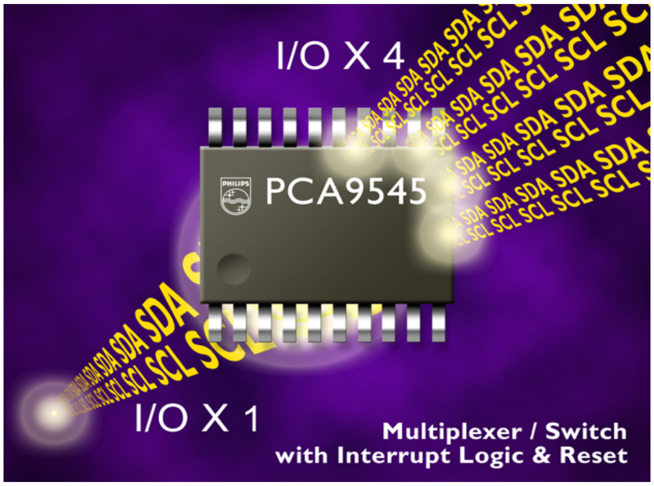
APPLICATION NOTE



AN262_2

PCA954X FAMILY OF I²C / SMBus MULTIPLEXERS and SWITCHES

PCA9540B, PCA9541, PCA9542A, PCA9543A, PCA9544A, PCA9545A, PCA9546A, PCA9547, PCA9548A, PCA9549

Abstract - Philips Semiconductors family of Multiplexers and Switches are detailed in this application note that discusses device operation, software programming, pull up resistor sizing/bus termination and typical applications.

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

Specialty Logic Product Line
Logic Product Group

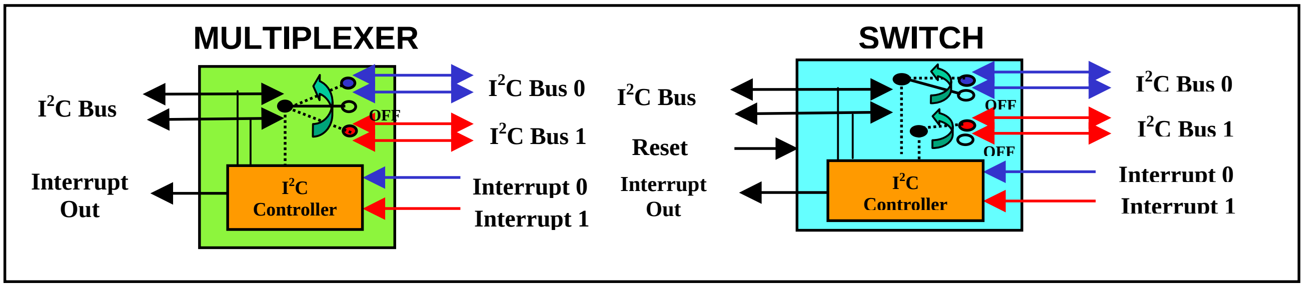
Philips Semiconductors October 27, 2004

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 OR 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 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 restore communication with the downstream channels without having to cycle power to the equipment or to other I^{2}C bus devices. The PCA9541 option /01 and /02 and the PCA9547 have one channel active at start up for applications that want a connection without having to address the multiplexer.

Hardware Pins - Up to three hardware pins (A0, A1, A2), except the PCA9541 that has an additional pin (A3), are provided to change the I^{2}C address and allow up to eight PCA954X devices (up to sixteen for the PCA9541) to share the same I^{2}C/SMBus.

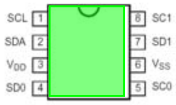
DEVICE NAME	MULTIPLEXER (In/Out)	SWITCH (In/Out)	FEATURES							
						PACKAGES				
			# of ADDRESSES	INTERRUPT (In/Out)	HARDWARE RESET	PIN COUNT	SO (narrow)	SO (wide)	TSSOP	HVQFN
PCA9540B	1-2		1			8	D		DP	
PCA9541	2-1		16	1-2	●	16	D		PW	BS
PCA9542A	1-2		8	2-1		14	D		PW	
PCA9543A		1-2	4	2-1	●	14	D		PW	
PCA9544A	1-4		8	4-1		20		D	PW	BS
PCA9545A		1-4	4	4-1	●	20		D	PW	BS
PCA9546A		1-4	8		●	16	D		PW	BS
PCA9547	1-8		8		●	24		D	PW	BS
PCA9548A		1-8	8		●	24		D	PW	BS
PCA9549		8-bit	8		●	24		D	PW	BS

