7. **Question:** I saw a note about use of the PCA9515 in a generic multi-master I²C environment (2000 Jul 06) saying that: "In a generic I²C environment only one repeater can be used because the output of one repeater will not be recognized as a low for arbitration purpose by a second repeater." Please explain why the repeater cannot be used repeatedly in the one I²C BUS. It is appreciated if you could explain the technical background.

Answer: Two PCA9515 cannot be used in series because of the way these devices sense whether a low on the line is being driven externally (i.e., by a device on the bus connected to that segment) to the device or internal (i.e., by a device on a bus connected to the opposite side of the device) to the device. A low on one end of the bus will not be seen as a low at the other end of the bus if it must travel through more than one repeater. If the buffer design allowed a 'low' on one side to cause a "normal" low on its other side then, because the device is symmetrical, it will pass that low back to the first side and just latch itself low. To avoid this, we introduced another special logic 'low' level, created by the device. It looks like a 'low' for externally connected devices but it is really about 0.5 V. A level of 0.5 V is not recognized by the device itself as 'low' so it is not passed by the device. When a second device is connected in series its output 'low' of 0.5 V is not recognized at the input of the second device. The PCA9511 is being designed to be used in series. The P82B96 only uses special logic levels on ONE side. You can connect devices in series or parallel if you use the side with normal logic (Tx/Rx). Note that two PCA9515 can be used in parallel if you have the configuration described in Figure 31, because slaves do not need to see the acknowledge, data, or clock-stretching from other slaves.

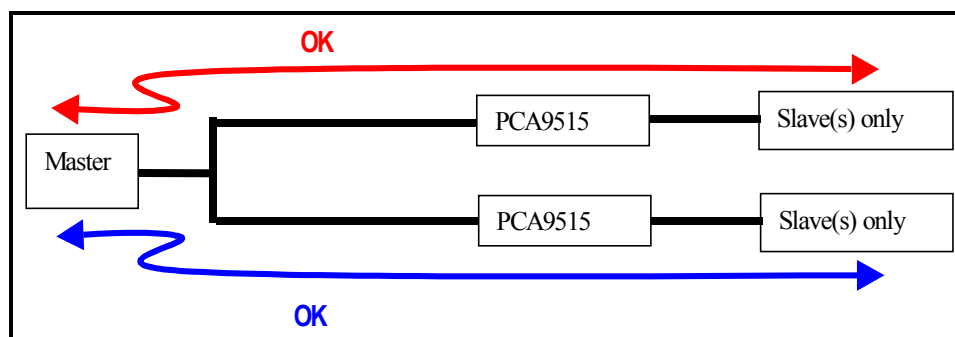


Figure 31. Using PCA9515 for parallel branches

8. **Question:** Is it possible to pull up each I²C segment to a different supply voltage (3.3 V or 5 V)?
Answer: Yes, the upstream and downstream segments can be pulled up to either 3.3 V or 5 V.
9. **Question:** I am powering the PCA9516 at 3.3 V, but the enable lines EN1 to EN4 would be at logic 5 V coming from another semiconductor device. The specification says that the PCA9516 inputs are 5 V tolerant. I just want to confirm this.
Answer: Yes, the enable and I²C inputs/outputs of the PCA9515/16/18 are 5 V tolerant even when powered at 3.3 V V_{CC} and allow a voltage up to 6.0 V.
10. **Question:** We are using the PCA9516 and are powering the device with 3.3 V. On the 1 Port, we have a 5 V I²C Bus. On the 0 Port, we have a 3.3 V I²C Bus. With a hard low (nearly ZERO Volts) on the SDA line of the 0 Port we get only 0.7 V to 0.8 V on the SDA line of the 1 Port. The spec says it should be 0.6 V maximum. We are using

2.2 k Ω pull-ups and have tried more than one device. While our system is working, we would like more noise margin. What is your recommendation?

Answer: The PCA9516 is tested for 6 mA at 0.6 V in production. The simulations indicate that the 0.6 V holds for currents well in excess of 10 mA. I do not know how you are getting the 0.7 V to 0.8 V that is being reported, because 2.2 k Ω at 5 V is only about 2.3 mA, and it should read < 0.6 V. Possible explanations may be an output series resistor of 50 Ω or more, ground offset of > 0.1 V, or a lot more current than expected. All three seem unlikely, however if the series resistor is present, which is a common technique, I would recommend removing it, or at least moving the pull-up so that the current does not have to flow through the series resistor. Ground offset or excessive currents would require solutions but I do not know what to suggest. Another possibility is that a rise time accelerator is on the 5 V line and is causing contention with the PCA9516.

11. **Question:** Can I safely interface a buffer with TTL level logic (not CMOS input level as required for true I²C)? My I²C master controller is inside a processor that is performing I²C protocol, but its input level is TTL. The big concern I have is the weak 0.8 V 'low' drive and the 0.6 V minimum 'low' input threshold on the P82B96. Is there any device that I can put in between the P82B96 and the processor input to take care of the level problem (i.e., a 0.8 V driven by P82B96 will become a 0.4 V low level for the processor)?

Answer: Yes, there is mismatch of logic level specifications at interfaces between the P82B96 and Micros with TTL levels when their I/O lines are used as I²C drive lines. Philips' Micros that feature an I²C interface provide I²C compatible CMOS levels for I²C bus pins so this mismatch situation will normally only occur when the Micros used have no hardware provision for I²C.

To avoid this logic level specification mismatch, and usually timing problems as well, we recommend that every effort is made to select a processor with a hardware I²C port whenever working in systems with buffer ICs. In practice, the theoretical mismatch with TTL is unlikely to cause any REAL problems and if we add components to cause a closer theoretical matching of the paper specifications this will actually degrade the REAL situation with regard to typical noise margins. A few precautions can minimize the chances of any problems:

For best margins in this situation, try to run the P82B96 Sx side at its minimum specified pull-up current (200 μ A). The P82B96 V_{OL} of 0.75 V at 25 °C or 0.85 V at -25 °C. TTL switching levels for various families of logic or Micros normally range between 1.1 V and 1.5 V so, in practice, the levels produced by P82B96 (with a light pull-up) offer reasonable typical noise margins, but no guarantees. TTL usually guarantees a low of 0.4 V when sinking 1 mA providing a guaranteed margin below the P82B96 input threshold of 0.65 V at 25 °C or 0.55 V at 75 °C. The actual Micro LOW may be specified as 0.4 V, and its real saturation at 200 μ A will probably be under 50 mV. However, ground currents flowing from other port pins of the micro will raise its 'internal' ground and therefore the actual voltage seen at the I²C buffer pin, so the real margin can be very small. The 'low' of 0.6 V driven by PCA9515/16/18 provides a guaranteed margin below the TTL requirement of 0.66 V (minimum for 3 V) but its input threshold (contention or repeated 'low' data) of < 0.4 V has a lower noise margin than P82B96 when driven by the TTL output low level.

The only buffer that does not use special logic thresholds is P82B715. Although it cannot be used above 100 kHz, it might offer the best compromise if processors with TTL levels must be used. It will not change the logic voltage levels by more than 100 mV. Although we usually say it cannot increase the total I²C capacitance beyond 400 pF, there is a special application that allows around 3000 pF, see Appendix 2.

12. **Question:** About that parameter "Low Level Output, Vol". In the PCA9515 device specification, May 13, 2002, page 6 the minimum "Vol" is stated as 0.47 V. In the Philips "I²C-Bus Specification, Version 2.1, Jan 2000", page 35, the "Vol" is given as 0.4 V maximum. The test conditions indicated are slightly different, but can you explain why the device specification is worst than the Bus Specification Standard?

Answer: The output of the PCA9515 is designed to drive a low of 0.52 V. It does not source any current so this low level can be easily over driven by an external driver, that is another driver that can drive the low down to 0.4 V or below without having to drive any current from the PCA9515. The PCA9515 drives 6 mA at less than 0.6 V that was the old high drive specification. The PCA9515 uses the difference between the 0.4 V level and the 0.52 V level to tell when the low is externally driven or when the PCA9515 is driving the low. This allows it to repeat clock stretching and arbitration signals without locking up. However, it does consume device of the noise margin. The 0.47 V minimum is proof that the input which is set at about 0.45 V will never see the low driven by the PCA9515 itself as if it was an externally driven low, which could cause lock up, is not possible from a DC test point of view. The PCA9515 is not compliant with the I²C -bus specification for Vol of 0.4 V @ 3 mA, it is compliant with the 6 mA condition of 0.6 V @ 6 mA that was recently dropped from the I²C -bus specification. The PCA9515 Vol is well below the Vil specification of the I²C -bus and is therefore interoperable, with some small loss of typical noise margin.

13. **Question:** We have trouble using the PCA9515 "I²C bus repeater" in our application. The acknowledge from the slave device (EEPROM in the Flat panel PHILIPS X150 with the DVI interface for plug and play) on the I²C bus 1 will not transmit to I²C bus 0. We try with our custom display and it works without any problems. We want to use the PCA9515 to isolate two I²C bus (we have two masters but these masters do not provide the I²C bus multi-master functionality) and to provide 3 V to 5 V translation levels. This I²C link is use to configure the video output for a graphic controller by reading the flat panel configuration in an EEPROM. (DVI specification: DDC compatible: plug and play)

Our design configuration is as following:

- The PCA9515 I²C bus 0 is connected to a graphic controller (Intel 810E master DDC I²C) on a motherboard (SDA0 and SCL0 have 2 k Ω pull-up to 3V3)
- The PCA9515 I²C bus 1 is connected to a serial EEPROM (PCF8582C-2 with 5V supply) (SDA1 and SCL1 lines have 2 k Ω pull-up to 5V) and two I/O (programmable device: high impedance during this test)
- Power supply 3V3 for the PCA9515

Answer: The waveforms with the Philips flat panel device (which shows a failure to propagate the acknowledge) clearly show that the low driven by the flat panel driver device is 1 V. The PCA9515 does not recognize a 1V level as a low, unless the pin was high first, since the pin was being driven low by the PCA9515 to ~ 0.5 V before the flat panel driver device held the pin low at 1 V. The PCA9515 interprets the 1 V as a slow rise rather than a low. The low level needs to be below 0.4 V for the PCA9515 to repeat it as is the case for the OK graph. The pull up resistor on the I²C bus where the Philips flat panel device is located is too low in value causing the low level to be 1 V. A sufficiently high value resistor that will result in a low level of ≤ 0.4 V should result in successful operation.

14. **Question:** Is there characterization data for the P82B96 device at 3.3V. We would like to know the V_{sx}, V_{sy} output low voltage (max) over temperature. I_{sx}, I_{sy} = 400 μ A. Can you extrapolate?

Answer: The P82B96 is specified over the full V_{CC} range 2 V to 15 V and its port sink characteristics hardly change over that whole voltage range. This is a bipolar device, not CMOS.

I will give you typical characteristics over voltage and temperature, but the data sheet contains the characteristics that we guarantee over the full range 2 V to 15 V. For very important business we can certainly consider setting tighter test limits and guaranteeing performance closer to the typical performance but please remember it is a very major QA exercise to change anything in a data sheet!

P82B96 characteristics at S_x/S_y:

The output low at S_x/S_y typical at 200 μ A is specified at 750 mV.

In a qualification report we see a measured spread 725 – 750 mV over the whole 2 – 15 V supply range.

Because there are no published data sheet limits, the tester limits set today are rather wide, at 650 - 950 mV

This voltage is set mostly by a diode-connected transistor so we quoted the nominal temperature coefficient at -2 mV/K. If that TC were added to the tester settings then the 'worst case' -40 °C limits would become 780 - 1080 mV.

The actual measured spread at -40 °C was 832 - 860 mV. Similarly for +125 °C the test limits are equivalent to 450 – 750 mV and measured spread was 540 - 569 mV. The actual TC calculated from device measurement at 200 μ A is nearer to -1.7mV/K.

To extrapolate to 400 μ A sink current instead of the published 200 μ A, just add 22 mV to all those figures.

To extrapolate other (lower range) currents apply the 60 mV/decade law for change of current in the 'diode' and assume the diode has a series resistor about 25 Ω . So the expected limits over -40 °C to +125 °C at 400 μ A will be closer to the range 560 – 880 mV. For other temperature ranges make corrections using -1.7 mV/K.

It is not worthwhile trying to make any corrections for 3.3 V because the change over 2 – 15 V is too small.

When the customer needs those limits they are probably interested in the input threshold at Sx.

There are limits published in the data sheet but for background here are the typical values.

Again this is diode-based and has typical spreads...

