APPLICATION NOTE (preliminary 1/30/02)

AN262
PCA954X FAMILY OF I²C / SMBus MULTIPLEXERS and SWITCHES

Paul Boogaards - Philips Semiconductors Field Application Engineer
Jean-Marc Irazabal - Philips Semiconductors PCA Technical Marketing Manager
Steve Blozis - Philips Semiconductors PCA International Product Manager

Specialty Logic Product Line
Logic Product Group

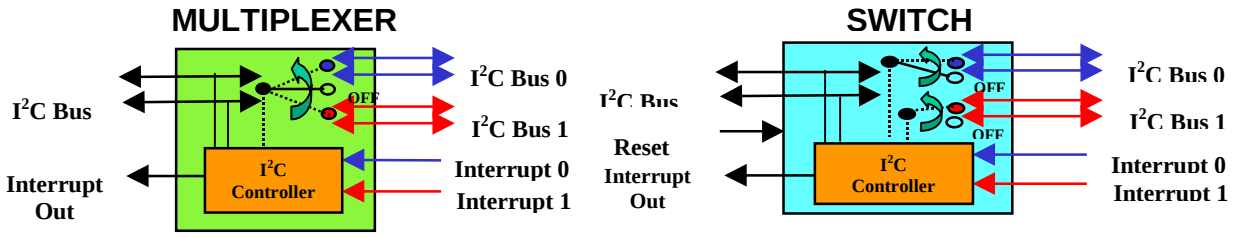
Philips Semiconductors

TABLE OF CONTENTS

OVERVIEW

Description

The Philips family of Multiplexers and Switches consists of bi-directional translating switches controlled via the I^{2}C or SMBus to fan out an upstream SCL/SDA pair to 2, 4 or 8 downstream channels of SCx/SDx pairs. The Multiplexers allow only one downstream channel to be selected at a time, while the Switches allow any individual downstream channel or combination of downstream channels to be selected, depending on the content of the programmable control register. Once one or several channels have been selected, the device acts as a wire, allowing the master on the upstream channel to send commands to devices on all the active downstream channels, and devices on the active downstream channels to communicate with each other and the master. External pull-up resistors are used to pull each individual channel up to the desired voltage level. Combined interrupt output and hardware reset input are device options that are featured.



Applications

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

I^{2}C Multiplexing - Some specialized devices only have one I^{2}C or SMBus address and sometimes several identical devices are needed in the same system. The multiplexers and switches split the I^{2}C bus into several sub-branches and allow the I^{2}C master to select and address one of multiple identical devices, in order to resolve address conflict issues.

Voltage Level Shifting - Many I^{2}C and SMBus devices operate at different voltage levels but need to operate on a common bus. The multiplexers and switches allow translation between 1.65 V and 5.5 V. So, for example, a 5 V I^{2}C master on the upstream channel can communicate with a 3.3 V (non 5 V tolerant) SMBus device on channel 0 and a 2.5 V I^{2}C device on channel 1. 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 that 1.8 V or 2.5 V or 3.3 V devices can communicate with 5 V devices without any additional protection. All I/O pins are tolerant up to 6.0 V. The Switches are best for this application since multiple downstream channels can be active at the same time.

Capacitive Load Sharing - Adding more I^{2}C and SMBus devices on the bus may exceed the 400 pF limitation. The multiplexers and switches can isolate devices that are not currently needed to reduce the overall system loading and maintain the total system load below 400 pF. When active, the channels act as a wire and the cumulative capacitive loading of the upstream channel and all active downstream channels must be considered.

Features

Interrupt Function - Interrupt inputs, one for each of the downstream channels, are provided as an option on both the Multiplexers and Switches. The single interrupt output acts as an AND of the interrupt inputs and is not latched.

Hardware Reset - An external active low hardware reset pin (/RESET) is provided on the Switches in addition to the Power On Reset (POR) feature found on both the Multiplexers and Switches. Either /RESET or POR resets the downstream channels to the default state of no channels selected. The reset feature is useful should a downstream rouge device lock up the bus and the master loses the ability to communicate. The master can use the reset to restore

communication within the upstream channel and then selectively try to restore communication with the downstream channels without having to cycle power to the equipment or to other I^2C bus devices.

Hardware Pins - Up to three hardware pins (A0, A1, A2) are provided to change the I^2C address and allow up to eight PCA954X devices to share the same I^2C/SMBus.

DEVICE NAME	MULTIPLEXER (In/Out)	SWITCH (In/Out)	FEATURES			PACKAGES			
			# of ADDRESSES	INTERRUPT (In/Out)	HARDWARE RESET	PIN COUNT	SO (narrow)	SO (wide)	TSSOP
PCA9540	1-2		1			8	D		DP
PCA9542	1-2		8	2-1		14	D		PW
PCA9543		1-2	4	2-1	●	14	D		PW
PCA9544	1-4		8	4-1		20		D	PW
PCA9545		1-4	4	4-1	●	20		D	PW
PCA9546		1-4	8		●	16	D		PW
PCA9548		1-8	8		●	24		D	PW

Table 1. PCA954X Features

Operating Characteristics

- I^2C and SMBus compatible
- ESD protection exceeds 2000 V HBM per JESD22-A114, 150 V MM per JESD22-A115 and 1000 V CDM per JESD22-C101
- JESDEC Standard JESD78 Latch-up testing exceeds 100 mA
- Manufactured in high volume BiCMOS process
- 2.3 V to 5.5 V operating voltage
- 6.0 V tolerant I/Os
- -40 ºC to 85 ºC operating temperature range
- 0 kHz to 400 kHz operating frequency