Table 1. PCA954X Features

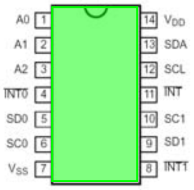
Operating Characteristics

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

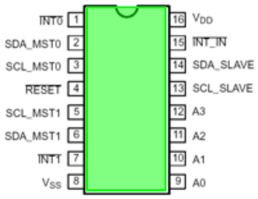
Device Pinout

<table>
<tr><th></th><th>2-Channel</th><th>2-Channel
w/Interrupt</th><th>4-Channel
w/Interrupt</th><th>4-Channel</th><th>8-Channel</th></tr>
<tr>
<td>M
U
L
T
I
P
L
E
X
E
R</td>
<td>PCA9540B


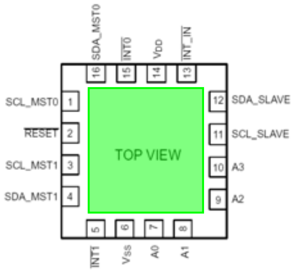
SCL 1
SDA 2
VDD 3
SD0 4
8 SC1
7 SD1
6 VSS
5 SC0

</td>
<td>PCA9542A


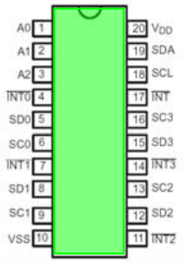
A0 1
A1 2
A2 3
INT0 4
SD0 5
SC0 6
VSS 7
14 VDD
13 SDA
12 SCL
11 INT
10 SC1
9 SD1
8 INT1

PCA9541


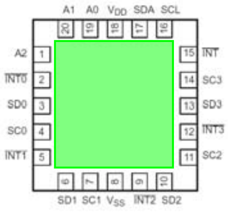
INT0 1
SDA_MST0 2
SCL_MST0 3
RESET 4
SCL_MST1 5
SDA_MST1 6
INT1 7
VSS 8
16 VDD
15 INT_IN
14 SDA_SLAVE
13 SCL_SLAVE
12 A3
11 A2
10 A1
9 A0



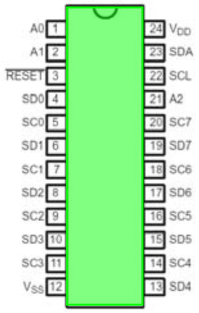
SDA_MST0 16
INT0 15
VDD 14
INT_IN 13
SCL_MST0 1
RESET 2
SCL_MST1 3
SDA_MST1 4
12 SDA_SLAVE
11 SCL_SLAVE
10 A3
9 A2
TOP VIEW
5 INT1
6 VSS
7 A0
8 A1

With Reset</td>
<td>PCA9544A


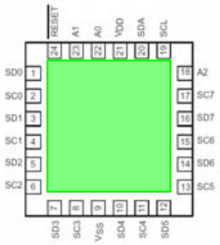
A0 1
A1 2
A2 3
INT0 4
SD0 5
SC0 6
INT1 7
SD1 8
SC1 9
VSS 10
20 VDD
19 SDA
18 SCL
17 INT
16 SC3
15 SD3
14 INT3
13 SC2
12 SD2
11 INT2



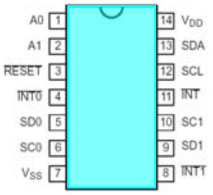
A1 A0 VDD SDA SCL
20 19 18 17 16
A2 1
INT0 2
SD0 3
SC0 4
INT1 5
15 INT
14 SC3
13 SD3
12 INT3
11 SC2
6 7 8 9 10
SD1 SC1 VSS INT2 SD2

</td>
<td></td>
<td>PCA9547


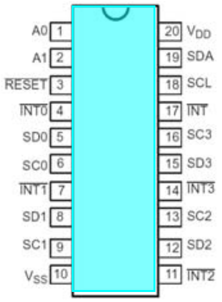
A0 1
A1 2
RESET 3
SD0 4
SC0 5
SD1 6
SC1 7
SD2 8
SC2 9
SD3 10
SC3 11
VSS 12
24 VDD
23 SDA
22 SCL
21 A2
20 SC7
19 SD7
18 SC6
17 SD6
16 SC5
15 SD5
14 SC4
13 SD4