-40 °C: 757 mV - 787 mV

+25 °C: 635 mV - 648 mV

+125 °C: 425 mV - 466 mV

The TC calculated from the mean values at -40 °C/+125 °C is -2.03 mV/K

15. **Question:** Can you give more ESD rating information on the P82B715 and P82B96 as they are intended for applications including long cables and the potential for an operator picking up static electricity walking on the carpet and then plugging in a cable?

Answer: The devices had to pass the usual qualification testing before release. Human Body Model testing today uses both polarity pulses in 500 V steps to 3500 V, source impedance 100 pF/1500 ohms, on all I/O pins. Machine Model tests run 1000 V with virtually no series resistance. That is severe, but still cannot compare with a human charged to 20 kV or even more! Design to withstand extremes of ESD requires attention to system design. PCB layout is important, the discharge paths for the 'lightning-like' high pulse currents are not always easy to predict. But the design of these bipolar devices means it is RELATIVELY easy to apply effective protection using external components when conditions justify this. Results from the qualification report for P82B96 show the breakdown voltage of the Sx pins ranged 27 V – 37 V at 5 mA. That is described as an LVceo or Vceo (sust) test where breakdown is first forced using a higher voltage and the device 'sustains' that voltage with 5 mA flowing. For the Tx pins the 5 mA LVceo mean value was 46 V. This means that the I/Os can be protected against quite extreme conditions by applying series resistors and shunt Zeners that can have relatively high voltages. The figure shows an example of a module that will be hot-plugged to a bus. The protection components are shown in Figure 32.

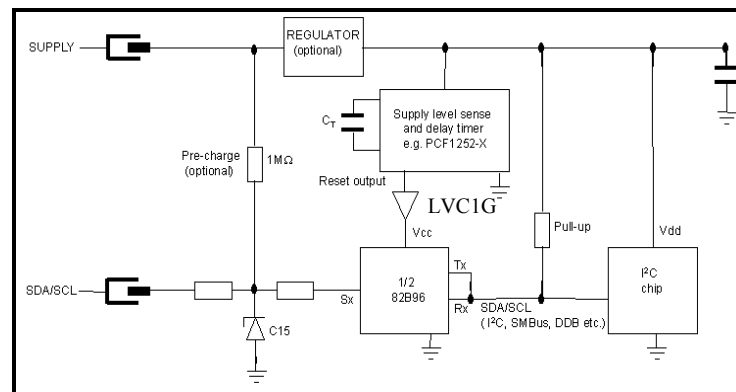


Figure 32. Hot Plug Example

In this example, a 'module' containing, for example, an I²C EEPROM device will be 'hot plugged' into equipment with power on and the I²C bus already active.

Hardware requirements are:

- Cause no disturbance to the main running I²C bus (no glitches to 'low')
- Ensure the SDA/SCL lines of the EEPROM follow the rise of the EEPROM supply rail with no 'glitches' to 'low' that could result from activity on the main equipment's I²C bus
- Provide effective ESD protection for the I²C devices in the module, for safer handling.

Meeting a) requires that the bus must not be pulled low by, e.g., diodes to the supply rail inside the EEPROM device during the time the module supply is being charged from zero to proper operating supply. (-- Or even just from charging the stray capacitance of the module bus lines from zero volts). This is achieved by delaying application of supply to P82B96 until after the EEPROM supply is valid. Without supply, the P82B96 will open circuit its I/O lines. At worst, the active I²C bus must charge the stray capacitance of the C15 Zener (about 30 pF) and P82B96 input (about 5 pF). When the supply/ground connections can be arranged to mate before the bus lines, 'precharging' the Zener and P82B96 input can prevent even small 'dips' on the main I²C bus, assuming it is high. Requirement 2) is also met because P82B96 releases Rx/Tx so the pull-ups on the EEPROM will cause the SDA/SCL lines to follow the rising supply. The 15 V tolerant I/Os of P82B96 allow ESD protection using high voltage Zeners having much lower capacitance than low voltage ones. (In any case 5 V devices cannot be protected by 5 V Zeners). Small series

resistors in the bus lines provide additional reduction of 'glitches' on the active bus. Because PCF1252 may 'enable' the bus connection during an I²C transmission, it is still necessary to use any software initialization routines as recommended by the EEPROM (etc.) device supplier. The protection concepts discussed above are also valid for 'floating cables' if you just accept that there is little chance to plug in high capacitance cables without causing glitches to zero. Of course PCB layout can play an important role in protecting any of the devices on it because the static discharge involves high currents flowing in paths that are hard to accurately predict. It is like controlling lightning! In general it should be possible to apply any of the customer's traditional solutions for protecting devices from large ESD spikes. Customers have fitted P82B715 as 'protection' devices for expensive Micros (they didn't need any buffer action). In more than 7 years no P82B715s have been returned to QA with static damage. P82B96 uses similar device processes.

16. **Question:** Could Philips provide a more detailed explanation of how the current based buffering approach used in the P82B715 preserves bi-directional operation (i.e., if current is flowing on either side (buffered or non-buffered) the P82B715 senses this and sinks current on both sides (pulls low)?
Answer: The P82B715 only senses and amplifies in ONE direction. That is, a current sunk at Sx is amplified 10 times. The signals from the buffered/amplified side (Lx) pass back to the normal I²C side via a passive 30 Ω resistor without any activity in the device. Our updated 2002 data sheet now includes a better explanation.
17. **Question:** If current is not flowing on either side (buffered or non-buffered) the P82B715 senses this and tri-states (hi-impedance) both sides of the bus?
Answer: It is best not to think about P82B715 in terms of the way logic devices operate, it is only a unidirectional current amplifier. So, no, it does not tri-state anything. The Sx side is always linked to the Lx side by a 30 Ω resistor. If a device at Sx pulls 'low' then the current in the 30 Ω resistor is sensed and an open-collector amplifier sinks extra current at Lx. If a Lx pin is pulled low there is no amplification but the normal I²C bus at Sx is passively pulled low via the 30 Ω resistor. When we named P82B715 we called it an 'extender', avoiding the use of the word 'buffer' because it does not really buffer. We called P82B96 a buffer because it more closely resembles a logic device and does provide true isolation/buffering of its I/Os.
18. **Question:** Does this same approach eliminate device latch-up?
Answer: None of the buffers/extendors that Philips currently offers have any known possibilities of latch up.
19. **Question:** Are there any issues with device latch-up that Philip's is aware of, and if so what does one need to do to avoid this?
Answer: No, there are no known issues of latch-up.
20. **Question:** We are using the PCA9516 in a 4:1 Multiplexer application. We are occasionally seeing a latch up condition on one of the slave ports and want to know of any errata's or other known issues with this device. I asked them to probe further to try and determine if the I²C latch up is being caused by the load on a slave port. Each port has a PIC micro and either an EEPROM or Philips ADC.
Answer: We are not aware of any latch-up, however if the Vil seen by the PCA9516 is greater than 0.4 V it will only be repeated (as a low) if the pin was high first. This affects clock stretching and acknowledge and most often results from the use of too low of pull-up resistors in combination with a device with a weak output drive.

I²C protocol

1. **Question:** We propose to use the PCA9516 to buffer and multiplex I²C signals to multiple sets of identically addressed devices and want to make a general broadcast in the I²C bus in order to send the same information to several of identically addressed devices at the same time. How will the acknowledge work? All the addressed devices will send an acknowledge information. Would it work?
Answer: Yes, this technique has been used for over 10 years, for example in diagnostic systems for telephone exchanges. In this type of application the first 'general broadcast' message will usually be more than just the address byte so it will cause all the addressed devices to do more than just acknowledge. For example they might be I/O devices and the same general data may be written to their I/Os. That I/O information will then generally be used for the next 'selective' messages on the bus. For example I/O port data might change the A0 and hence the effective bus address of another I²C device. Or all the devices might be Micros and the following data bytes can be 'decoded' by all Micros to effectively address only one Micro. If a device fails to receive (and/or acknowledge) the general

broadcast then this usually becomes evident in the NEXT communications and the system software needs to be designed to do the necessary checking and correction. This is simply saying that an I²C acknowledge is nice, but more secure communications should use techniques like writing data and immediately reading it back to be sure the transmission was correct. For 'general broadcast' it is not possible to check the 'general' transmitted data that way, so the checks need to be made on the actions that were expected to happen **AFTER** that first message.

2. **Multi-part Question:**

Question 2-1: In the theory of operation, you refer to what appears to be a bus conflict as clock stretching. What is it?

Answer 2-1: Yes: the event on the SCL line defined in the I2C specification as clock stretching looks like low contention, or since it is an open drain/open collector bus, it is also referred to as a "wired bus".

Question 2-2: What is the advantage?

Answer 2-2: The advantage is that multiple independent masters can share the bus with a simple arbitration scheme without the need for centralized control.

Question 2-3: The device can recognize if two inputs are being driven low. Does it do anything with this information?

Answer 2-3: When one is released and the other remains low it will hold the first input to the 0.5 V low, so that the master(s) will know that the line is still low.

Question 2-4: We have a large skew in t_{phl}-t_{plh}, my assumption is that this will vary with load and needs to be taken into account when determining the worst case clock pulse skew?

Answer 2-4: The t_{phl} is usually faster than the t_{plh} unless the RC is very small and the t_{phl} is determined by the slew rate control in the driver and not by the capacitance. The t_{plh} is a direct RC time constant so it is very sensitive to the load. The clock skew should only need to consider the difference in load between SDA and SCL. Any repeater will slow down the bus frequency by extending the pulse width low and pulse width high time requirement. The pulse width low is stretched by the repeater itself but for the PCA9515 and PCA9516 the master needs to allow a longer than minimum pulse width high

Question 2-5: Is the test load 7 pF, 50 pF or 57 pF?

Answer 2-5: The external load is 50 pF and the pin capacitance is about 7 pF so the reported numbers correspond to a total capacitance of 57 pF.

3. **Question:** For the PCA9516, I was wondering more specifically, if two or more enable pins see highs at the same time, what would happen in the arbitration logic? The logic that is driving the I²C hub may have two or more enable pins high for a small period of time (ns).

Answer: There is not any problem having more than one enable high at a time. Even all high at the same time is not a problem, as long as the enables change state after a stop and before a start, that is during a bus idle state. If you choose to change the enable pins during a transaction, there are a number of possible problems associated with the incomplete / interrupted transaction. When the enable pin is taken low, it turns off the outputs associated with that enable and the inputs coming from those pins. If the pins are high as in an idle state, everything remains high. However, if the pin was being driven low externally when disabled, the other pins will now rise because the logic sees a disabled pin the same as a high. If the pin being enabled is low (which it should not be), the internal logic will see the low as a falling edge and will propagate to all the other enabled pins. If another pin is disabled shortly after the low pin is enabled, there will be a momentary low going glitch on the disabled pin. If the enable is changed during a transaction, some I²C devices will see the incomplete transaction, some may be left with an incomplete transaction waiting for a completion and false start or stop conditions may be seen by some devices. Not all I²C devices tolerate these conditions well, thus the instruction is that the **enable pins should only be changed during an idle state**. The PCA9516 enable pins are direct acting and switch the associated port on or off immediately independent of transactions in progress. The internal logic drives all enabled pins (SDAn or SCLn) low when a falling pin is detected and holds them low until the last pin being driven low externally is released, provided that the externally driven lows are below 0.4 V.

Architecture / Applications

1. **Question:** Some engineers working on a rack mount server unit liked the idea of the PCA9516 and being able to isolate the bus but they need master/slave capability on both sides. The application is a server management board. It has I²C interface to up to twenty 1U servers with an embedded controller and the servers need to be able to master from any one of the twenty 1U servers. Can they use five 9516 hubs in parallel with the Main System I²C Bus Master so every subsystem can talk to every other subsystem like is shown in Figure 33?