Device Pinout

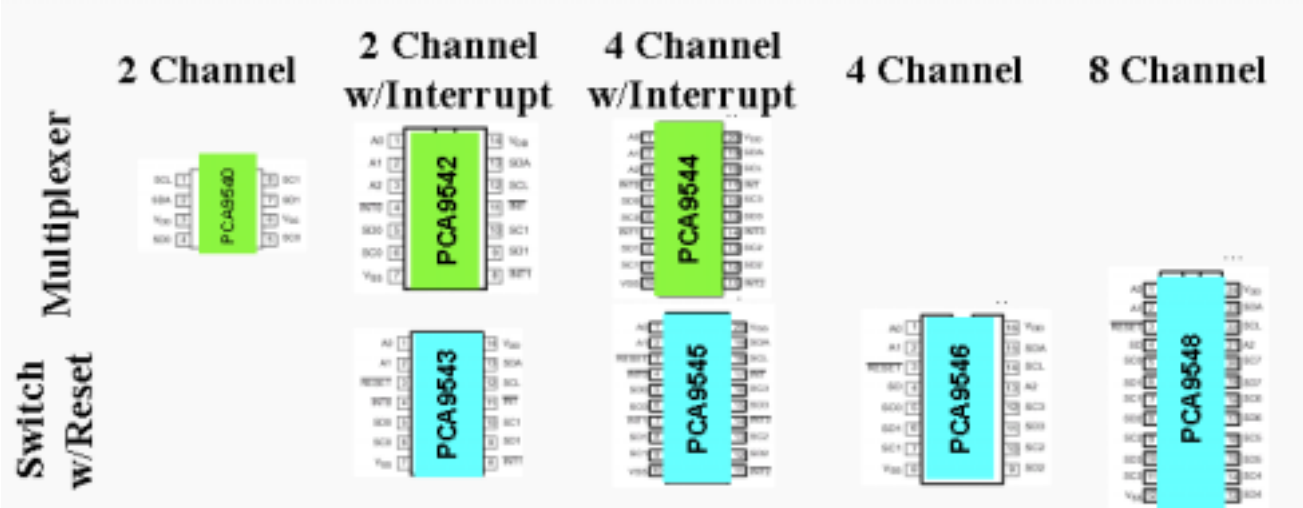


Table 2. PCA954X Pin Out

Ordering Information

Package	Container	PCA9540	PCA9542	PCA9543	PCA9544	PCA9545	PCA9546	PCA9548
SO	Tube	PCA9540D	PCA9542D	PCA9543D	PCA9544D	PCA9545D	PCA9546D	PCA9548D
	T & R	PCA9540D-T	PCA9542D-T	PCA9543D-T	PCA9544D-T	PCA9545D-T	PCA9546D-T	PCA9548D-T
TSSOP	Tube	Not available	PCA9542PW	PCA9543PW	PCA9544PW	PCA9545PW	PCA9546PW	PCA9548PW
	T & R	PCA9540DP-T	PCA9542PW-T	PCA9543PW-T	PCA9544PW-T	PCA9545PW-T	PCA9546PW-T	PCA9548PW-T

Table 3. PCA954X Ordering Information

Data Sheets and IBIS Models

Data sheets and IBIS models can be downloaded from www.philipslogic.com/i2c

TECHNICAL INFORMATION

Block Diagram

The PCA954X devices are bi-directional translating Multiplexers and Switches, controlled via the I²C bus. The block diagram is shown in Figure 1. The SCL/SDA upstream pair fans out to downstream pairs, or channels. The number of downstream pairs is device dependent. Exactly the same I^2C signals on the upstream channel are passed onto all the downstream channels without amplification and the 400 pF I^2C bus limitation must be observed for the upstream channel and all active downstream channels. Pull-up resistors are **REQUIRED** on all upstream and downstream channels.

I^2C commands from the bus master on the upstream channel or any active downstream channel can turn on or turn off any channel. The channel status is changed when the stop command is sent. A master on an unconnected downstream channel (not active) cannot send commands to the device. The Multiplexers and Switches are basically the same with the only difference being within the I^2C Bus Controller. The I^2C Bus Controller in the Multiplexer activates only one channel at a time while the I^2C Bus Controller in the Switch activates as many channels as there are available, in any combination, as determined by the contents of the programmable control register.

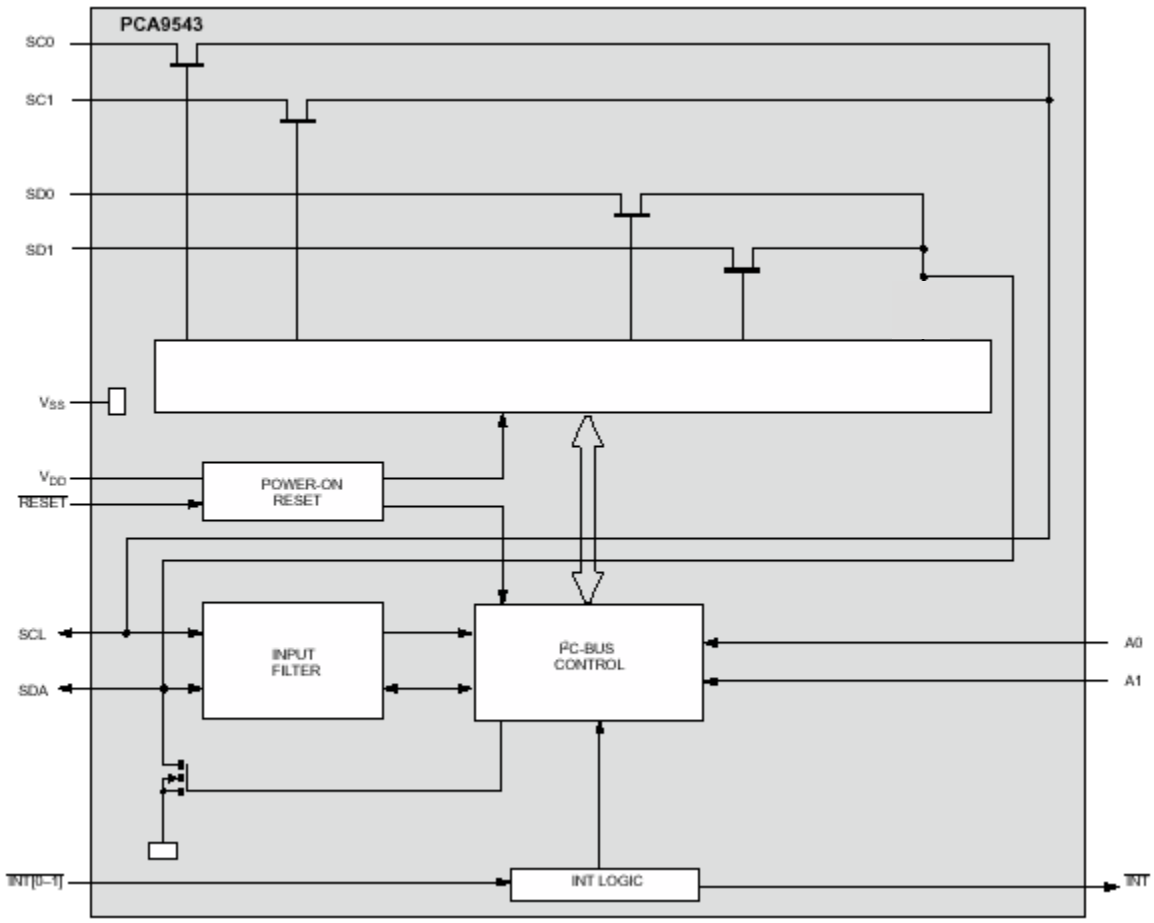


Figure 1. PCA9543 Block Diagram

All PCA954X devices offer a built-in Power On Reset (P.O.R.) circuit block, which ensures all downstream channels are deactivated while the device is being powered up. The outputs are held in a high impedance state that supports hot insertion.

The PCA954X Switches offer an external hardware pin reset capability. An active low asynchronous reset will deactivate all the downstream channels, as well as reset the I²C state machine. This is useful when a rogue device on one of the downstream channels is holding the bus low. It allows the master to deactivate all downstream channels and regain control of the upstream bus by simply asserting the reset without having to cycle power to the equipment and performing a POR to all I^2C devices. A 100 kΩ (or sized to the master's output capacity) pull up resistor is required to hold the reset pin high to allow normal operation of the device.