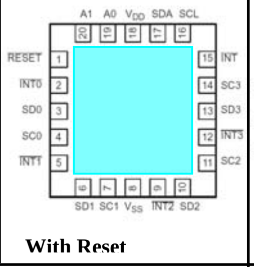
RESET A1 A0 VDD SDA SCL
24 23 22 21 20 19
SD0 1
SC0 2
SD1 3
SC1 4
SD2 5
SC2 6
18 A2
17 SC7
16 SD7
15 SC6
14 SD6
13 SC5
7 8 9 10 11 12
SD3 SC3 VSS SD4 SC4 SD5

With Reset</td>
</tr>
<tr>
<td>S
W
I
T
C
H</td>
<td></td>
<td>PCA9543A


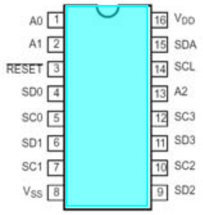
A0 1
A1 2
RESET 3
INT0 4
SD0 5
SC0 6
VSS 7
14 VDD
13 SDA
12 SCL
11 INT
10 SC1
9 SD1
8 INT1

With Reset</td>
<td>PCA9545A


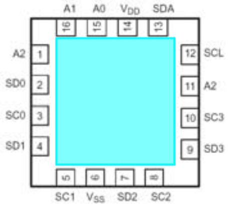
A0 1
A1 2
RESET 3
INT0 4
SD0 5
SC0 6
INT1 7
SD1 8
SC1 9
VSS 10
20 VDD
19 SDA
18 SCL
17 INT
16 SC3
15 SD3
14 INT3
13 SC2
12 SD2
11 INT2



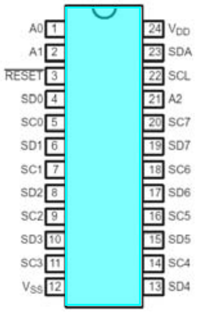
A1 A0 VDD SDA SCL
20 19 18 17 16
RESET 1
INT0 2
SD0 3
SC0 4
INT1 5
15 INT
14 SC3
13 SD3
12 INT3
11 SC2
6 7 8 9 10
SD1 SC1 VSS INT2 SD2

With Reset</td>
<td>PCA9546A


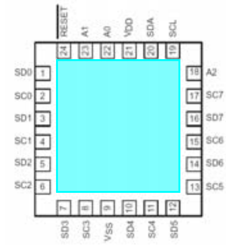
A0 1
A1 2
RESET 3
SD0 4
SC0 5
SD1 6
SC1 7
VSS 8
16 VDD
15 SDA
14 SCL
13 A2
12 SC3
11 SD3
10 SC2
9 SD2



A1 A0 VDD SDA
16 15 14 13
A2 1
SD0 2
SC0 3
SD1 4
12 SCL
11 A2
10 SC3
9 SD3
5 6 7 8
SC1 VSS SD2 SC2

With Reset</td>
<td>PCA9548A / PCA9549


A0 1
A1 2
RESET 3
SD0 4
SC0 5
SD1 6
SC1 7
SD2 8
SC2 9
SD3 10
SC3 11
VSS 12
24 VDD
23 SDA
22 SCL
21 A2
20 SC7
19 SD7
18 SC6
17 SD6
16 SC5
15 SD5
14 SC4
13 SD4



RESET A1 A0 VDD SDA SCL
24 23 22 21 20 19
SD0 1
SC0 2
SD1 3
SC1 4
SD2 5
SC2 6
18 A2
17 SC7
16 SD7
15 SC6
14 SD6
13 SC5
7 8 9 10 11 12
SD3 SC3 VSS SD4 SC4 SD5

With Reset</td>
</tr>
</table>

Table 2. PCA954X Pin Out

Ordering Information

Package	Container	PCA9540B	PCA9541/0X (1)	PCA9542A	PCA9543A	PCA9544A
SO	Tube	PCA9540BD	PCA9541D/0X	PCA9542AD	PCA9543AD	PCA9544AD
	T & R	PCA9540BD-T	PCA9541D/0X-T	PCA9542AD-T	PCA9543AD-T	PCA9544AD-T
TSSOP	Tube	Not available	PCA9541PW/0X	PCA9542APW	PCA9543APW	PCA9544APW
	T & R	PCA9540BDP-T	PCA9541PW/0X-T	PCA9542APW-T	PCA9543APW-T	PCA9544APW-T
HVQFN	T & R	Not available	PCA9541BS/0X-T	Not available	Not available	PCA9544ABS-T