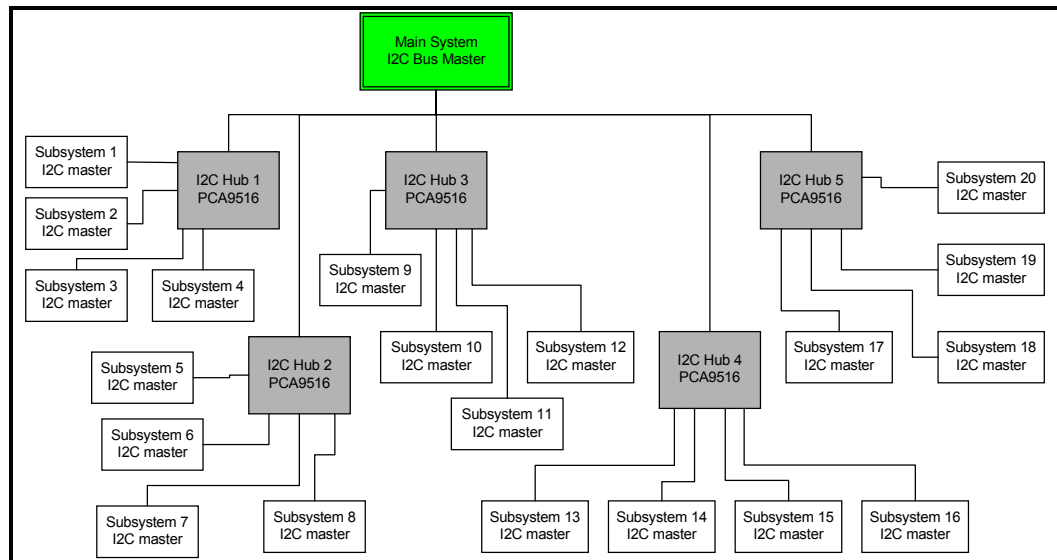


Figure 33. Twenty Subsystem Point to Point I²C System

Answer: The PCA9515 and PCA9516 (acting exactly as a 5-channel version of the 2-channel PCA9515) can handle multi-masters, that is any of the I²C lines can be tied to a master and both devices will handle bus contention and arbitration procedures common to a multi-master environment. In this mode however, **only one repeater/hub can be in the I²C circuit**. The PCA9516 low level is controlled to about 0.5 V which is not recognized as a low for contention resolution by the PCA9516 and the internal control logic assumes that if there is no contention the pin will go high when the PCA9516 stops driving it low. This guarantees latch-up free operation in a single hub system. However it also causes problems in a multiple hub system, because one hub will not repeat a low from another hub.

In the drawing, if all data movement is between the "Main System I²C Master" and a component on a hub branch it may be possible to make the system work. If the intent is for a slave on a branch of one hub to communicate only with the "Main System I²C Master" and if slaves on branches of the other four hubs will not be confused by hearing only the master part of the conversation, then this configuration will be acceptable. If the intent is for a master on a branch of one hub to communicate with a master on a branch of a different hub this is not possible because the second hub will not repeat the low from the first hub and this arrangement cannot be used.

If the loading on each 1U server is low enough to maintain the entire system at less than 400 pF you could consider using five PCA9545 switches to isolate the 1U server masters through I²C commands and still allow all 1U server masters to talk to each other. Note that the main master must be used to unisolate each 1U server master through I²C commands as the 1U server master has no control on the PCA9545 downstream channel selection when that channel is not selected. The 1U server master can isolate itself since the channel is not deselected until after the stop command is received. The best option is to use the PCA9518 that is designed specifically for this application. As many PCA9518 can be used in a system as required (five PCA9518s would be required in this system). The PCA9518 has four hardware enable pins per device that are used to enable and disable the channels but the bus must be idle/free prior to enabling or disabling the channels. The fifth channel of the PCA9518 cannot be isolated and can be used to connect to the system master or held high if not used.

2. **Question:** In our architecture the PCA9516 will be used to interface between the Master (SDA0 / SCL0) and slave devices on the downstream channels (SDAn / SCLn). The master has to control the correct enable pin to switch on and off the corresponding I²C segment. Is it possible to use the Hub in a master/master mode? How do you ensure the enable pins activation between the two masters?

Answer: Yes, it is possible to use the Hub in a master/master configuration with channel arbitration handled by a single master or a logical combination of the two masters. The only requirement is that the master enables or disables the segment during an idle period. Enabling and disabling of the segment is a problem that is isolated from the I²C bus, so it must be handled by the master or masters independent of where they are located for the multiple master case. One master could have sole control of the hub or a cooperative control scheme could be used. The enable inputs are active high with internal pull up resistors so an open drain / wired OR design could be used to control the enable signal. However some sort of “dirty flag” would be needed to communicate that a transaction is in progress and that the state of the enables should not be changed until the bus is idle.

3. **Multiple Part Question:** Please refer to the system diagram of the mass storage in Figure 34. We are planning to interconnect the boards by the using the I²C bus.

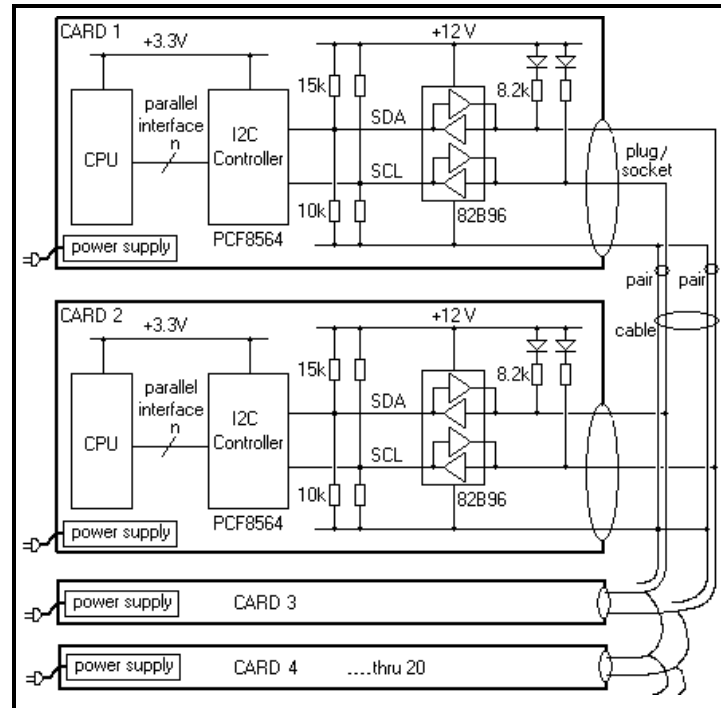


Figure 34. Mass Storage system, suggested configuration.

Each of the boards is connected by cable using the I²C buffer P82B96. We would like to know if the cable is allowed to be plugged in and pulled out while the system is working, i.e. we want to have hot-swapping capability.

Question 3-1: Hardware protection - We would like to know if the P82B96 has any protection circuit inside to allow hot swapping of the cable or do we need to design an additional protection circuit?

Answer 3-1: The P82B96 has very high breakdown voltages, the Tx output is typically 45 V. The bipolar transistors are also large because Tx has a typical sink capability of 100 mA. This makes the P82B96 very rugged and easy to protect against ESD. Without any external protection it passed 3.5 kV ESD testing to JESD22-A114 (human body model) and 1 kV to JESD22-C101 (machine model). Your System circuit diagram shows 12 V protection Zener diodes and they will give extra protection if required.

There is no hardware circuit inside P82B96 to 'delay' any connections to the long cable bus. The connection timings will be the same as the mechanical connection of the pins in the connector. When a new unit is mechanically connected to the 12 V Buffered bus then the capacitance on its (buffered) bus lines must get charged to the voltage level on the bus at the time of connection. During that charging time the bus logic level can be wrong for a short time. If the time is designed to be less than 50 ns then the I²C controller (PCA9564) will ignore that transient and there will be NO interference with communications in the system.

The system can be designed in different ways, but here is one suggestion. If it is possible to have the 12 V power supply active on the board before the cable is plugged to the board then the Tx/Rx side of P82B96 will be pulled high by the bus pull-up resistors. When that board is connected to the cable bus then there are two possibilities:

- a) The bus is logic high (12 V) at that time. In this case there is no transient. No problem created.
- b) The bus is logic low (0 V) at that time. In this case the capacitance of the bus lines must be discharged from 12 V to 0 V by the P82B96 that is driving the bus low. There will be some positive spike on the bus but it is allowed to exceed 6 V (the logic threshold of P82B96) for 50 ns before the PCA9564 will recognize it because it has an input spike filter that rejects transients less than 50 ns.

Because P82B96 has about 100 mA sink capability it can discharge more than 800 pF to below 6 V in less than 50 ns. The capacitance of one board, including Zeners and connectors, will be much less than 800 pF unless there are cables connected. So again there will be no problem. Perhaps we should claim "the combination of P82B96 + PCA9564 provides hardware protection for hot swapping when the board 12 V supply is active".

If the board power supply is not active during hot-swap then its bus can be at 0 V. When joined to the cable bus at its high state there will be a transient low signal with a time determined by the board capacitance and the pull-up present on the system. If the pull-up was, say, 4 k Ω (there must be at least two other boards or there is no bus communication to interrupt!) then it takes less than 20 pF stray capacitance to cause a false 'low' for >50 ns on the bus. If that causes problems in the PCA9564 then software will be needed to overcome the problem. This illustrates why it is better to have the 12 V present during hot-swap.

Question 3-2: Software protection - We understand that the I²C communication problems caused by hot swapping can be prevented by the software design. Could you advise or provide know-how on the I²C software programming?

Answer 3-2: We will try to get help for any particular problems that are found. The problems created by hot-swap mostly relate to the interruption of a message when the plug is pulled during a message. The 'master' will get no valid response from the 'slave' but the master will not ALWAYS know that invalid data has been provided. As one example, consider a master receiver is reading a data byte from a slave transmitter. If the plug is pulled in the middle of the data byte then the master thinks all following data bits are bus high level. The master transmitter provides the ACK and the STOP so it does not know the slave was disconnected. It thinks the transmission was OK. Only software testing of the links can overcome that problem. As an example, if important data has been read then the master can simply address the slave at the end of the data transfer and check for the acknowledge that means it is still connected. For even more security the master can request a data byte from the slave (microcontroller) that contains a calculated 'checksum' for the data it just supplied.

Question 3-3: Power shut down of a board – What about the case that one of the boards of the system shuts down its power for maintenance? We are concerned about power running into the board and would like to confirm that P82B96 becomes high impedance when its power is down and does not inhibit the I²C bus.

Answer 3-3: Yes, P82B96 releases all its I/Os to high impedance when V_{CC} falls below typical 1.5 V. There are no paths from any I/Os to the supply pin. Diodes in series with the pull-ups are needed to block this external path. Pull-ups on the Sx side to 5 V are OK but the supply to PCA9564 must be 3.3 V.

If the 5 V supply fails on one card, but the 12 V supply remains active, then the 12 V bus lines will be pulled low and the main 12 V bus cannot operate. If the 5 V supply is generated from the 12 V supply this should not be any problem. Another solution can be to use the voltage dividers shown, from the 12 V supply, instead of simple pull-ups on the 5 V interface. If the 3.3 V power supply (alone) fails then PCA9564 releases its SDA/SCL and will cause no system problem.

Question 3-4: Please confirm the driving capability of P82B96 allows these values:

Number of the board in the system:	20
Maximum length of the cable:	10 m
Driving current of P82B96:	30 mA
Current sink required for each board:	1.5 mA
I ² C Pull up:	12 V, with a resistor of 8.2 k Ω

Answer 3-4: Yes, 30 mA is the static drive current for P82B96. The corresponding guaranteed low is 0.4 V. The typical drive capability is over 100 mA at 0.4 V. Having distributed pull-up resistors is a good idea because it makes all boards identical and adjusts the pull-up to the number of boards connected. Sinking the pull-up current of 20 boards at 1.5 mA each is guaranteed by the data sheet, 30 mA static sink specification. There could be (speed) problems if 10 m were allowed for each board and the total cable was then 200 m. For 10 m, the longest cable propagation delay is about 100 ns. That delay won't cause any speed restriction in a 100 kHz system with only PCA9564 devices connected.

Question 3-5: We would like to confirm that under the condition above, the t_r (1000 ns) and the t_f (300 ns) at the data rate of 100 kHz is guaranteed. Or do we need to compromise to reduce the transfer rate too less than 100 kHz?

Answer 3-5: There will be no problems running the system at 100 kHz with PCA9564. The typical fall-time when P82B96 is driving low is around 100 ns. As a guide, another customer wants to use P82B96 and opto-couplers in a 400 kHz system. I think even that might be possible! The clock options for PCA9564 seem to be 90 kHz then 150 kHz. You could even use 150 kHz provided the bus rise-times allow it.

The most significant timing limitation will probably be the buffered bus rise-time. With 8.2 k Ω resistors, the normal maximum bus capacitance for 100 kHz would be about 120 – 150 pF. If 20 cables make up the 10 m total length then the average is 0.5 m cable. The cable capacitance will be only about 40 pF. The connectors + board traces + Zener + P82B96 might be about 50 - 60 pF total, so it looks OK. I would suggest using 15 V or 16 V Zeners (e.g. BZX284-C15) because they have lower capacitance than 12 V ones. If the ESD specifications of P82B96 are sufficient then Zeners are not necessary. I have found that the fast fall-times of P82B96 (less than 50 ns) can cause ringing on the 12 V bus because it is not terminated in its characteristic impedance. A good way to prevent bus undershoot is to fit small Schottky diodes like BAT54A to prevent negative swings and dissipate the energy (see Appendix 5 for RC alternatives)

Question 3-6: Why do you introduce discussion of propagation delays in buffers when they are not found in the I²C specifications?