Interrupt input is provided for each SCx/SDx downstream pair on select PCA954X devices. An active low open-drain interrupt output is also provided and acts as an AND of the interrupt inputs and is not latched.

Up to three external hardware address pins are provided that allow modification of the I^2C address, which allows up to eight PCA954X devices to operate on the same I^2C bus. The address pins must be held high to V_{DD} or low to ground. It is recommended they be held high through a 10 kΩ pull –up resistor to limit current flow in case of a short but if desired they can be tied directly to V_{DD}.

The PCA954X devices can operate with a supply voltage range anywhere between 2.3 V to 5.5 V and the SDx/SCx and interrupt pins are all tolerant to voltages up to 6.0V. The pass gates of the PCA954X 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 parts can communicate with 5 V parts without any additional protection. External pull-up resistors pull the bus up to the desired voltage level for this channel.

I²C Communications

PCA954X devices support both standard mode (100 kHz) and fast mode (400 kHz) I^2C protocols. Once the channels have been selected and the stop command sent, the PCA954X devices act as a wire and will support up to 400 kHz I^2C protocol throughput. A standard I^2C communication between a master controller and a PCA954X device contains the following sequence:

- A Start condition
- A 8-bit word with the following information:
  a) PCA954X device addressing. 7 bits (as shown in Table 4) compose the address.
  b) The 8th bit (LSB) is the Read (LSB at "1") or Write (LSB at "0") instruction
- Acknowledge from the slave (PCA954X addressed device)
- If a Write instruction is requested, the next 8-bit word is the Control register. It contains channel selection information. This register is explained in the Tables 2 and 3 below.
- If a Read instruction is requested, the master controller turns to a master receiver and the slave PCA954X device turns to a slave transmitter. Interrupt status (if the device has this feature) and channel selection status (2 or 4 LSB or the entire register depending on the device) are then provided to the master controller.
- If the previous 8-bit word was a Write, the slave PCA954X will send an Acknowledge to the master controller.
- If the previous 8-bit word was a Read, the slave PCA954X will not send an Acknowledge to the master controller.
- A Stop condition. When this condition will be detected by the PCA954X, the new channel configuration will be generated (if requested in the previous I^2C communication).

Device Type	**I^2C Address**							**R/W**
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PCA9540	1	1	1	0	0	0	0	1/0
PCA9542	1	1	1	0	A2	A1	A0	1/0
PCA9543	1	1	1	0	0	A1	A0	1/0
PCA9544	1	1	1	0	A2	A1	A0	1/0
PCA9545	1	1	1	0	0	A1	A0	1/0
PCA9546	1	1	1	0	A2	A1	A0	1/0
PCA9548	1	1	1	0	A2	A1	A0	1/0

Table 4. PCA954X Addresses

Note:
- A0, A1 and A2 are hardware programmable input pins that are connected to either V_{DD} (logic level 1) or GND (logic level 0).
- Up to eight PCA954X devices can be attached to the same I^2C bus.

The PCA954X multiplexers contain one Control register, which can be written to or read from. When writing to this register, the lower bits or the entire register (depending on the device) determine the active channel(s). At Power-up and when a Reset is initiated (for devices offering this feature), all channels are deactivated and no channels are active.

Table 5 describes the channel selection for the 2 channel PCA9540/42/43 and the 4 channel PCA9544 Multiplexer.

Control Register								**Device Channel Selection**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**PCA9540/42**	**PCA9543**	**PCA9544**
x	x	x	x	x	0	0	0	None	None	None
x	x	x	x	x	0	0	1	None	Channel 0	None
x	x	x	x	x	0	1	0	None	Channel 1	None
x	x	x	x	x	0	1	1	None	Channel 0 & 1	None
x	x	x	x	x	1	0	0	Channel 0	None	Channel 0
x	x	x	x	x	1	0	1	Channel 1	Channel 0	Channel 1
x	x	x	x	x	1	1	0	None	Channel 1	Channel 2
x	x	x	x	x	1	1	1	None	Channel 0 & 1	Channel 3

Table 5. 2 Channel Multiplexer and Switches and 4 Channel Multiplexer Channel Selection

Table 6 describes the channel selection for the 4 channel PCA9545/46 (in grey) and 8 channel PCA9548 Switches. Only the 8 channel PCA9548 will respond to the channels in yellow (in addition to the channels in grey).

Control Register								Active Channels							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	0	0	0	0	0	0								
0	0	0	0	0	0	0	1	◆							
0	0	0	0	0	0	1	0		◆						
0	0	0	0	0	0	1	1	◆	◆						
0	0	0	0	0	1	0	0			◆					
0	0	0	0	0	1	0	1	◆		◆					
0	0	0	0	0	1	1	0		◆	◆					
0	0	0	0	0	1	1	1	◆	◆	◆					
0	0	0	0	1	0	0	0				◆				
0	0	0	0	1	0	0	1	◆			◆				
0	0	0	0	1	0	1	0		◆		◆				
0	0	0	0	1	0	1	1	◆	◆		◆				
0	0	0	0	1	1	0	0			◆	◆				
0	0	0	0	1	1	0	1	◆		◆	◆				
0	0	0	0	1	1	1	0		◆	◆	◆				
0	0	0	0	1	1	1	1	◆	◆	◆	◆				
0	0	0	1	0	0	0	0					◆			
0	0	0	1	0	0	0	1	◆				◆			
…	…	…	…	…	…	…	…								
0	0	1	0	0	0	0	0						◆		
0	0	1	0	0	0	0	1	◆					◆		
…	…	…	…	…	…	…	…								
0	1	0	0	0	0	0	0							◆	
0	1	0	0	0	0	0	1	◆						◆	
…	…	…	…	…	…	…	…								
1	0	0	0	0	0	0	0								◆
1	0	0	0	0	0	0	1	◆							◆
…	…	…	…	…	…	…	…								

Table 6. 4 Channel and 8 Channel Switch Channel Selection

Note:

- Several channels can be enabled at the same time. For example, in the PCA9548, “10011101” means that channels 0,2,3,4 and 7 are enabled and channels 1,5 and 6 are disabled.

The control register can also be read in order to determine which channel(s) is (are) enabled or to check that a previous switch command has been correctly interpreted by the slave. For devices offering an Interrupt capability, a reading of the control register after Interrupt detection from the master controller will also determine which downstream channel(s) generated an Interrupt signal. A more detailed description of the interrupt function follows

Interrupts

Devices offering the Interrupt capability provide the following pins:

- An active low Interrupt input pin for each SCx/SDx downstream pair.
- An open-drain Interrupt output. This signal acts as an AND of the interrupt inputs and is not latched.

When no Interrupt is present (all the Interrupt inputs are at logic level High), then the Interrupt output is also at logic Level High. When any of the Interrupt inputs is logic level Low, then the Interrupt output is also Low. The control register reflects the inverted state of the interrupt inputs (as shown in Table 7). When the interrupting input signal goes away (returns to a High state), the output will also return to a High state and the device does not keep in memory what caused the interrupt (the control register interrupt bits return to '0'). The PCA9542/43 only have interrupt channels 0 and

1 and don't include any of the interrupts in channel 2 and 3 (yellow highlighted columns). The PCA9544/45 have interrupt channels 0 through 3. "◆" in the Interrupt channel columns indicates that there is an interrupt.