Package	Container	PCA9545A	PCA9546A	PCA9547	PCA9548A	PCA9549
SO	Tube	PCA9545AD	PCA9546AD	PCA9547D	PCA9548AD	PCA9549D
	T & R	PCA9545AD-T	PCA9546AD-T	PCA9547D-T	PCA9548AD-T	PCA9549D-T
TSSOP	Tube	PCA9545APW	PCA9546APW	PCA9547PW	PCA9548APW	PCA9549PW
	T & R	PCA9545APW-T	PCA9546APW-T	PCA9547PW-T	PCA9548APW-T	PCA9549PW-T
HVQFN	T & R	PCA9545ABS-T	PCA9546ABS-T	PCA9547BS-T	PCA9548ABS-T	PCA9549BS-T

Note (1): X = 1, 2 or 3 and applies to the 3 different versions that are available

Table 3. PCA954X Ordering Information

The PCA9540/42/43/44/45/46/48 were made in a BiCMOS process in the Philips Semiconductors Albuquerque fabrication plant (ABQ), which was shut down in 2003, and the devices have been discontinued (DN-53). The replacement devices, PCA9540B/42A/43A/44A/45A/46A and 48A, are made in a CMOS process from Philips Semiconductors Boblingen Germany fabrication plant (PSB) and are drop in replacements. Technical comparison data can be requested from I2C.Support@philips.com to assist with the conversion.

Data Sheets and IBIS Models

Data sheets and IBIS models can be downloaded from www.philips.com/i2clogic

TECHNICAL INFORMATION

Block Diagram PCA954x devices (Except PCA9541 and PCA9549)

The PCA954X devices are bi-directional translating Multiplexers and Switches, controlled via the I^{2}C-bus. The PCA9543A block diagram is shown in Figure 1 as an example for the entire family. The SCL/SDA upstream pair fans out to downstream pairs, or channels that are controlled by FET switches. The number of downstream pairs is device dependent. Exactly the same I^{2}C signals on the upstream channel are passed onto all the downstream channels without amplification and the 400 pF I^{2}C 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^{2}C 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. From the default condition of no channels active (except for the PCA9547 which defaults with Channel 0 active), only a master located on the upstream channel can activate downstream channels. A master on an unconnected downstream channel (not active) cannot activate its own or any other channel since the commands cannot reach the I^{2}C-bus Control block. Once that downstream channel is activated however, any master on that channel can communicate with the I^{2}C-bus Control block and can control other downstream channels.

The Multiplexers and Switches operate basically the same with the primary difference being within the I^{2}C-bus Control block. The Multiplexer I^{2}C-bus Controller activates only one channel at a time while the Switch I^{2}C-bus Controller activates as many channels as there are available, in any combination, as determined by the contents of the programmable control register.