Answer 3-6: For several customers wanting to expand their I²C systems, we need to introduce bus propagation delays into the SCL and SDA signals but the existing I²C specifications do not expect bus delays, they were written about small systems inside one enclosure. So I²C devices that "conform" to 100 kHz or 400 kHz bus specifications are allowed to use up ALL of the available timing windows allowed in the I²C specifications and that means it is not possible to introduce any other propagation delays. But if we know the timings ACTUALLY needed for a certain group of I²C devices in a system, then we can calculate whether there are really margins available. For example, if we use faster 400 kHz devices in a 100 kHz system we should have margins available that allow us to introduce other delays.

In a system we want to suggest for our customers, the only I²C devices in the whole system are the PCA9564. There will be 20 of them used in a multi-master system as shown in Figure 34.

This system will introduce two bus propagation delays:

1. Delays caused by buffering with P82B96
2. Cable propagation delays up to about 100 ns. (That is what I call 'round trip' delays, meaning the sum of SCL delays from master to slave then the SDA delays back to the master. The total cable can be 10 meters long). The P82B96 delays include internal logic propagation delays (400 ns from master to slave and 400 ns slave back to master) and effective propagation delays caused by bus rise and fall times on the long cables. The bus signals pass through two P82B96 devices that cause 300 + 100 ns for each direction. Because the delays will be identical on the SCL and SDA lines, when the master supports 'clock stretching' then effectively about half the total introduced propagation delay does not need to be considered because the rise of the SCL line is delayed by P82B96. The SCL line at any one PCA9564 cannot rise until after ALL other sections of the bus system transporting the SCL signal have risen. Effectively, PCA9564 attempts to release SCL but it will not rise to a logic 'high' level until that 'attempt' to rise has propagated to the ends of the longest cable and back again. That total delay time will be 400 ns for the P82B96 device delays, 100 ns for cable delays, and some allowance for bus rise times. Total stretch of SCL 'low' will be something over 500 ns. To design a system with introduced signal delays, an important hardware timing is the time for a slave to output its valid data or an Acknowledge on its SDA line, following the falling edge of its SCL line that effectively 'requests' this response. The data must be available to the Master before/at the subsequent SCL rising edge. The timings simulated for PCA9564 are given in Appendix 7 and based on those timings, we can say that P82B96 and cable lengths of 10 meters will not cause any reduction in the system speed limit (about 330 kHz for a Master)

4. **Multi-part Question**

Question 4-1: In our server application, we are trying to daisy chain the I²C bus between servers and need to get 256 nodes, or more, at 100 kHz. We are currently using the P82B715 (two per server) in a series arrangement as shown.

Answer 4-1: Using two 82B715's does not help. They do not 'buffer'. Multi-drop is the better option.

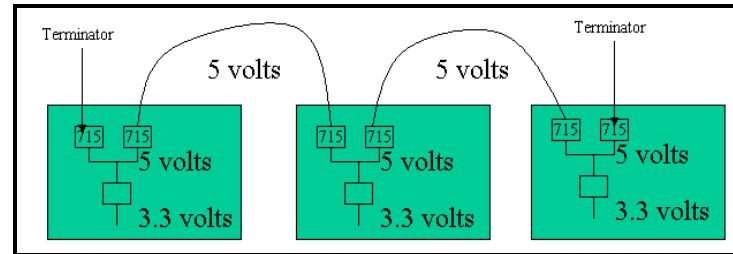


Figure 35. Multi-drop application using P82B715

Question 4-2: We cannot get more than 16 servers to talk before the ground offset combination no longer has a valid low.

Answer 4-2: I am surprised you can even get 16! The bus capacitance limit will be exceeded.

Question 4-3: We are using the LTC1694 to terminate either end to allow for better waveforms.

Answer 4-3: The LTC1694 will only make a marginal difference. They are designed for 3 mA I²C. Once you buffer to 30 mA their contribution becomes small, presuming you already have a low resistance pull-up. P82B715 has NO GAIN for rising bus signals. It just looks like a 30 Ω resistor joining its I/Os. See P82B715 Block diagram in Figure 8. Maybe distributing them around can help but it is not worth following this line.

Question 4-4: We also need to change the 5 V to 3.3 V on the card for the card components.

Answer 4-4: The P82B96 translates voltages. Alternatives can create problems.

Question 4-5: We may be able to solve our problems if we can get the device schematic.

Answer 4-5: We give the 'equivalent circuit' for P82B715 in Figure 8, and in the data sheet. It is an analog amplifier. The best equivalent circuit for understanding P82B96 is slide 9 of our slide set on our website:

<http://www.philipslogic.com/support/appnotes/anp82b96.pdf>

Question 4-6: We did not want to hook it up as in the proposed solution since all the capacitance of the wires, connectors and devices would be combined and the P82B96 would have to fight against this very large combined capacitance.

Answer 4-6: Agreed! To provide a solution we just need to know how large that capacitance really is.

Question 4-7: The idea behind the use of the P82B715 was to isolate the capacitance to each wire segment (which I do not think it does).

Answer 4-7: Correct, it does not isolate the capacitances. It won't work for 256 nodes. The P82B715 is not suitable for your application. It really does not isolate any section from another, and the total capacitance referenced to one I²C node must not exceed 400 pF. With 256 nodes, that is impossible. There is also a problem with level shifting and delays through each P82B715. The original drawing using the P82B715 shows a 'node' as a 'module' - maybe a card, or a complete server etc, that probably has multiple devices on it. To provide you a solution, the number of devices in each module isn't critical, as long as the total bus loading within each module is less than 400pF.

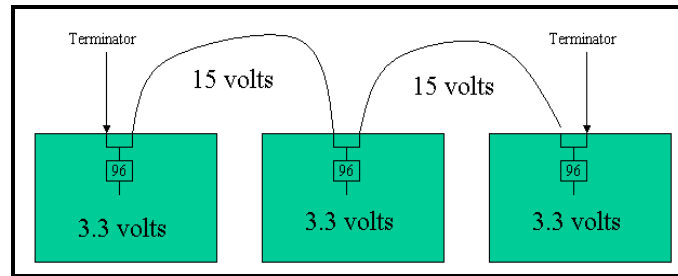


Figure 36. Multi-drop application using P82B96

Do you really have to interconnect up to 256 such modules in one multi-master, multi-drop system?

Being honest, I do not know of ANY existing buses that can do that, so if you know some, please tell me so I can see if we can adapt to them so they carry I²C signals. I think the limit for CAN bus is about 110 nodes, and RS485 only allows 32 transceivers. It is true that P82B96 makes it theoretically possible to build such a system – but you might become the pioneer!

The proposed solution using the P82B96 looks generally OK. Only 3 nodes are drawn and it gets complex if there will be 256 in total on the 'multi-drop' bus and running high logic voltages. The P82B96 alone can only directly drive 2 nF of total cable wiring when the logic voltage is 15 V (30 mA sink so 500 Ω pull-up means, conservatively, 2 nF). For 256 nodes that allows only 8 pF/node and I'm guessing you will need to allocate some tens or even hundreds of pF of wiring/connector capacitance for each node. If I start with, say, 250 pF/node (so maybe 10 feet of cable per node) then the total system capacitance is around 64 nF and we are also allowing for a very wide-spread system - 2560 feet of cable or 1/2 mile in total! I begin to doubt that is really what you have in mind. Such a system cannot ever run at 100 kHz anyway, cable propagation delays would limit it. When the total system capacitance gets that big, it is better to try to run the cables terminated in their characteristic impedance so we do not think about 64 nF, we think about 50 to 150 Ω depending on the wiring. If you have ground potential differences, you have to use either a high logic voltage or go to a differential bus. If your systems have already used some multi-drop differential bus, then probably the P82B96 can interface to its transceivers.

Maybe you have 256 nodes, but maybe only inches between each module in a rack. So maybe only 30 pF of wiring capacitance per module? The P82B96 is typically 5 pF. That is a workable total, less than 10 nF. You are allowed to boost the drive capability of the P82B96. Staying with this 'brute force' version of I²C, you add a PNP transistor to the Tx output stage of the P82B96 and then you can easily drive 1/2 A into the cables. Even 15 V logic drive into 50 Ω needs only 300 mA. So that is ONE possibility, but it uses quite a lot of power and you can imagine the possibility for EMI issues. Is energy efficiency critical? Can the wiring be shielded? This is a complex system, it needs careful design.

5. Multi-part Question:

Question 5-1: We want to use the I²C Bus under a multi-master and multi-point environment. Can we do that?

Answer 5-1: Yes, all solutions Philips offer will be multi-master and the bus will have multi-point and hot-swap capability.

Question 5-2: We want to have the hot swap capability on the board so that the cable can be connected and unplugged without having to power down of the board.

Answer 5-2: Yes, Philips can do whatever a customer requires, but there are different levels of performance and cost to think about before choosing a solution.

1. No ESD damage to the board components or the other boards on the bus.

What test will be specified, what model e.g. Human Body Model, and how many kV?

2. No disturbance to any messages on the running cable bus.

This means that when a new board is added, during a bus communication, there are not allowed to be any spikes or glitches on the cable bus that exceed 50 ns (for 400kHz bus).

Alternatively, adding a board while other boards are communicating may be allowed to cause failure of the communication of the message sent during adding the board. Then the system software will need to include checks that will enable a message to be sent again when there are problems. This is lower cost because it uses less hardware.

3. The board that is simply connected during a bus communication will receive only part of a message. It is possible that a device receiving a part message can "lock up" and will require a special software message on the bus to re-set it.

There are different solutions:

- a) Hardware to prevent any 'ends' of messages being received
- b) Software solutions that will generate re-send messages when a board does not respond.

Question 5-3: We have a single-master, multi-drop distribution bus with P82B96 and we want to keep the main bus signals safe on our board even if an extension bus cable to any slave boards is pulled off accidentally.

Answer 5-3: Which signals must be kept safe?

- a) The signals on the cable bus linking all the other boards that remain connected to the cable?

That is no problem. Those signals are safe.

- b) The signals on the board that is disconnected?

There can be a problem if the disconnection happens during a message to that board because the devices will keep waiting for the message to end, and no end will come. This can only be handled by special software. The devices on the disconnected board (that remains powered but loses bus signals) might require a reset message after the bus is re-connected. When the slave board message is interrupted the master might not receive any 'acknowledge'. In that case the master knows there was a problem. If the master was reading the last byte of data when the break happens then it cannot know about the break (because the master provides the acknowledge). So after this transfer the master must send a new message and test for acknowledge. Then it knows if the slave connection was broken. It must send recovery messages and poll the slave address until communication is restored.

Question 5-4: We want to design our I²C bus to be trouble free even if one of the power supplies of the board is shutdown, so it will not make any trouble to the bus and to the other boards.

Answer 5-4: Yes, it can be safe. We just arrange the power supply circuit of P82B96 so it causes bus disconnection when its supply (or some other supply) is removed. No problems when P82B96 is used as the bus interface. If a differential bus is used we need to select the correct transceiver as in Question 5-5 below.

Question 5-5: We would like to use a differential bus drive to protect the bus from noise and EMS.

Answer 5-5: Do you have special reasons to want a differential bus, or just good noise immunity is needed?

When a differential bus is used it raises several difficulties:

- a) Twice as many bus wires needed in the cable and twice as many connection pins on the plugs and sockets.
- b) Every bus wire needs static damage protection components. So this doubles the number of protection components.
- c) The differential bus driver/receiver devices will need to have a 'bus release' feature that guarantees no disturbance to the differential bus if their power supply fails. I know there are several ICs like this, but you need to confirm this feature for the actual ICs that will be used. Philips' CAN bus drivers have this feature. You will need two drivers for every board (in addition to one P82B96 on each board). Part numbers are e.g. PCA82C251 or TJA1040. Those CAN bus ICs, and most differential transceivers, require a 5V power supply. Is a 5V supply available?

Do you have another differential bus transceiver device you would like to use and can you send a data sheet?