Control Register								Interrupt Channel			
7	6	5	4	3	2	1	0	0	1	2	3
0	0	0	0	x	x	x	x				
0	0	0	1	x	x	x	x	◆			
0	0	1	0	x	x	x	x		◆		
0	0	1	1	x	x	x	x	◆	◆		
0	1	0	0	x	x	x	x			◆	
0	1	0	1	x	x	x	x	◆		◆	
0	1	1	0	x	x	x	x		◆	◆	
0	1	1	1	x	x	x	x	◆	◆	◆	
1	0	0	0	x	x	x	x				◆
1	0	0	1	x	x	x	x	◆			◆
1	0	1	0	x	x	x	x		◆		◆
1	0	1	1	x	x	x	x	◆	◆		◆
1	1	0	0	x	x	x	x			◆	◆
1	1	0	1	x	x	x	x	◆		◆	◆
1	1	1	0	x	x	x	x		◆	◆	◆
1	1	1	1	x	x	x	x	◆	◆	◆	◆

Table 7. Interrupt Table

Note:

- Interrupt inputs (values in the control register) and output are not latched
- Interrupt register cannot be changed programmatically (Read-only information).

The main function of the Interrupt feature is to let the upstream master know that it needs to service one of the downstream buses. By reading the control register, it is easy for the master to determine which of the downstream buses requires servicing.

Command Sequencing

Although the PCA954X multiplexers and switches are very simple to use, **it must be understood that a <u>STOP condition</u> must be generated before the channel is switched.**

For example, the sequence shown in Figure 2 sends a command to the multiplexer to switch to channel 1, followed by a command to read the Philips PCA9556 (the S is a START or RESTART condition while the P is a STOP condition). Although this is a valid I²C sequence, it could give unexpected results if you are not aware that you are reading the Philips PCA9556 from the original channel, not channel 1 because the multiplexer will switch to channel 1 **ONLY** after the STOP condition has been sent.

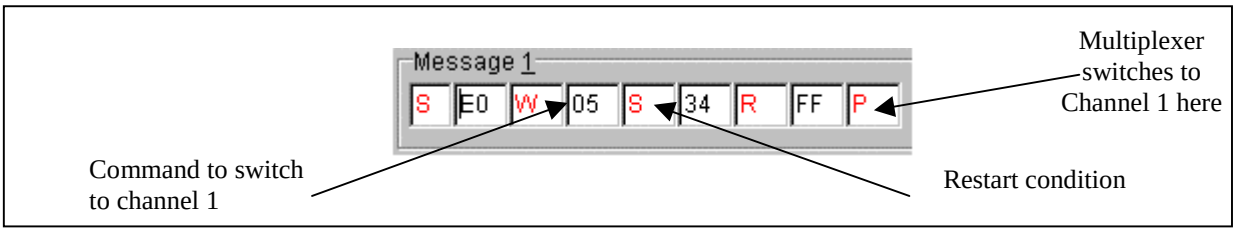
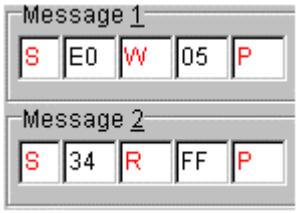


Figure 2. Channel Selection

In order to get read valid data from the Philips PCA9556 located on channel 1, the following message should be sent:



This sequence switches the multiplexer to Channel 1 **AFTER** the Stop condition

This sequence reads data from a Philips PCA9556 on Channel 1

Figure 3. Correct Channel Selection

To ensure that a communication with the correct downstream channel is initiated, the address information must be sent first, followed by the desired channel (in this case channel 1), and then followed by a Stop command, as shown in Figure 3. Now the I²C messages are being transmitted through the correct channel.

Note that once the master controller has configured the PCA954X, the slave device behaves like a wire and will keep the programmed configuration until the master controller addresses it again in order to change the configuration. Each I²C message addressed to another device connected to a downstream channel will pass through PCA954X device without any need to access and configure the PCA954X device again (transparent switch behavior).

Voltage Clamping

The multiplexers contain pass transistor, which are very fast and have very low on-resistance. When the switch is enabled and the input voltage is low, the output is also low and the switch has a typical on-resistance around 25Ω. As the input voltage rises, the output voltage should track the input voltage closely until it reaches a value approximately 1V below V_{DD}. At this voltage, the output is clamped (V_{clamp}). This phenomenon can be seen in Figure 4 below.
The clamping voltage will be somewhat lowered by a load on the output (shown with open output).

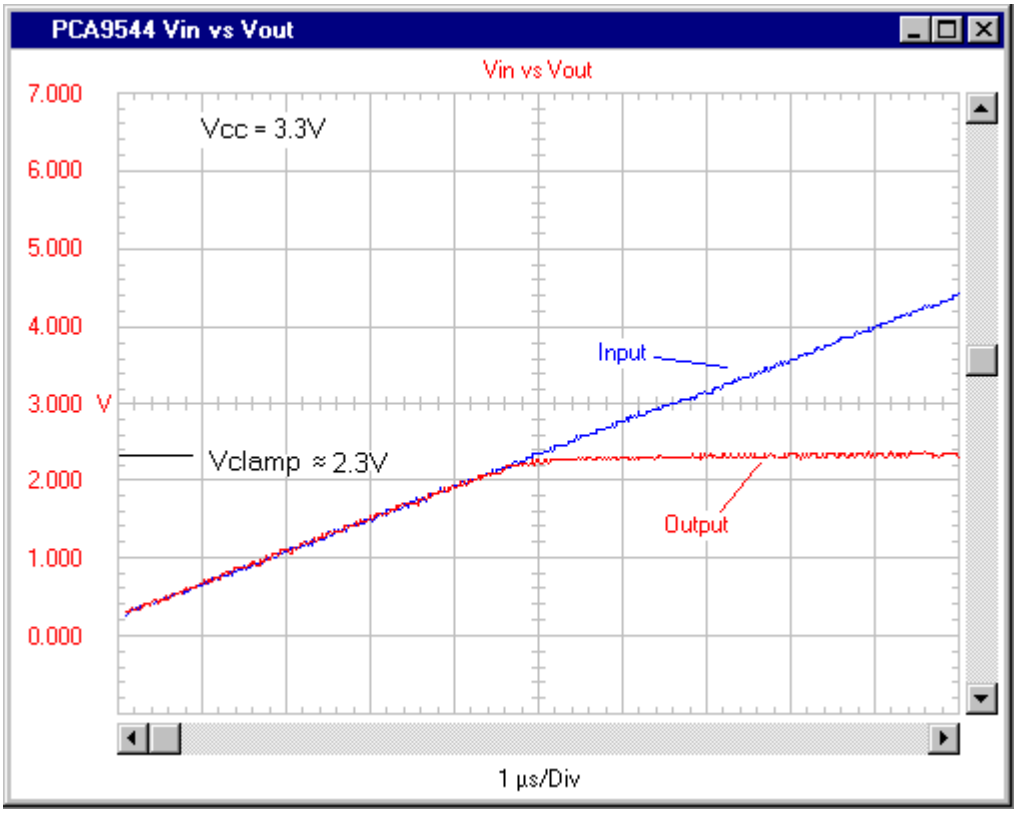


Figure 4. Vout Vs Vin

Voltage Translation

The voltage clamping can be used to our advantage when we need to communicate between two I²C voltage levels. The PCA954x will clamp the voltage to a value below its V_{DD}. Then pull-up resistors can be used to pull-up the output voltage to a suitable range on the receiving end. This can be seen in Figure 5.