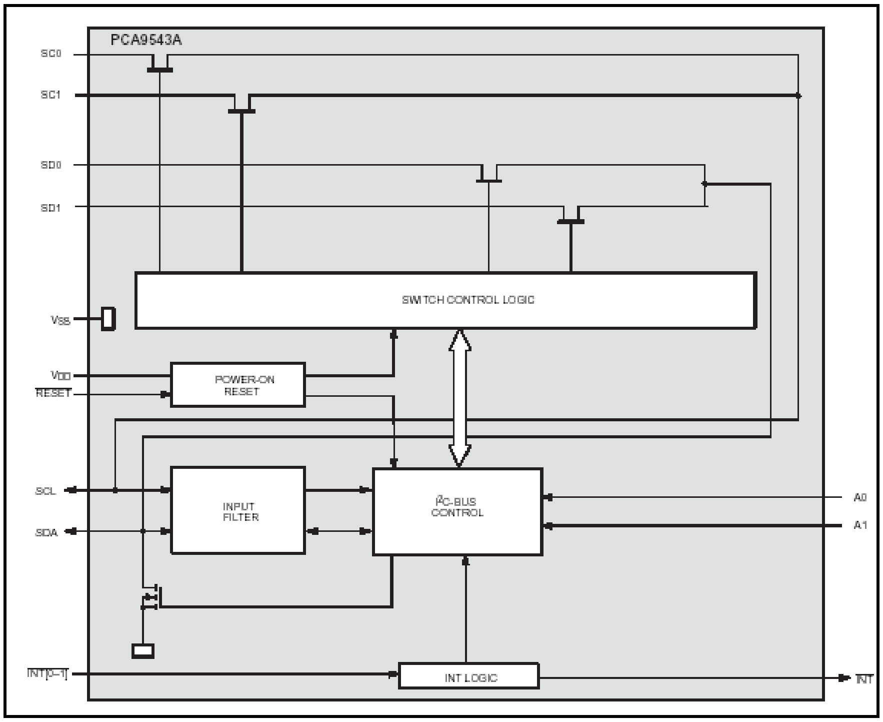


Figure 1. PCA9543A Block Diagram

Both PCA954X Multiplexers and Switches offer a built-in Power-On Reset (POR) 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.

Only certain devices offer an external hardware pin reset capability (/RESET). Holding the /RESET pin low 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 without having to cycle power to the equipment to perform a POR to all I^2C devices. A 10 kΩ (or sized to the master's output capacity) pull-up resistor is required to hold the /RESET pin high for normal device operation.

An Interrupt input pin is provided for each SCx/SDx downstream pair on select PCA954X Multiplexers and Switches. The active low open-drain interrupt output acts as an OR of the interrupt inputs and is not latched (When interrupt inputs are logic level high, the interrupt output is logic level high. When one or several interrupt occur and the interrupt inputs go to logic level low, the interrupt output goes logic level low). When the interrupt input condition disappears, the Interrupt output condition disappears as well.

Up to three external hardware address pins (A0, A1, A2) 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}. Package pin limitations limit the number of address pins on some devices and that address bit is fixed internally.

All PCA954X Multiplexers and Switches can operate with a V_{DD} between 2.3 V to 5.5 V and the SDx/SCx pins are tolerant to voltages up to 6.0 V. 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 devices can communicate with 5 V devices without any additional protection. External pull-up resistors pull the SCx/SDx pairs up to the desired voltage level.

The PCA954X devices can drive the SDA line and will do so to acknowledge commands sent to its specific I^2C address. The PCA954X devices are slaves that **CANNOT** drive the SCL line. SCL and SDA signals driven by other devices pass transparently through the PCA954X device FET switches to any active channel.

Block Diagram PCA9541

The PCA9541 is primarily a 2-to-1 I^{2}C master selector designed for high reliability dual master I^{2}C applications where system operation is required, even when one master fails or the controller card is removed for maintenance. The two masters (e.g., primary and back-up) are located on separate I^{2}C-buses (SDA/SCL_MASTER0 and SDA/SCL_MASTER1) that connect to the same downstream I^{2}C-bus (SDA/SCL_SLAVE) slave devices. I^{2}C commands are sent by either I^{2}C-bus master and are used to select one master at a time. Either master at any time can gain control of the slave devices if the other master is disabled or removed from the system. The failed master is isolated from the system and will not affect communication between the on-line master and the slave devices on the downstream I^{2}C-bus.

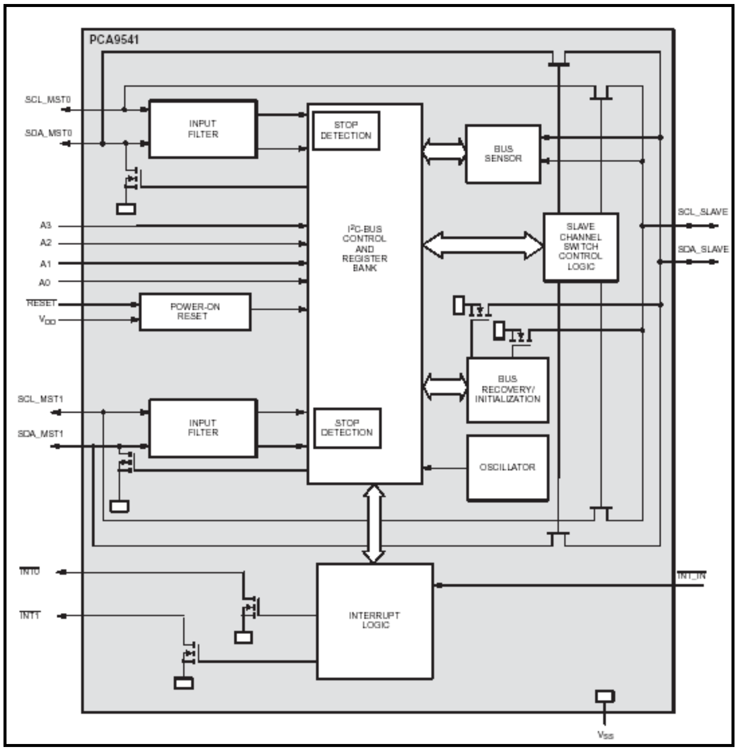


Figure 2. PCA9541 Block Diagram

Three versions are offered for different architectures:

- PCA9541/01 with channel 0 selected at start-up
- PCA9541/02 with channel 0 selected after start-up and after stop condition is detected
- PCA9541/03 with no channel selected after start-up.

The 3 versions offer a built-in Power-On Reset (POR) 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 devices also offer an external hardware pin reset capability (/RESET). Holding the /RESET pin low will deactivate all the downstream channels, as well as reset the I²C state machine. It allows the master to initialize the device in a known default state (different depending on the version that is used). A 10 kΩ (or sized to the master's output capacity) pull-up resistor is required to hold the /RESET pin high for normal device operation.

The interrupt outputs are used to provide an indication of which master has control of the bus. One interrupt input (/INT_IN) collects downstream information and propagates it to the 2 upstream I^{2}C-buses (/INT0 and /INT1) if enabled. /INT0 and /INT1 are also used to let the previous bus master know that it is not in control of the bus anymore and to indicate the completion of the bus recovery/initialization sequence. Those interrupts can be disabled and will not generate an interrupt if the masking option is set.

External pull-up resistors pull the bus to the desired voltage level for each channel. All I/O pins are 6.0 V tolerant.

The PCA9541 device targets primarily applications different from the multiplexers and switches that are discussed in this application note. However, it can be used as a 2-channel multiplexer that defaults to channel 0 under certain conditions. Please refer to the paragraph “How to use the PCA9541 as a 2-channel multiplexer” for more detail.

Block Diagram PCA9549

The PCA9549 is similar to the PCA9548A with each channel selection now digital CBT type switch instead of an I^{2}C-bus channel. The primary application is to allow digital signals to be turned on and off remotely via the I^{2}C-bus. One example is it can be used to control the resistance in a test set up with different resistors (e.g., 100 Ω, 1 kΩ, 10 kΩ, 100 kΩ, 1 MΩ, 10 MΩ, 100 MΩ) on each switch channel, allowing various resistance combinations controlled via the I^{2}C-bus.

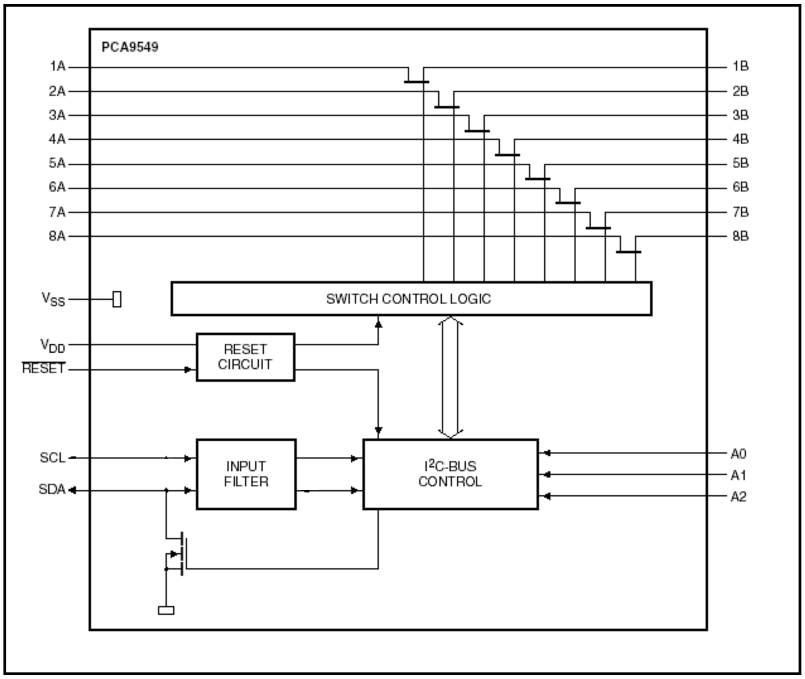


Figure 3. PCA9549 Block Diagram

I²C Communications

All PCA954X devices support both standard mode (100 kHz) and fast mode (400 kHz) I^{2}C 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^{2}C protocol throughput. A standard I^{2}C 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, Least Significant Bit) 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^{2}C communication).