Alternative proposal:

Use the P82B96 to generate high voltage logic on the cables. If there is a 12 V supply then drive the cables with 12 V logic signals. Is there any 12 V power supply available? The 12 V is only required at one place in the system, it is convenient to send it to other places by using pairs of wires SCL + 12 V, SDA + ground. The bus 'noise margin' is then around 6V because the logic threshold used by P82B96 is exactly half of V_{CC}. That means it needs interfering signals bigger than 6V coupled into the cables before there can be any wrong data. That noise performance is approximately the same as CAN transceivers like PCA82C251, and still about half the guaranteed immunity of our new devices like TJA1040. (I take the noise immunity limit as its maximum allowed common-mode voltage that is ± 12 V for TJA1040)

6. **Multi-part question:** We are developing a mass storage system and there are 20 boards in the system. We want to control these 20 boards using the I²C bus. Each board has its own Microcontroller. The architecture of each board is like this:

Microcontroller → PCF8584 or PCA9564 → P82B715 or P82B96 → BUS, and all 20 boards have the same architecture.

Question 6-1: We want to use this I²C bus in a multi master environment (20 masters). Can the I²C Multiplexer be used as a way to eliminate addressing conflict?

Answer 6-1: The I²C bus controller PCF8584 or PCA9564 has a 7-bit register where the designer can set the address of the device. So there can be up to 127 individual addresses and you could connect up to 127 boards in one system. 20 board application is not a problem and no multiplexer is needed. They can all connect to one bus. As you will see below, all the cards are allowed to connect to one "distribution" bus that has EXACTLY the same multi-master operation as any standard I²C bus.

Question 6-2: We want to connect each board by a 5 meter maximum length cable and want to reduce noise. We are thinking of using the P82B715 or P82B96. Can the P82B715 or P82B96 work in a multi master environment?

Answer 6-2: I think there is only one choice, the P82B96, because I think the total system capacitance will be too big for P82B715. With P82B96 there is NO limit on the total system capacitance. There could even be 127 boards in one system.

Yes, both P82B96 and P82B715 work in a multi-master environment. They do not change any of the usual operation of the I²C Bus, they just extend its application to very big systems with large capacitance. For this application the cables between the boards can also use high voltage logic to have best noise immunity if there is any high voltage supply available. Perhaps +12 V? It is best to use at least 5 V, but the P82B96 can use any voltage in the range 2 V to 15 V. In a server application for another customer we just used 3.3 V on some 2 foot long cables and there were no problems. I still recommend using 5 V or higher if there is a power supply available.

Question 6-3: When one of the boards in the system has trouble (e.g., power down or stops working), does this affect the operation of the other boards with a multi-drop bus architecture like this?

Answer 6-3: When using the P82B96 to drive the cables between boards it is very simple to add some simple logic gates to test the power supplies and to isolate any card that has a power failure. It is not as easy to build a system that can detect a failure of one of the I²C devices anywhere in the system, but it is certainly possible if the system designer adds some low cost logic and writes some dedicated software routines for each microcontroller. The microcontroller must be able to detect a general failure, for example by attempting to send a message. If a bus failure is detected then the microcontroller must disconnect its card (power down its P82B96) and test again. If the fault is still present then the fault is on its own card and it can keep its P82B96 powered down so this faulty card is isolated and the rest of the system can run. Other cards must then recognize that one card is not responding and raise an alarm to have that board repaired. That is only a simple example to explain one system protection. The total system reliability depends on the software and there are many more techniques to improve reliability.

Question 6-4: We want to have hot-swapping capability for the boards. How do we add hot-swap to this bus architecture?

Answer 6-4: If I understand correctly, the cards do not simply plug into a rack, there are also long cables and connectors in the system. The P82B96 offers a big system advantage when trying to hot-plug the cables/boards. See Question 15 on Page 34 for an example of some system design techniques. Today there are newer reset ICs than PCF1252 in our range, but their function is the same.

Question 6-5: For differential bus driving with the P82B96, is there some additional device required other than the P82B96?

Answer 6-5: If you want to use differential signaling on the cables then we can provide solutions for this. But first we should explain that just using a higher logic voltage on the cables makes a simpler solution for working in a noisy environment. If a differential bus is required then we need to know your preferred bus driver for the system.

Question 6-6: Don't the connection made by the P82B715 or P82B96 break the rules of I²C? The addressing and multi-master modes of the I²C bus do not work in the cable connected to P82B96?

Answer 6-6: Our buffers/extenders do not change any of the normal operating modes of I²C. Addressing and multi-master operation are exactly the same. When you make a low impedance or higher logic voltage signaling system for driving the cables then that bus has all the same multi-master features as any normal I²C bus. So you make one special bus for driving signals on all the cables and it has exactly the same characteristics as I²C. You can connect as many different cables/boards as you wish. Addressing is exactly the same. When using PCF8584 or PCA9564 you can have up to 127 cards in one system, all connected to one bus. The only system restriction when using long cables is that the operating frequency will be limited. The limit for PCF8584 is 90 kHz and the system speed depends on the slowest I²C device.

7. **Question:** I want to use P82B96 for opto-isolation in a Battery Management Project. There is one microcontroller master and up to 20 microcontroller slaves as shown in Figure 37. What is the correct configuration for the opto isolation of each slave? Speed is not important and I plan to use 6N138 opto devices. The opto devices take about 1.5 mA each. Do I need to do anything with the clock return line (pin 6 Ry of P82B96) in this single master application? Tie it to V_{CC} or GND?

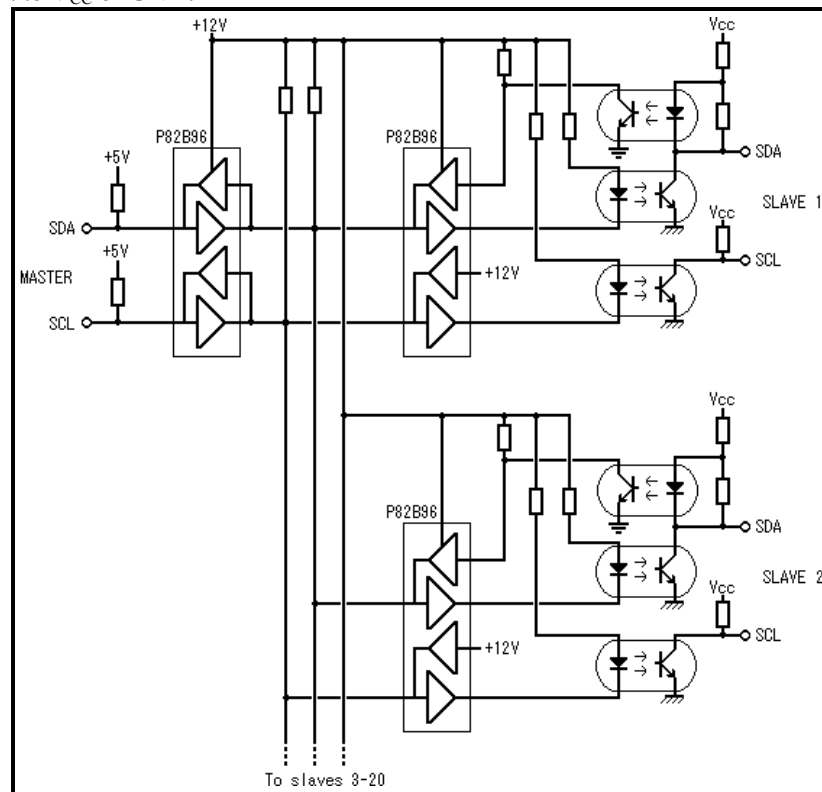


Figure 37. Opto-isolation application

Answer: Interesting Opto application. You have recognized that if the system is single master and the clock speed and software make sure there is no chance of clock stretching then you can save on opto-couplers in the SCL signal line. It can be uni-directional. So yes, you could tie the unused Rx inputs high with a resistor, but since you are never going to use them, just tied directly to the device supply is OK also. Don't tie it low! Tying low pulls the bus low.

The arrangement is slightly unusual, but certainly allowed. The Master drives the 'distribution bus' with its 30 mA static capability but any slave has to drive this bus with its nominal I²C interface side (Sx). Fortunately on a 12 V distribution bus the bus 'low' level hardly matters. Low is anything less than 6 V. The Sx side of P82B96 is guaranteed to sink 7 mA at a low of 2 V and in practice it will do it with a normal <1 V low. So I would recommend that the distribution bus pull-ups be calculated for, say, 6 mA sink current. That is 2 k Ω pull-ups to 12 V. That should be OK for about 1 nF (say < 10 meters) of wiring capacitance. If the bus and capacitance is small then you do not need to waste energy and you can use larger resistance pull-ups. The cheapest, simplest interface in AN460 really is very slow with 4N36 and the warning about 5 kHz should be taken seriously - for the values given and a 5 V supply. 6N138 looks very up-market in its published specs, but I never tried a Darlington opto for this job. Should work well. Just monitor the waveforms around the opto devices and check they are doing what the data implies.

8. **Question:** How do you include Opto-isolators into the bus wiring?

Answer: AN460 shows both low speed and 100 kHz opto-coupling circuit values using the P82B96. See also the "I²C Opto-Electrical Isolation" section of this application note.

9. **Question:** Could you please comment on 400 kHz Opto isolation applications?

Answer: It is necessary to know the timing requirements of the devices used in the system. In particular the response time delays to data valid or acknowledges from slave devices. 400 kHz can be achieved, even with P82B96 propagation delays, by using fast Opto-couplers as shown in Figure 25.

ADDITIONAL INFORMATION

The latest data sheets and IBIS models for the I²C and SMBus Repeater/Hub/Expander family of products and other I²C and SMBus products can be found at the Philips Semiconductors website <http://www.semiconductors.philips.com/i2c> or the Philips Semiconductors Logic Products Group website at <http://www.philipslogic.com/i2c>.

Only the Logic Product Groups website will include data sheets and IBIS models for new unreleased sampling I²C devices.

Software tools for most of Philips' I²C products can be found at: <http://semiconductors.philips.com/i2c/support>

Additional technical support for the I²C and SMBus Repeater/Hub/Expander family of products and other I²C and SMBus products can be requested by e-mailing the question to: pc.mb.svl@philips.com

See Philips Application Note AN460 for further application information about P82B96

See Philips Application Note AN10160-01 for further application information about PCA9511/12/13/14

See Philips Application Note AN462 for further application information about PCA954x Multiplexers/Switches

See Philips Application Note AN10145-01 for further application information about GTL2000/02/10 Bi-Directional Low Voltage Translators

APPENDIX 1 – Long Bus Calculations

Calculation of maximum clock frequency for long buses

The length of cable that may be connected between two devices in an I²C system is limited by the total of all the signal delays in the system and the required bus clock frequency. The requirement closely parallels the timing requirements for a processor reading a memory device. There is a maximum allowable delay between the timing signal that requests data and the time that valid data must be available. In an I²C system the falling edge of the SCL signal is the ‘request for data’ and the corresponding valid data must be available on the SDA line before the following rising clock edge that will effectively strobe in the data.

There is a specified data set-up time, 250 ns minimum for Standard mode and 100 ns minimum for Fast mode. In a true I²C system that allows for ‘clock stretching’, the master device will strobe the SDA signal using the real SCL clock edges that appear on the SCL wire as the reference timing. However, in many I²C systems the master can be a microcontroller with the bus signals generated on port lines using software and the strobe timing of the actual data read time is more often not accurately known. It is unlikely to be referenced to the real SCL bus clock edges, unless specifically designed for ‘clock stretching’. The system delays must meet the specified timings for two different cases:

1. When working with the Acknowledge from a slave (SDA has to be pulled to 0 V)
2. When reading data from a slave that involves the SDA line changing from 0 V to V_{CC}.

The important timings are different because the bus falling edges are determined by different factors from those affecting the rising edges. In general the falling edge delays will be shorter.

An example of the important timings for the P82B96 working with Acknowledge is shown in Figure 1.