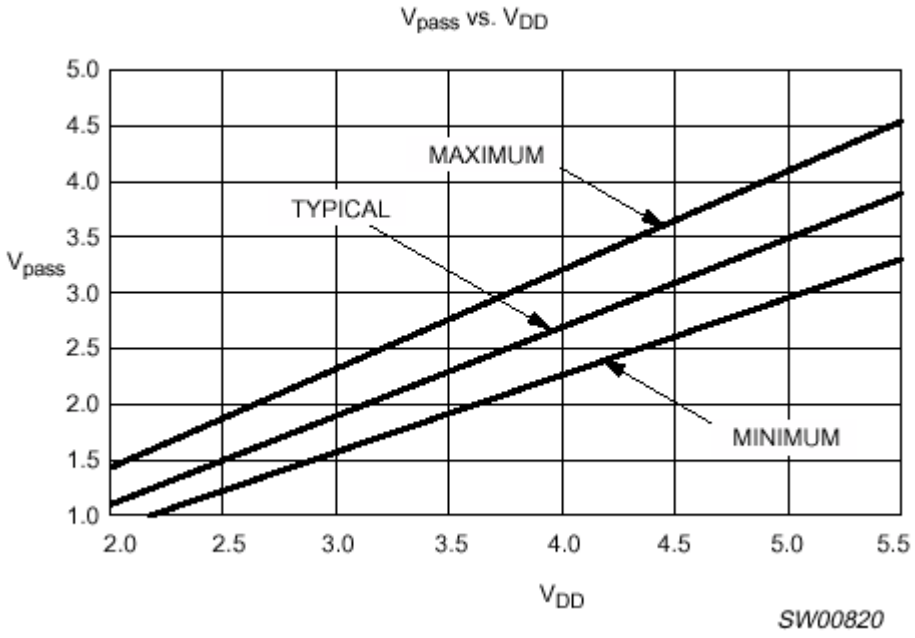


Figure 5. Voltage Translation

The graph shows that the I^2C voltage can be translated between the various channels of the multiplexer/switches. For example, let's say that the upstream channel uses 5V while the downstream channel uses 3.3V. If the PCA954X is supplied with 3.3V, it will clamp the voltage to about 2.3V so the 5V will not appear on the 3.3V side. A pull-up resistor on the 3.3V side then pulls it all the way up to the 3.3V rail. In most situations, a design engineer should use the maximum voltage curve since this is the situation you would find over the entire temperature range. The important thing to note is that the multiplexer/switches should normally be supplied with the lowest I^2C voltage to ensure proper voltage translation.

Choosing Pull-Up Resistors

The PCA954x multiplexers provide excellent isolation between the channels but do not provide any additional drive capability between the upstream and downstream buses. Therefore, both the upstream and downstream loads (bus capacitance and device input loads) must be taken into consideration when choosing the value of a pull-up resistor. Note that the PCA954X should normally be connected to the lowest voltage bus to ensure proper voltage clamping.

- Example 1: Typical application using a multiplexer. Pull-up resistors calculations
  Here is an application example to illustrate this point:

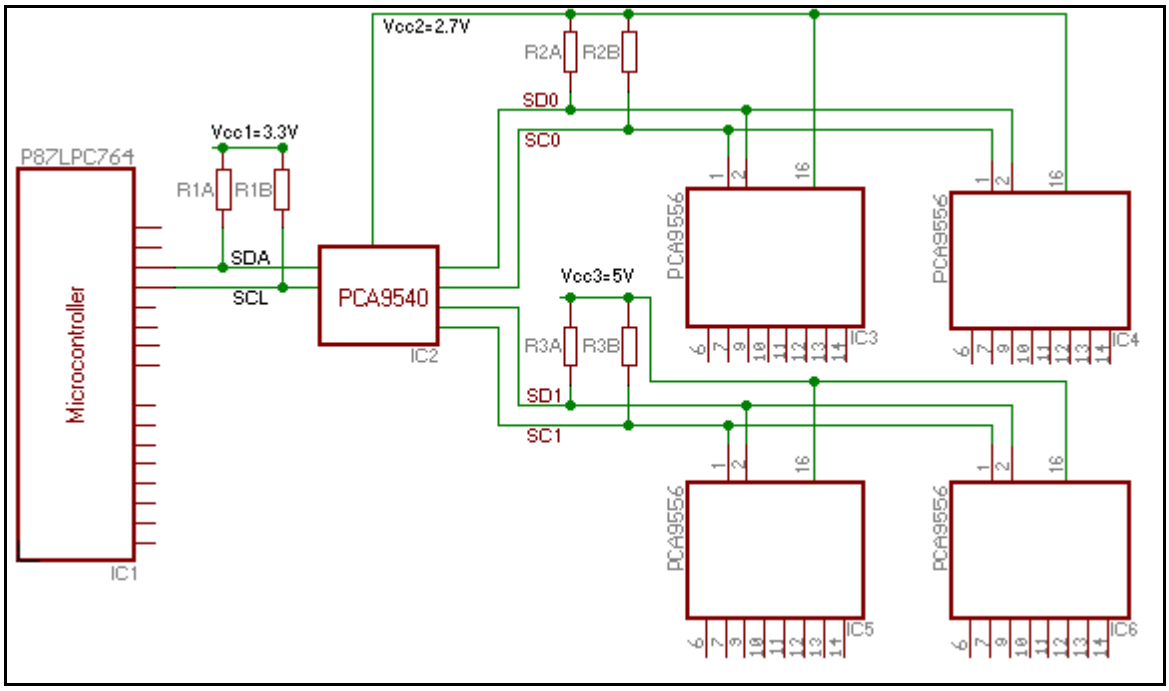


Figure 6. Typical Circuit Example

Assumptions are explained in Table 8.

	Supply Voltage	Load capacitance
Upstream channel	3.3 V	$C_{upstream}$ = 100 pF
Channel 0	2.5 V	$C_{channel0}$ = 300 pF
Channel 1	5.0 V	$C_{channel1}$ = 200 pF

Table 8. Electrical parameters

Note: Load capacitance includes device input capacitance and board capacitance.

The main considerations in choosing the pull-up resistor are:

1. Ensuring that the current does not exceed the maximum I_{ol}=3mA at 0.4V (minimum resistor value)
2. Ensuring that the rise time does not exceed 1.0μs for a standard mode (100 kHz) bus or 300 ns for the high speed mode (400 kHz) (affected by the bus capacitance and pull-up resistor).