Device Type	I^2C Address							R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PCA9540B	1	1	1	0	0	0	0	1/0
PCA9541	1	1	1	A3	A2	A1	A0	1/0
PCA9542A	1	1	1	0	A2	A1	A0	1/0
PCA9543A	1	1	1	0	0	A1	A0	1/0
PCA9544A	1	1	1	0	A2	A1	A0	1/0
PCA9545A	1	1	1	0	0	A1	A0	1/0
PCA9546A	1	1	1	0	A2	A1	A0	1/0
PCA9547	1	1	1	0	A2	A1	A0	1/0
PCA9548A	1	1	1	0	A2	A1	A0	1/0
PCA9549	1	1	1	0	A2	A1	A0	1/0

Table 4. PCA954X Addresses

Notes:

- A0, A1 and A2 in blue (dark gray) are the hardware programmable input pins that are connected to either V_{DD} (logic level 1) or GND (logic level 0) and modify the device's I^2C address.
- Up to four PCA954X devices can be attached to the same I^2C bus when there are 2 address pins available.
- Up to eight PCA954X devices can be attached to the same I^2C bus when there are 3 address pins available.
- Up to sixteen PCA9541 devices can be attached to the same I^2C bus when there are 4 address pins available.
- Since all PCA954X devices have the same fixed bits 7, 6, 5 and 4, the maximum allowed on the I^2C bus is 8 of any combination of PCA954X devices.
- PCA9541 has 4 programmable address pins thus allowing 16 devices to share 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 operation (Switches only) is initiated (/RESET asserted low), all channels are deactivated and no channels are active.

Table 5 describes the channel selection for the 2-channel PCA9540B/42A/43A, the 4-channel PCA9544A and the 8-channel PCA9547. "x" is don't care (0 or 1).

Control Register							Device Channel Selection			
6	**5**	**4**	**3**	**2**	**1**	**0**	**PCA9540B/42A**	**PCA9543A**	**PCA9544A**	**PCA9547**
x	x	x	0	0	0	0	None	None	None	None
x	x	x	0	0	0	1	None	Channel 0	None	None
x	x	x	0	0	1	0	None	Channel 1	None	None
x	x	x	0	0	1	1	None	Channel 0 & 1	None	None
x	x	x	0	1	0	0	Channel 0	None	Channel 0	None
x	x	x	0	1	0	1	Channel 1	Channel 0	Channel 1	None
x	x	x	0	1	1	0	None	Channel 1	Channel 2	None
x	x	x	0	1	1	1	None	Channel 0 & 1	Channel 3	None
x	x	x	1	0	0	0	None	None	None	Channel 0
x	x	x	1	0	0	1	None	Channel 0	None	Channel 1
x	x	x	1	0	1	0	None	Channel 1	None	Channel 2
x	x	x	1	0	1	1	None	Channel 0 & 1	None	Channel 3
x	x	x	1	1	0	0	Channel 0	None	Channel 0	Channel 4
x	x	x	1	1	0	1	Channel 1	Channel 0	Channel 1	Channel 5
x	x	x	1	1	1	0	None	Channel 1	Channel 2	Channel 6
x	x	x	1	1	1	1	None	Channel 0 & 1	Channel 3	Channel 7

Table 5. 2, 4 and 8 Channel Device Active Channel Selection

Table 6 describes the channel selection for the 4-channel PCA9545A/46A and 8-channel PCA9548A/PCA9549. The PCA9545A/46A/48A and PCA9549 will respond to the channels in gray but only the 8-channel PCA9548A/PCA9549 will respond to the channels in blue (dark gray). "◆" in the Active Channel columns indicates that the channel is active.

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 and 8 Channel Device Active Channel Selection

Note:
Several channels can be enabled at the same time. For example, in the PCA9548A/PCA9549, "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 OR of the interrupt inputs.

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 (not latched) and the device does not keep in memory what caused the interrupt (the control register interrupt bits return to '0', values are not latched). The PCA9542A/43A only have interrupt channels 0 and 1 and don't include any of the interrupts in channel 2 and 3 (blue (dark gray) highlighted columns). The PCA9544A/45A have interrupt channels 0 through 3. "◆" in the Interrupt channel columns indicates that there is an interrupt.