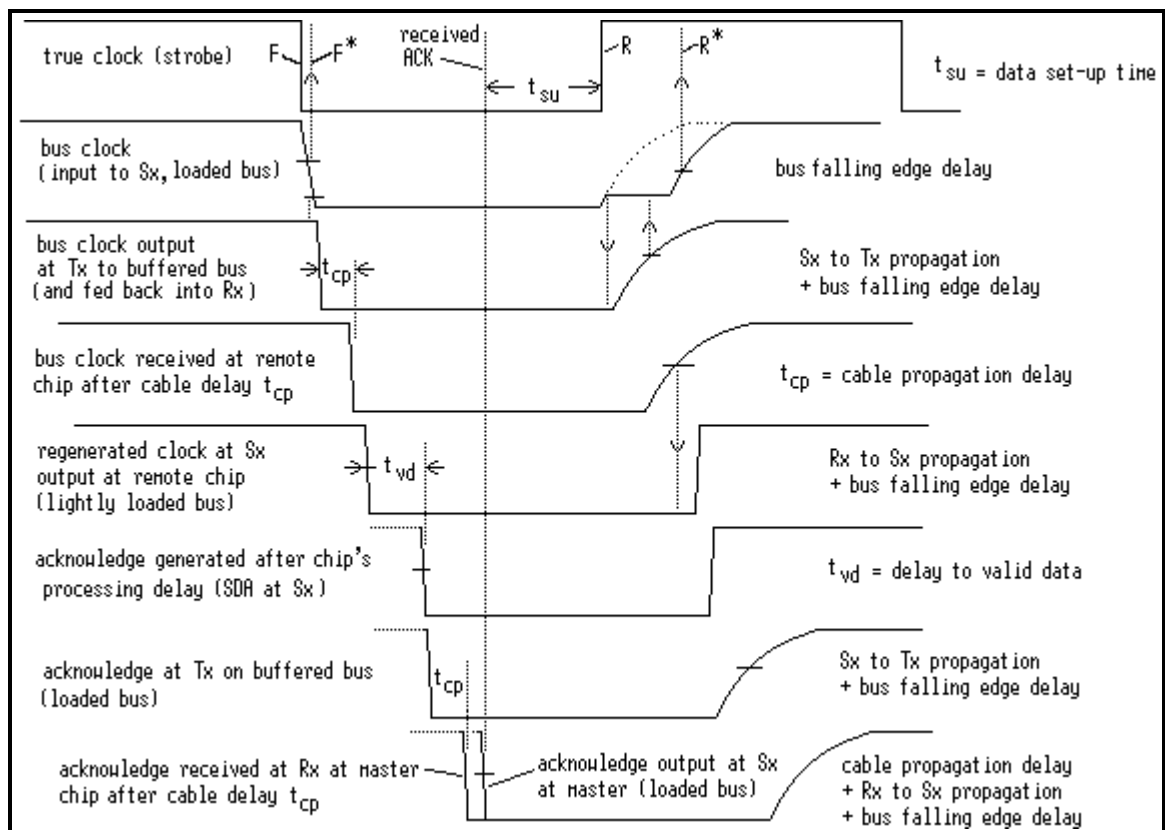


Figure 1. P82B96 handling an Acknowledge

The edge F represents the falling edge of the internally generated clock (SCL) that requests the Acknowledge. The actual timing of this edge as appears on the bus is F*. The internally generated rising edge of the clock is represented by R while again the real timing of the effective voltage waveform on the bus line is shown as R*. After the important system

delays, the received Acknowledge 'low' is shown. For true I²C operation the set-up time is measured with respect to R*, but where the processor timings are uncertain is safer to work with respect to R.

The corresponding diagram for the P82B96 reading a data 'hi' from a slave is shown in Figure 2.

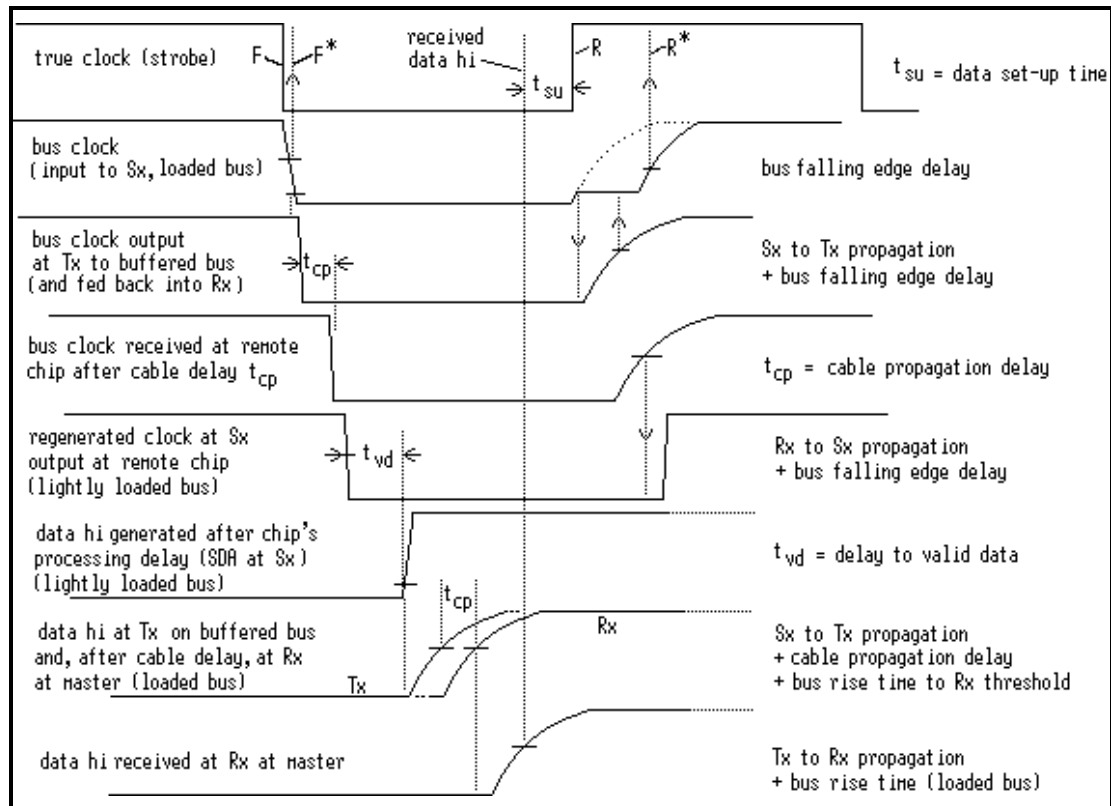


Figure 2. P82B96 handling a data low-to-high transition on SDA

For a Standard mode system with a minimum clock low period of 4.7 microseconds the set-up time is 250 ns.

The following device propagation delay times are almost independent of the application. Note however that they are specified independently of the bus rise and fall times that are determined by bus pull-ups and capacitive loading.

Low to high for PCA9515/6	55 ns
High-to-low for PCA9515/6	115 ns
Sx to Tx for P82B96 (both edges)	100 ns
Rx to Sx for P82B96 (both edges)	300 ns
Sx to Lx for P82B715 (both edges)	400 ns*
Lx to Sx (both edges) for P82B715	0 ns*

*Note the 400 ns propagation delay results, in practice, in delaying the falling or rising edge of the Sx input by this value. The input and output pins are tied with a 30 Ω resistor so they always have the same logic voltage level to within 100 mV. The input pin cannot rise/fall until the buffered bus output pin rises/falls. The signal path Lx to Sx is just via the internal 30 Ω resistor. There is no active buffer action and hence no propagation delay.

Wiring:

Where specific data is not known, signal propagation in plastic insulated cables may be estimated at 5 ns/meter.

There are specified maximum limits for the I²C bus rise and fall times as follows:

	Standard Mode (100 kHz)	Fast Mode (400 kHz)
Bus fall time	300 ns	300 ns
Bus rise time	1000 ns	300 ns

Some examples of the current datasheet delay time to valid data or valid Acknowledge are given here. See Appendix 7 for the measured values.

	Delay to Ack	Delay to data	Clock speed
PCF8570		3.4 μ s	100 kHz
PCA9540, PCA9544	1 μ s	1 μ s (h/l)	100 kHz and 400 kHz
PCA9555	3.45 μ s		100 kHz
PCA9555	0.9 μ s		400 kHz

The ‘shortcut’ approach to long bus applications

Often the delay time information for older I²C devices will not be published or the exact ‘read’ time for the master microcontroller may not be accurately known. In such cases the following conservative approach is recommended.

1. Select components that are specified for conformance to the complete I²C specifications at the full bus speed, for example 100 kHz for Standard mode. So avoid ‘bit-bash’ software used with non-I²C port micros. It is then certain that these products are fully compatible up to their maximum bus speed and ‘clock-stretching’ will be allowed.
2. Calculate the special delays that will be present in the proposed system due to additional buffer and cable propagation delays.
3. Add the net additional delay time to the nominal full speed ‘clock low’ time and use this figure as the half-period of the maximum allowable clock speed for the proposed system.

For example, a system including two P82B96 devices, one at each end of a 50 meter long cable, with maximum allowed bus rise/fall times at the sending and receiving I²C buses and also on the connecting cable, will introduce the following ‘additional’ delays.

Note: a normal system anticipates one bus having maximum delays. In this example two new buses and their delays have been introduced.

For the clock signal to reach the remote device:

Additional fall time from half-rail to 0.65 V	20 ns approximate
Sx to Tx propagation delay at master.	100 ns
Tx fall time	250 ns (maximum allowed, typical will be 100 n)
Cable propagation delay	250 ns
Rx to Sx propagation delay at slave	300 ns
Bus fall time at slave	180 ns maximum (Sx minimum sink is 7 mA, >5.5 mA charges 400 pF)

The slave then outputs its data and that returns to the master:

Slave bus rise-time is 1 μ s, delay to Sx threshold	200 ns (bus only has to rise to 0.65 V, the Sx threshold)
Propagation delay Sx to Tx at slave	100 ns
Rise time of cable bus	1 μ s (maximum allowed I ² C spec)
Cable propagation delay	250 ns
Rx to Sx propagation delay at master	300 ns

Total additional delay	2.95 μ s (The rise time of the master bus is anticipated.)

The following compensating ‘clock stretch’ will be caused by the buffer at the master side: (see Figure 2)

Sx to Tx propagation delay at master	100 ns
Rise time on cable bus	1 μ s (maximum allowed I ² C was assumed)
Rx to Tx propagation at master	300 ns

Total	1.4 μ s

Effective additional delay introduced into system 1.55 μ s

Add 1.55 μ s to the nominal 5 μ s of 100 kHz clock yields 6.55 μ s for minimum safe half period of SCL. Full period is 13.1 μ s. So 76 kHz is a totally safe clock speed for this maximally loaded system.

Some calculation examples

1) Standard I²C devices or PCA9515/16/18 used with relatively short cables

The limitation in this case becomes the I²C bus specification of 400 pF maximum bus capacitance. As shown above in “pull-up resistor sizing”, selecting a 1.8 k Ω resistor with standard 3 mA devices can decrease the bus rise times to 610 ns for 400 pF loading. Typical cable capacitances are 100 pF/meter for coax and 50-80 pF/meter for twisted-pair phone cables. Allowing 50 pF for a couple of I²C devices and some board traces leaves 350 pF for wiring, permitting around 4 meters (13 feet) of twisted pair wiring. When selecting cable pairs, never run SDA and SCL on one pair. Run each logic signal with either V_{CC} or ground on a separate cable pair. Propagation delays will not be significant for two such runs at 100 kHz with PCA9515/16/18 at the center.

2) P82B715 used at each end of a ‘long’ cable

This buffer reduces the ‘effective’ wiring capacitance by a factor 10. So a system with small I²C nodes at each end, say 25 pF each, leaves 350 pF for ‘effective’ cable capacitance or 3500 pF actual cable capacitance. That will allow around 50 meters (160 feet) of cable and will introduce two cable delay times, 250 ns each way. The buffer delays the clock seen by the slave device by 400 ns (buffer propagation) + 250 ns (cable propagation). The return data (rising edge) is delayed 400 ns by the SDA buffer propagation + 250 ns for cable delay. Total additional delay 1.3 μ s. The SCL signal at the master is ‘stretched’ by the propagation delay of the master end buffer because the clock line rises 400 ns later. The effect is similar to charge storage in turn-off of a switching transistor. So the net additional system delay is 0.9 μ s. For total safety select a clock period 2 x 5.9 μ s or clock frequency 85 kHz maximum.

3) Very long cables

In 2) we saw that I²C bus capacitance limits theoretically restrict P82B715 to around 50 meters of cable. In practice the buffered bus side of P82B715 does not comply with I²C rules so limiting that bus capacitance to 3500 pF is not really necessary provided the I²C bus rise/fall timings can always be met. It is possible to select a pull-up on the buffered bus to achieve 1 microsecond rise times with about 1.6 times larger capacitance, say 80 meters of cable in this example.

The P82B96 provides complete buffering of the long bus and really removes all I²C restrictions about capacitance limits and even rise/fall times on this ‘isolated’ bus. The timings on this bus do not affect the timings on other buses at its Sx I/Os. The specified static/dynamic sink capability of one P82B96 permits driving a 150 Ω bus pull-up on 5 V and over 4500 pF - even within the I²C bus rise/fall time limits. When really long buses are required, then the bus pull-ups can be further reduced by adding a cheap switching transistor to its Tx output as shown in Figure 3.