   When the input voltage to the multiplexer is low, the resistance of the switch is assumed to be negligible in comparison to the pull-up resistors.

For this example of devices operating in the standard mode (100 kHz), the current consumption is not very important, so the maximum 3mA current is allowed to flow when SDA and SCL are low.

I_{UPA} is the current through R_{UPA}.
I_{0A} is the current through R_{0A}.
I_{1A} is the current through R_{1A}, etc.
If the voltage across the open-drain FETs is assumed to be zero, then the following equation is used to calculate the pull-up resistors:

Since the capacitance of the upstream branch is ¼ of the total capacitance of the SCL + SC0 branch, set I_{UP} = 3 mA/4 = 0.75mA. Therefore, the pull-up resistor in the upstream channel can have a current of I_{UP} = 0.75 mA and pull-up in the downstream channels can be set to $I_0 = I_1$ = 2.25 mA.

$R_{UPA} = R_{UPB}$ = 3.3 V / 0.75 mA = 4400 Ω

$R_{0A} = R_{0B}$ = 2.5 V / 2.25 mA = 1110 Ω
$R_{1A} = R_{1B}$ = 5.0 V / 2.25 mA = 2200 Ω

Additional verification:
make sure the rise time specification of 1µs for standard mode I²C is not exceeded.
Consider the V_{DD}-related input threshold of $V_{IH} = 0.7*V_{DD}$ and $V_{IL} = 0.3*V_{DD}$ for the purposes of RC time constant calculation.

$V(t) = V_{DD}(1–1/e^{-t/RC})$ where t is the time since the charging started and RC is the time constant.
$V(t1) = 0.3xV_{DD} = V_{DD}(1–1/e^{-t\ 1/RC})$; then t1 = 0.3566749xRC
$V(t2) = 0.7xV_{DD} = V_{DD}(1–1/e^{-t\ 2/RC})$; then t2 = 1.2039729xRC
T_{rise} = t2–t1 = 0.8472979xRC

Scenario 1: No downstream channel enabled
$T_{rise} = 0.8472979x\ R_{UPA}C_{upstream}$
= 0.8472979x4400x100e-12
= 0.37µs

Scenario 2: Channel 0 enabled
$T_{rise} = 0.8472979x(R_{UPA}// R_{0A})(C_{upstream} // C_{channel0})$
$= 0.8472979x \frac{4400x1110}{4400+1110} x (100\ e\text{-}12 + 300\ e\text{-}12)$
= 0.30µs
Scenario 3: Channel 1 enabled
$T_{rise} = 0.8472979x(R_{UPA}// R_{1A})(C_{upstream} // C_{channel1})$
$= 0.8472979x \frac{4400x2200}{4400+2200} x (100\ e\text{-}12 + 200\ e\text{-}12)$
= 0.37µs

All rise times are well below the maximum rise time of 1µs.

- Example 2: Equivalent resistance and capacitance values for multiple channel devices

The example in Figure 7 is made with the PCA9543, 2-channel switch. It can be easily extended to the 4 and 8-channel devices.

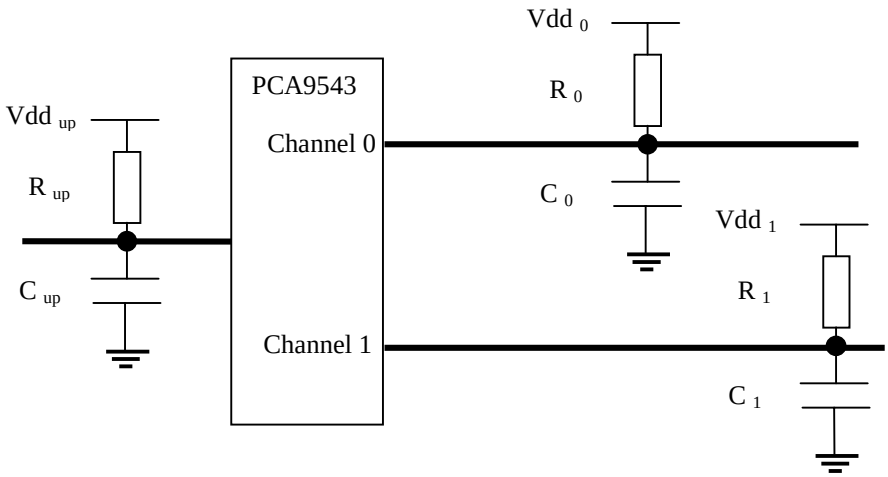


Figure 7. Multiple Channel Example

- R_{up}, R_0 and R_1: pull-up resistors on respectively the upstream channel, downstream channel 0 and downstream channel 1.
- C_{up}, C_0 and C_1: equivalent capacitance on respectively the upstream channel, downstream channel 0 and

downstream channel 1.
- The upstream channel is the main I^2C bus and can be connected to no downstream channels, to channel 0, channel 1 or both channel 1 and 2 at the same time.

- No channel selected:
  - Equivalent resistance of the I^2C bus = R_{up}
  - Equivalent capacitance of the I^2C bus = C_{up}

- Upstream channel connected to downstream channel 0:
  - Equivalent resistance of the I^2C bus = $R_{up} // R_0 = (R_{up} \times R_0)/ (R_{up} + R_0)$
  - Equivalent capacitance of the I^2C bus = $C_{up} // C_0 = C_{up} + C_0$

- Upstream channel connected to downstream channel 1:
  - Equivalent resistance of the I^2C bus = $R_{up} // R_1 = (R_{up} \times R_1)/ (R_{up} + R_1)$
  - Equivalent capacitance of the I^2C bus = $C_{up} // C_1 = C_{up} + C_1$

- Upstream channel connected to downstream channel 0 and downstream channel 1:
  - Equivalent resistance of the I^2C bus = $R_{up} // R_1 // R_2 = 1/(1/R_{up} + 1/R_1 + 1/R_2)$
  - Equivalent capacitance of the I^2C bus = $C_{up} // C_0 // C_1 = C_{up} + C_0 + C_1$

- For applications using more than 2 channels:
  - Equivalent resistance of the I^2C bus = Upstream pull-up resistors and all the connected downstream channel pull-up resistors in parallel
  - Equivalent capacitance of the I^2C bus = Upstream equivalent capacitance and all the connected downstream channel equivalent capacitance in parallel

APPLICATIONS

I^2C Multiplexing

Some specialized devices only have one I^2C address and sometimes several identical devices are needed in the same system. The multiplexers and switches split the I^2C bus into several sub-branches and allow the I^2C master to select and address one of multiple sets of identical devices, to avoid address conflict issues.

1. Eliminating address conflicts
   In many PCs, the EEPROM on DIMMs respond to the same I²C address (0xA0). Therefore, if more than one DIMM is installed in the card, there must be a method to read the data from each EEPROM. This can effectively be done using the PCA9540 as shown in Figure 8.

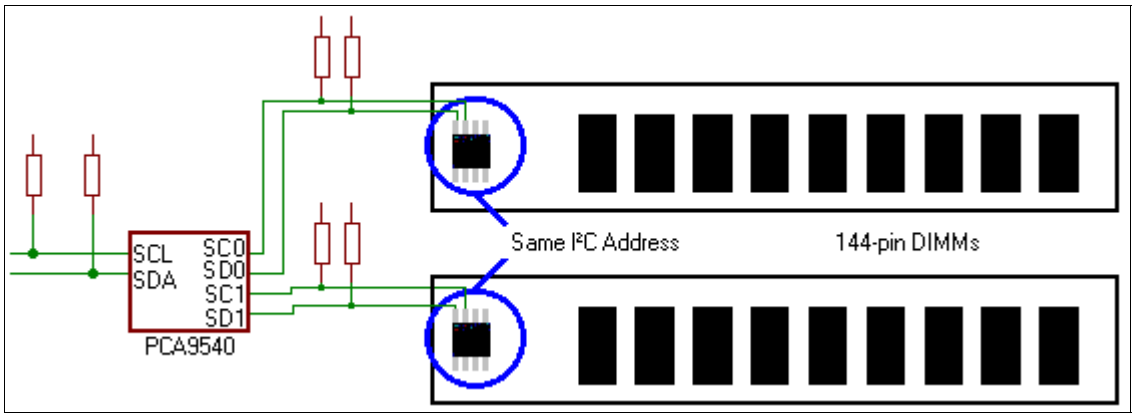


Figure 8. Example of DIMM Serial Presence Detect Application

Generally speaking, any application requiring more than one device with the same I²C address can use one or several PCA954X multiplexers to solve any addressing conflict.

Voltage Level Shifting

Many I^2C and SMBus devices operate at different voltage levels but need to operate on a common bus. The multiplexers and switches allow voltage translation between 1.65 V and 5.5 V. For example, a 5 V I^2C master on the upstream channel can communicate with a 3.3 V (non 5 V tolerant) SMBus device on channel 0 and a 2.5 V (non 3.3 V tolerant) I^2C device on channel 1. 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 that 1.8 V or 2.5 V or 3.3 V devices can communicate with 5 V devices without any additional protection. All I/O pins are tolerant to 6.0 V. The PCA954X switches are best for this voltage level shifting application since multiple downstream channels can be active at the same time.

Capacitive Load Sharing

Adding more I^2C and SMBus devices on the bus may exceed the 400 pF limitation. A multiplexer and a switch can isolate devices that are not currently needed to reduce the overall system loading and maintain a total load below 400 pF. When active, the channels act as a wire and the cumulative capacitive loading of the upstream channel and all active downstream channels must be considered. If total active channel system loading must be above 400 pF, then the Philips PCA9515/PCA9516/PCA9518 bus buffers should be considered.

Typical Application

Figure 9 shows the PCA9545 in a typical application.

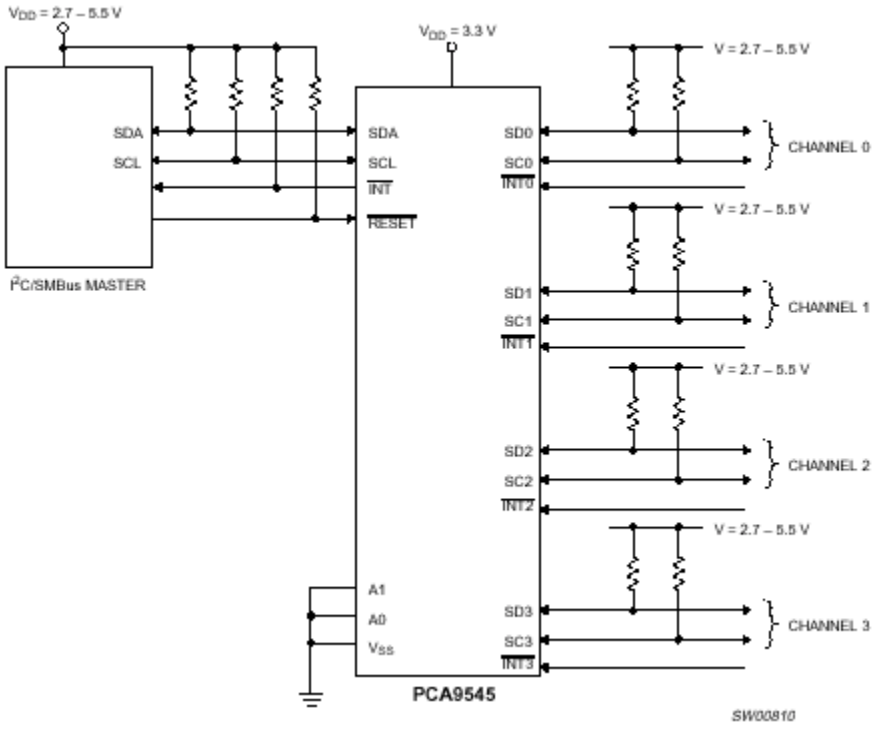


Figure 9. PCA9545 In a Typical Application

FREQUENTLY ASKED QUESTIONS

1. **Question:** What is the impedance of inputs/outputs when the part is powered down?
   Answer: The I^2C channel inputs/outputs will have a High impedance value if the voltage applied to the pin is below 5.5 V and above –0.5 V.

2. **Question:** What are all the uses for the interrupt line on the Philips PCA954X? I thought one use was for a time-out interrupt as it is used on the 8051 based I^2C controller. I now see that the PCA954X does not even have an internal timer for that. Another use I thought they could be used for transition detection for the Sxx line.
   Answer: The way the Interrupt inputs are implemented in these devices is more of a AND function, if one of the interrupts inputs goes low, then the interrupt output goes low. The interrupt is not latched. When the interrupt input signal goes away, then the interrupt output returns high. The main function of the interrupt is to let the upstream master know that it needs to service one of the downstream buses. By reading the control register, it is easy for the master to determine which of the downstream buses requires servicing.

3. **Question:** Exactly, what events or conditions generate an interrupt? And when exactly does an interrupt occur?
   Answer: The user defines the interrupt conditions. The device itself does not generate interrupts; it just collects them from any device and signals a master that an interrupt has occurred. The interrupt can happen at any time. The interrupt output signal is completely asynchronous.

4. **Question:** Is it allowed to connect a different voltage bus to the I/O of the PCA9544? The supply voltage of the PCA9544 is 3.3 V in our application. The sequence of the 3.3 V and 2.5 V I/O are not completely simultaneous but almost simultaneous.
   Answer: This is actually a feature of the part. You can connect different bus voltages to the various busses. This is shown in Figure 5 of the PCA9544 data sheet. The PCA9544 powers up with the channels open (high impedance) so if the 2.5 V powers up after the 3.3V there are no problem. Also, if the 3.3V powers up after the 2.5V, there will be no problem because the pins of the PCA9544 will be high impedance when the part is unpowered.