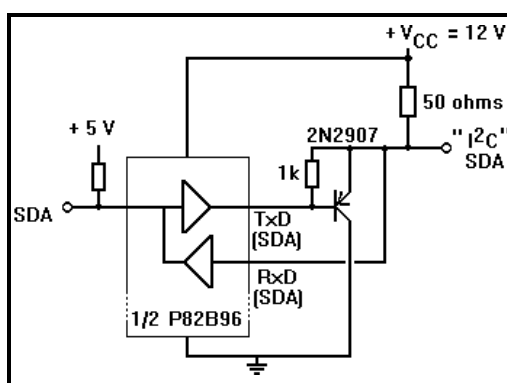


Figure 3. External Transistors to Increase P82B96 Output Drive

This circuit can be used to sink at least 300 mA (static), or 600 mA dynamic, on a very long bus. The logic levels on this bus can even be increased up to 15 V to improve immunity to induced noise. With the option to select terminating resistors to match the cable's characteristic impedance, the problem of driving capacitive loads is virtually eliminated and the only consideration becomes the system's propagation delay. As shown above in Figure 1 & 2, the fixed delays are...

For the clock signal to reach the remote device:

Additional fall time from half-rail to 0.65V	20 ns approximate
Sx to Tx propagation delay at master.	100 ns
Tx fall time	250 ns (maximum allowed, typical will be 100 ns)
Rx to Sx propagation delay at slave	300 ns
Bus fall time at slave	180 ns maximum (Sx minimum sink is 7 mA, >5.5 mA charges 400 pF)

The slave then outputs its data and that returns to the master:

Slave bus rise-time is 1 μ s, delay to Sx threshold	200 ns (bus only has to rise to 0.65 V, the Sx threshold)
Propagation delay Sx to Tx at slave	100 ns
Rise time of cable bus	1 μ s (maximum allowed I ² C spec)
Rx to Sx propagation delay at master	300 ns

Total 'buffer' delay	2.45 μ s (The rise time of the master bus is anticipated.)

The following compensating 'clock stretch' will be caused by the buffer at the master side: (see Figure 1)

Sx to Tx propagation delay at master	100 ns
Rise time on cable bus	1 μ s (maximum allowed I ² C was assumed)
Rx to Tx propagation at master	300 ns

Total	1.4 μ s

Effective buffer delays introduced into system 1.05 μ s

Add cabling delays at 10 ns / meter (measured between any two I²C devices) and decrease the maximum clock frequency as before.

For example:

- 100 meter cable, 1 μ s delay, total 2.05 μ s, clock half-period 7.05 μ s, max clock = 71 kHz
- 1000 meter cable, delay = 10 μ s, total delay 11.05 μ s, half-period 16.05 μ s, max clock = 31 kHz

A point to watch when driving very long cables will be the signal attenuation caused by the DC resistance of the wires, especially when working with very low pull-up resistors. The signal voltage drops should be checked to ensure the voltage thresholds (nominal half device supply voltage, limits 42/58% of V_{CC}) will always be crossed with reasonable design margins.

APPENDIX 2 – Using P82B715 to increase the total allowable bus capacitance

Normally, P82B715 is used only to reduce the effects of WIRING capacitance and the total I²C bus capacitance limit remains 400 pF. However, if there is no large wiring capacitance component in the system then P82B715 can be used to increase the total I²C bus system capacitance to around 3000 pF as shown in Figure 1.

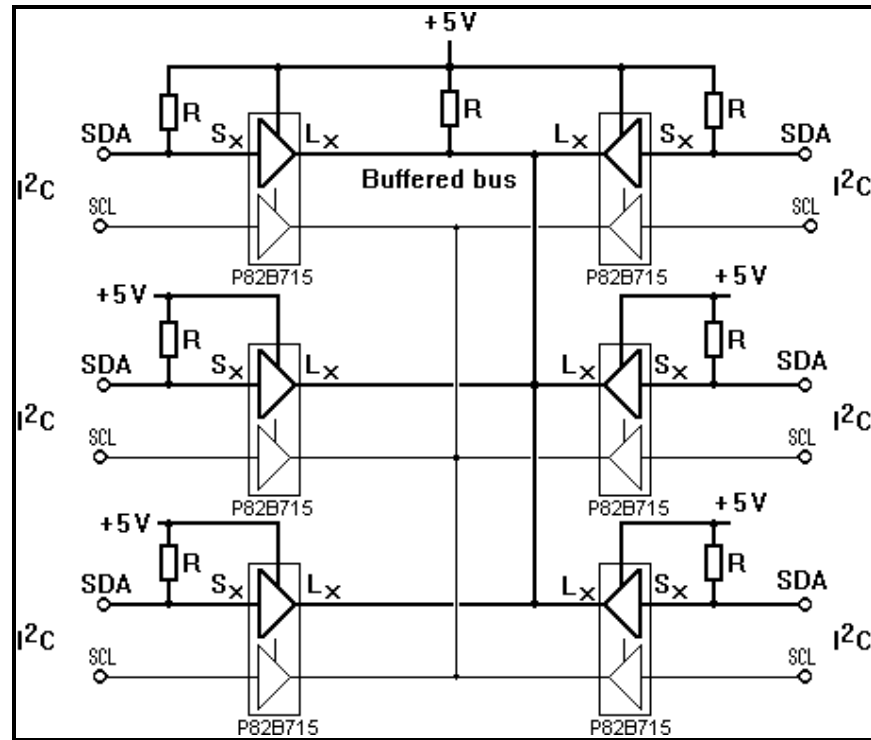


Figure 1. 3000 pF (bus capacitance loading not shown for clarity)

In this special application the total system capacitance is divided into a number of segments with roughly equal capacitance per segment. Then the pull-ups for each section can be chosen to be equal (just to simplify the calculations). The effective pull-up load on any one of the I²C buses is equal to the local pull-up in parallel with all the other pull-ups multiplied by 10. So in this example of 6 buses, and some system wiring, all having roughly equal capacitance and pull-ups, the total load = R in parallel with R divided by 6 (for 6 parallel resistors) and multiplied by 10 (for buffer action).

That is R in parallel with R/0.6 or $R/1.67R = 0.63R$.

Now the minimum allowed pull-up (5 V) is 1700 ohms therefore $0.63R = 1700 \Omega$ or $R = 2.7 \text{ k}\Omega$. For a bus rise-time of 1.18 μs (see Paragraph "Pull-up Resistor sizing" page 10), the allowable capacitance with 2.7 k Ω is 430 pF. But the I²C specification limit is 400 pF, so in this example with 7 segments, this system allows a total capacitance of $7 \times 400 \text{ pF} = 2800 \text{ pF}$.

This arrangement can prove useful in systems that include TTL logic levels because P82B715 does not affect any of the logic thresholds in the system. It should be noted that failure of any V_{CC} supply to any of the devices in this system will result in holding the bus low because P82B715 has internal diodes from all I/Os to V_{CC}. All other buffers/extenders (i.e., PCA9511/12/13/14/15/16/18 and P82B96) open circuit their I/Os if their supply is removed.

APPENDIX 3 - Operating P82B715 on lower supply voltages

The following parameters are guaranteed down to $V_{CC} = 3V$

P82B715 CHARACTERISTICS, $V_{CC} < 4.5V$

At $T_{amb} = +25^{\circ}C$ and $V_{CC} = 5V$, unless otherwise specified.

SYMBOL	PARAMETER	LIMITS			
		min	typ	Max	units
Power Supply					
V _{CC}	Supply voltage (operating)	4.5	—	12	V
I _{CC}	Supply current	—	16	—	mA
I _{CC}	Supply current at V _{CC} = 12 V	—	22	—	mA
I _{CC}	Supply current, both I ² C inputs LOW both buffered outputs sinking 30 mA	—	28	—	mA
Drive currents					
I _{SX} , I _{SY}	Output sink on I ² C bus, V _{CC} > 3 V V _{SX} , V _{SY} LOW = 0.4 V V _{LX} , V _{LY} LOW on Buffered bus = 0.3 V (and I _{LX} , I _{LY} = 3 mA, buffer is passive)	3	—	—	mA
I _{LX} , I _{LY}	Output sink on Buffered bus V _{LX} , V _{LY} LOW = 0.4 V V _{SX} , V _{SY} LOW on I ² C bus = 0.3 V	30	—	—	mA
Additional for V _{CC} > 3V					
I _{LX} , I _{LY}	Output sink on Buffered bus, V _{CC} > 3V V _{LX} , V _{LY} LOW = 0.4 V to 1.5 V I _{SX} , I _{SY} sinking on I ² C bus < 4 mA	24	—	—	mA
I _{LX} , I _{LY}	Output sink on Buffered bus, V _{CC} > 3V V _{LX} , V _{LY} LOW = 1.5 V to V _{CC} I _{SX} , I _{SY} sinking on I ² C bus = 7 mA	24	—	—	mA
Input Currents					
I _{SX} , I _{SY}	Input current from I ² C bus when I _{LX} , I _{LY} sink on Buffered bus = 30 mA	—	—	3	mA
I _{LX} , I _{LY}	Input current from Buffered bus (V _{CC} > 3V) when I _{SX} , I _{SY} sink on I ² C bus = 3 mA (note: buffer is passive in this test)	—	—	3	mA
I _{LX} , I _{LY}	Leakage current on Buffered bus, V _{CC} = 3 to 12V V _{LX} , V _{LY} = V _{CC} and V _{SX} , V _{SY} = V _{CC}	—	—	200	μA
Impedance Transformation					
Z _{in} / Z _{out}	Input/Output impedance	8	10	13	

APPENDIX 4 - The special logic levels and noise margins of buffer logic

There have been many attempts at making I²C buffers and opto-isolators. All result in (and most admit to) problems with glitches resulting from signal propagation delays. They all block the signal propagation in one direction to prevent latch-up. So they all produce a 'glitch' during the signal propagation time needed to clear that blocking.

To test whether a buffer violates the I²C timings (due to glitches) apply this test:

- Take one side low, take the other side low, and release the first side.
- In a proper system the first side should just stay low, held by the second side. Switching-style buffers with timing delays will all produce a 'glitch' to high on the first side before it goes low again.

To avoid glitches, one solution is to introduce '2 logic-low levels' as the means to avoid latching of the buffer. That is the approach used in the PCA9511/12/13/14/15/16/18 and P82B96. The P82B715 that does not use special logic levels provides uni-directional current amplification instead. The P82B96 uses only 'static' logic thresholds and responds only to DC levels. It has no 'memory' of previous I/O states. A 'low' at Sx will always be recognized and transmitted to Tx, but only when its DC level is below 0.65 V.

PCA9515, PCA9516 and PCA9518 feature 'dynamic' logic thresholds. In normal operation when simply passing logic signals in either direction they use conventional CMOS thresholds with typical switching at half V_{CC} and published 0.3/0.7 V_{CC} limits. However while one input is driven low and its corresponding output is driven low by the device, to around 0.5 V, the level at the output that is required to be recognized as a 'low' (for the bi-directional I²C signaling) becomes slightly lower than the 0.5 V driven by the device. This level, around 0.45 V, remains the low level input threshold UNTIL that output pin has been released and risen to above the CMOS 'high' level (typical 0.5 V_{CC} , maximum 0.7 V_{CC}).

In an application where a master has just sent 'low' data bits to a slave on the SDA line, and then tries to read back an Acknowledge 'low' bit from the slave, it is quite possible that stray bus capacitance together with a quick response from the slave, will mean the SDA line will never cross the CMOS 'high' level before being clamped 'low' again by the Acknowledge from the slave*. In this case the EFFECTIVE logic low threshold, necessary for correct reading of that Acknowledge 'low', is 0.45 V.

So the Sx input of P82B96 and all inputs of PCA9515/16/18 should be considered as having their low level input logic thresholds at around 0.65 V and 0.45 V respectively. While these thresholds are guaranteed by the I²C static low of 0.4 V (maximum) at 3 mA, and are the compromise that allows designing a non-latching bi-directional buffer, it should be noted that the noise margin is relatively small. Care must be taken when designing with these I/Os to ensure the system noise will not exceed this margin.

* [Acknowledge from a slave happens after the 8th SCL clock pulse. The falling edge of the 8th SCL clock pulse is the signal to a slave to put an acknowledge 'low' on SDA before the rising edge of the 9th SCL clock pulse. The slave response can be very fast, normally there is only a simple counter of the SCL pulses and the falling edge of SCL can result in an almost immediate acknowledge 'low' on the slave's SDA. Consider the master had sent a data 'low' (via PCA9515) so on the 8th clock pulse SDA0 was low and so was SDA1. SDA0 will be released by the master at (or maybe just after) the falling edge of the 8th SCL pulse and will start to rise towards 'high'. If SDA1 has near maximum bus loading it will rise 'slowly' with some time-constant, let's say 1 μ s. Because SDA1 is the 'output', it is not recognized as 'low' by PCA9515 during this slew towards V_{CC} , until it has risen above a CMOS 'high' level. The only level that will be recognized as low (before the line reaches 'high') is one that is below the original SDA1 output level, i.e. a true I²C low, about 0.4 V. So when a slave places its acknowledge 'low' onto the SDA line, if this is not maintained below about 0.4 V then this Acknowledge from the slave will not be maintained as a 'low' at SDA0. Any noise affecting the 0.4 V level can appear as pulses to 'high' at SDA0. With some delay caused by capacitive loading on SDA1 and a fast acknowledge from the slave (as must be expected) it is most probable that the acknowledge will NOT get transmitted unless SDA1 can be held below about 0.4 V.]