5. **Question:** What is your recommended value for the pull up resistor? According to the typical application shown in the datasheet, the values for the pull-up resistors vary from 2.7 V to 5.5 V for Channel 0 to Channel 3 when V_{DD} is 3.3 V. We want to make sure that there is no problem using 2.5 V pull up resistor because the V_{IH} is specified within 2.31 V ($0.7V_{DD}$) to 6 V.
   Answer: The PCA9544 creates a physical connection between the SDA and the selected SDn pin, and between the SCL and the SCn pin. This means that the resistors on both sides of a selected channel are in parallel and their summed current is limited by the I^2C family specification to 3 mA. At 3.3 V, the combined parallel resistance of the selected channel is, 3.3 V / 3 mA = 1.1 kΩ minimum. Therefore, the two pull-up resistors should be at least 2.2 kΩ each. Considering power supply variation and resistor tolerance, a safer choice may be 2.8 kΩ each.

6. **Question:** Can the PCA954X work in a multi-master environment, more specifically when masters are located upstream and downstream on the bus?
   Answer: The PCA954X can work in a multi-master environment. Arbitration between the two masters is possible through the multiplexer, assuming the channel is enabled. The PCA954x products however do not have the capability to detect activity on a downstream bus before the upstream master enables it. Therefore, care must be taken when switching buses if there is a master device on the downstream bus. The downstream bus must be idle when the upstream master enables that channel in order to avoid data corruption. When the downstream channel is not enabled, the master on that channel is unable to communicate with any of the other channels.

7. **Question:** I programmed the Philips PCA954X multiplexer to select Channel 1 in order to address a slave downstream device on this channel. I need to access this downstream device several times. Do I have to access the muliplexer first each time I have to access the downstream device?
   Answer: No, only one access sequence to the PCA954X is necessary to establish the physical connection between the upstream channel and the selected downstream channel. Once this access sequence is done and **after a Stop condition**, the PCA954X will go to the selected channel and then behave like a wire. It will keep the selected channel active until the next time the master controller addresses the PCA954X device (by its unique I^2C address) in order to change to another channel. Note that if a power-down occurs, the PCA954x's power-on reset circuit will initialize the device back to a state where no downstream channel is selected. The explanation above is also applicable to PCA54X Switches. The only difference when using a Switch is that several channels can be selected

and active at a time and one access sequence can select all the channels at the same time.

8. **Question:** How can I realize a twelve channel multiplexer with the Philips PCA954X devices?
   Answer: Simply connect three 4-channel multiplexers/switches and program them to use 3 different addresses so that the bus master can individually address each device and individually control each channel on each device. The upstream I^2C channels of each device are connected to the upstream bus master channel. The PCA9548 8 channel switch and any of the 4 channel multiplexers/switches could also be used in any combination. Using the PCA9548, the maximum number of downstream channels possible is 64 (8 channels x 8 devices) without using some sort of address change scheme. Since the PCA9548 is a Switch, all 64 downstream channels could be selected and active at the same time but in practice, this is not practical since the capactive loading would exceed the specification. The Philips PCA9515/PCA9516/PCA9518 bus buffers should be considered for large capacitance load situations.

9. **Question:** Does PCA9544 support 2.5 V I^2C signal at V_{DD} = 3.3V?
   Answer: The graph shows that at 3.3 V V_{DD}, the maximum voltage is 2.6 V and minimum is 1.8 V. At 3.0 V the maximum is 2.25 V and minimum is 1.7 V. At 3.6 V the maximum is 2.8 V and minimum is 2.0 V. So yes, the PCA9544 will support downstream channels at 2.5 V, 3.3 V and 5 V if powered at 3.3 V $V_{DD.}$

10. **Question:** What happens to a PCA954X devices if the I^2C bus is powered up (bus at V_{DD}) when the PCA device is still off (not yet powered up or slowly ramping up)? Is there a risk of latch-up, partial or permanent damage of the part?
    Answer: There should be no latch-up or damage, permanent or otherwise, to the PCA954X device under the conditions. In addition, the device will not cause any disruption to the I^2C bus under these conditions.

11. **Question:** Can we use both 3.3 V and 5 V I^2C busses on different sides of the multiplexers?
    Answer: Yes, if the multiplexer is powered at 5 V you can use 3.3 V and 5 V or if powered at 3.3 V you can use 2.5 V, 3.3 V or 5 V pull-up resistors on any of the channels. For further information, please refer to Figure 5 “Voltage Translation” showing voltage drop range based on $V_{DD.}$

12. **Question:** Are there any voltage shifts on the I^2C busses when using the PCA954X multiplexers/switches?
    Answer: The only voltage shift would be caused by the voltage used for the pull-up resistor.

13. **Question:** Can we use multiple PCA954X I^2C multiplexers/switches in series? If so, how many? What are the limitations?
    Answer: The PCA954X devices can be used in series or parallel. The limitation would be I^2C addresses (up to 8 of the devices on the active bus depending on the type of device used in the application) and capacitive limit of 400 pF since the PCA954X devices don't have the ability to isolated the capacitive load.

14. **Question:** How do we gracefully recover from a stuck bus scenario?
    Answer: The only way to fix a stuck situation is to send extra clock pulses if the data line is stuck low. Sending 8 extra clock pulses will re-initialize the slave state machine. If the clock line is stuck, then there is not much that can be done without extra hardware . One hardware solution would be to use a pair of 74LVC1G66 analog switch picogates to isolate the I^2C lines. If the system includes the PCA954X switch, it is possible to do a hardware reset and isolate all of the bus connections on the downstream side of the device. The PCA9516 Hub can also be used since each channel has a separate hardware enable .

15. **Question:** Is there some “intelligence” in the PCA954X, I mean are they able to “finish” some I^2C sequences if a reset occurs in the multiplexer/switch while a communication is on going or initiate some downstream re-initialization in order not to stay stuck in the middle of a communication? If the multiplexer/switch is reset during a communication, the downstream device(s) will not see the end of the sequence (Stop condition) and when a new channel will be open, some failures or miscommunication could occur in the I^2C bus.
    Answer: The multiplexer/switches are "simple devices" meaning that they are just used to send information bi-directionally between an upstream bus and one/several downstream. The devices can't take any initiative and send any kind of I^2C sequence on their own. They can't initialize or reset some devices by their own: as soon as the reset pin is asserted, the switch goes directly to the reset mode, initializes its state machine and disables all the downstream channels.

16. **Question:** Is there any interrupt generated to the master upon a stuck bus condition?
    Answer: No.

17. **Question:** Is there any affect on the slave (downstream device) when the PCA954X switch Reset pin is asserted?
    Answer: There is no affect on the downstream device. The Reset only disables all the downstream channels.

ADDITIONAL INFORMATION

The latest datasheets for the PCA954X family of products and other SMBus/I^2C products can be found at the Philips Semiconductors website:
http://www.philipslogic.com/i2c

Software tools for most of Philips' products can be found at:
http://www.demoboard.com

Additionnal technical support for PCA954X devices can be provided by e-mailing the question to:
Email: pc.mb.svl@philips.com