APPENDIX 5 - Using P82B96 to create a multi-drop “distribution” bus using 12 V logic levels

When distributing I²C signals over cables run between separated pieces of equipment that can have slightly different ground references it is helpful to increase the logic voltage swings and so increase noise margins.

The P82B96 can be used to distribute I²C signals over a ‘distribution bus’ that has all the same characteristics as I²C but is operated at lower impedance and with higher logic voltages.

Figure 1 shows one example where the logic voltages are set at 12 V. The Rx input of P82B96 switches very closely at half the supply rail, so in this case 6 V. The Tx output pulls down within 100 mV of its ‘local ground’ voltage leaving a typical noise margin of 6 V. So even if the ground references are 2 V or 3 V different, and external noise sources induce 4 V p/p onto the wiring, the logic signal integrity is still preserved.

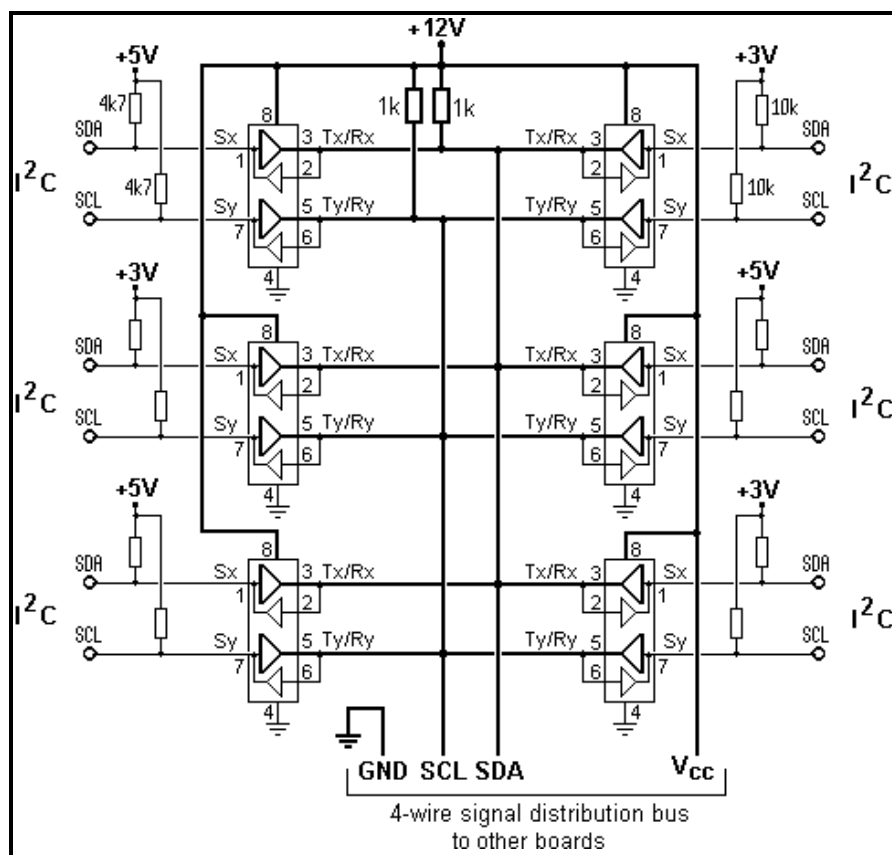


Figure 1. Multi-drop distribution of I²C signals

Note that it is necessary to connect the V_{CC} of all the P82B96s to the 12 V supply because it is V_{CC} that sets the logic thresholds at Rx. Just like CMOS, the typical threshold is V_{CC} / 2 but because P82B96 uses a bipolar process this threshold is very accurate – typically within 2% and with guaranteed limits 42% and 58% of V_{CC}. It can be convenient to use a flat 4-conductor cabling or two ‘twisted pairs’ as widely used for telephone wiring. In this case one logic signal is ‘paired’ with one of the power wires. The SCL and SDA signals should not be run on adjacent conductors of a flat cable or together on one twisted pair. Because these cables are not correctly terminated by their characteristic impedance there will usually be significant ringing associated with the fast falling edges of the bus signals. To avoid exceeding the rating of the IC pins, this may need to be suppressed using Schottky clamping of the distribution bus wires to the local ground at each IC. High frequency damping using a series connection of a resistor and capacitor can also be effective. Typical values for such damping components for use on a standard (400 pF) I²C bus, say 2 m long, would be BAT54 Schottky and for the RC a capacitance in the range 50 – 150 pF in series with a resistance in the range 50 - 150 Ω.

In extreme cases the alternative non-linear network of Figure 2 can be used to permit greater damping of the falling edges without significantly slowing the rising edges.

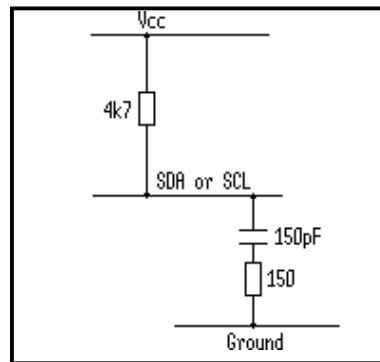


Figure 2.
Damping I²C lines
using RC termination

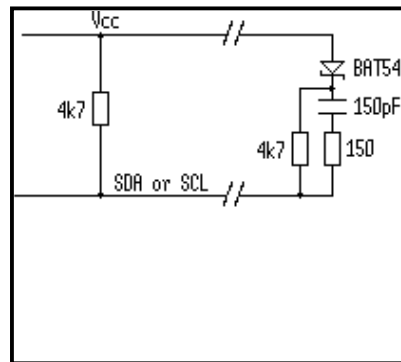


Figure 3.
Damping I²C lines
with a non-linear termination

Similar solutions can be applied on distribution buses where the bus characteristic impedances are similar but the loading effect of such networks is much less of a problem. Damping components as shown are best applied at both of the extreme ends of cable runs. When T-off lines are used then they can also benefit from damping at their ‘open’ end. Always keep in mind that the I²C signals are bi-directional so there is no single ‘send’ and single ‘receive’ end of the cables.

When sections of the system will be optionally connected (even hot-plugged) it can be an advantage to use distributed bus pull-up.

When environmental conditions can be expected to induce very large voltages or currents into the distribution wiring (for example when the wiring is, say, 50 meters long and nearby lightning strikes), they will cause significant voltage drops between the local ground potentials at each end of the wiring and extra protection will be required. Solutions include adding some series resistance into the bus lines, and applying 15 V Zener clamps to the bus lines.

Distinction should be made between protection measures that ensure the survival of the bus system components and those that ensure integrity of the data being carried. In conditions equivalent to nearby lightning strikes it is virtually impossible to ensure integrity of the data and software security methods need to be used. Often the simplest solution is to verify all data by writing and immediately reading it back as the check it was correctly delivered. When reading from a slave, a double reading technique might be used.

The operating parameters for the ‘distribution bus’ may be set by the system designer. Because it is not a conventional I²C bus it is not mandatory to apply any of the bus specifications. So, for example, the bus rise and fall times do not need to conform because they will conform again after conversion back to I²C on the Sx sides. And the capacitance limit is not set by timing or the bus pull-ups. However, unless there is a valid reason for departing from the I²C parameters, they can be used as convenient benchmarks and the distribution bus conveniently viewed as “I²C with 10x current and larger voltages”. As shown in Appendix 1, even the 10x drive capability of P82B96 is not a limiting factor because further external amplification remains an option.

After the I²C signals are delivered via the distribution bus they are locally converted back to I²C (or SMBus, etc...) compatible signals. Figure 1 shows a mix of 3 V and 5 V systems. When working with 3 V logic at Sx always select the weakest pull-up compatible with a reasonable bus rise-time.

APPENDIX 6 - Differences Between the LTC1694 and LTC1694-1

Feature list:

- LTC1694: Improves SMBus Rise Time Transition
- LTC1694-1: Improves SMBus/I²C Rise Time Transition and several devices can be used in the same bus (mentioned in the feature list and there is a chapter in this topic in the data sheet (page 7, LTC1694-1)).

Electrical characteristics:

- I_{CC} (supply current) is different:

	Min	Max	Typ	Units
LTC1694	20	60	100	mA
LTC1694-1	15	45	80	mA

Architecture LTC1694:

LTC1694 has three current sources improving the signal rising times.

Current 1 = 100 μ A (always ON) + current 2 = 175 μ A (can be ON/OFF) + current 3 = 1.925 mA (can be ON/OFF)

- Nominal mode (SMBus command in the bus):

I = 275 μ A when V(SMBus) < 0.65 V, I = 2.2 mA when V(SMBus) > 0.65 V

Negative slew rate is also controlled by the device (I = 275 μ A)

Current source also improves the low state noise margins.

- Stand-by mode (Bus idle)

I = 100 μ A

Architecture LTC1694-1:

LTC1694 has one current source improving the signal rising times.

Current 1 = 2.2 mA (can be ON/OFF)

- Nominal mode (SMBus command in the bus):

I = 0 μ A when V(SMBus) < 0.65 V, I = 2.2 mA when V(SMBus) > 0.65 V

Negative slew rate is controlled by the passive pull-up resistor.

Resistor also determines the logic low DC level.

- Stand-by mode (Bus idle):

I = 0 μ A (it is not really a stand-by mode though)

Note:

On both devices, the active pull-up remains ON until the voltage on the SMBus is within 0.5 V of V_{CC} and/or the slew rate drops below 0.2 V/us.

Pull-up resistors:

- LTC1694 does not require passive pull-up resistors since there is an internal current source to V_{DD}.
- LTC1694-1 requires passive pull-up resistors to V_{DD}.

Number of devices in the same bus:

- LTC1694 data sheet does not talk about the possibility of having several devices in the same bus.
- Several LTC1694-1 can be present in the same bus.

APPENDIX 7 - Typical response times to 'data valid' on SDA, measured from the falling edge of SCL

While we do not like publishing TYPICAL data, we think these measurements are the only data that a customer will find to be able help design systems with delays caused by buffers or long cables. They enable calculations about HOW fast the P82B96 (or other buffers) may be allowed to clock in a purpose-designed system where all device timings are known. For example, some customers want a buffered or opto-coupled bus running at 400 kHz. It maybe possible PROVIDED the timing of all the system devices is known. It is rare to find specifications about Micros and PCF8584, and even slaves are not quoted. Unless otherwise stated in data sheets it must be assumed that a Master device specified as conforming to I²C specifications will include clock-stretch capability.

The following table shows the times measured for these devices to put out data or an acknowledge on their SDA, measured from the falling edge of the SCL pulse that requests this response. The maximum value of these delays is one of the critical factors that set maximum bus speed. In all cases they are the DELAY times to the start of bus change. Rise/falls are NOT included. Bus clock speed used was about 140 kHz (limit for Win-I2CNT on the testing personal computer).

The measurements confirm there would be no expected problems for most devices, even with very long wiring at 100 kHz.

The simulation figures for PCA9564 also show it is useable in a system with P82B96 buffering without any loss of speed (Both PCA9564, and whole system, limit is 330 kHz)

These are measured values only and are not guaranteed in the current data sheets.

All times are in ns.

	Data hi-to-low	ACK	Data low-to-hi
PCA9559	175	160	150
PCF8575	200	200	200
PCA9556	700	700	700
NE1617	700	700	700
PCF8593	350	<400	350-375
PCA9540		<125	<125
PCA9544		<125	125
PCA8550	125	130	130
PCK2001M		100	0
PCF8563 ⁽¹⁾	600	600	600
PCF8583 ⁽¹⁾	600	600	600
SAA1064 ⁽²⁾	3200	4200	3200
PCF8574/A ⁽¹⁾	500	500	500
PCD3312P ⁽¹⁾	500	500	500
PCA9564 ⁽³⁾	363	363	275

Notes:

⁽¹⁾- Measured on older demoboard, and less accurately.

⁽²⁾- SAA1064 has no published operating bus speed or signal delays. Based on this measurement it cannot be recommended for applications at the full 100kHz with a symmetrical clock if Ack is required by the system.

⁽³⁾- Not measured, but based on design simulations.