Control Register								Interrupt Input 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

Notes:

- 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.

Interrupt input signals:

Interrupt inputs don't have any internal pull-up resistors and **MUST NOT** be left floating in order to avoid any undesired interrupt conditions or additional current consumption due to undefined logic states. If one or several inputs are not used, they must be connected to V_{DD} through a pull-up resistor (4.7 kΩ for example).

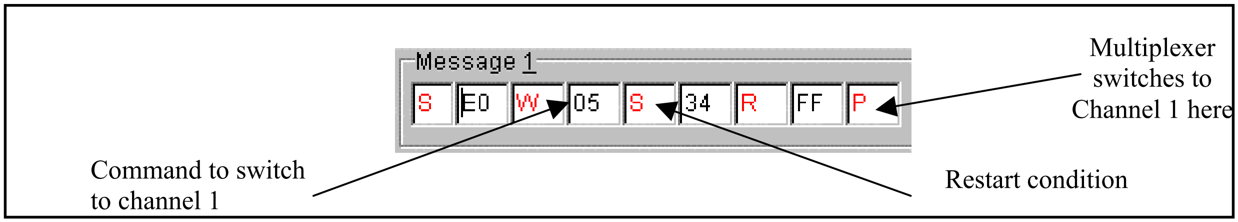
- If the device generating the interrupt has an open-drain structure or can be 3-stated, a pull-up resistor to V_{DD} must be used.
- If the device generating the interrupt has a totem pole output structure and cannot be 3-stated; pull-up resistors to V_{DD} are not required.

Interrupt inputs have been implemented to offer an Interrupt capability (e.g., a downstream device sends request to the master to request a maintenance or a supervision operation). The interrupt inputs can also be used as regular GPI (General Purpose Inputs) since the information can then be read through the Interrupt register.

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 4 sends a command to the multiplexer to switch to channel 1, followed by a command to read the Philips PCA9557 (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 PCA9557 from the original channel, not channel 1 because the multiplexer will switch to channel 1 **ONLY** after the STOP condition has been sent.



Message 1

S | E0 | W | 05 | S | 34 | R | FF | P

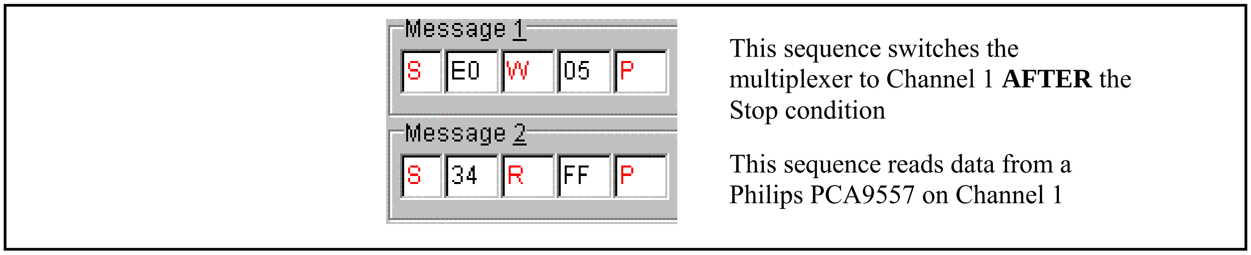
Multiplexer switches to Channel 1 here

Command to switch to channel 1

Restart condition

Figure 4. Incorrect Channel Selection

In order to get read valid data from the Philips PCA9557 located on channel 1, the following message should be sent:



Message 1

S | E0 | W | 05 | P

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

Message 2

S | 34 | R | FF | P

This sequence reads data from a Philips PCA9557 on Channel 1

Figure 5. 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 5. Now all I²C messages are being transmitted through the correct channel.

Note that once the master controller has configured the PCA954X, the device behaves like a wire (transparent switch behavior) 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 the PCA954X device and there is no need to access and configure the PCA954X device again.

Power up / Reset default state

All the devices power up with no downstream channel selected, except for the PCA9541/01, PCA9541/02 and PCA9547, which powers up with Channel 0 selected.

A reset performed on the PCA9543A, PCA9545A, PCA9546A, PCA9548A and PCA9549 cause all the downstream channels to be unselected (open).

A reset performed on the PCA9541/01, PCA9541/02 and PCA9547 causes Channel 0 to be selected.

Voltage Clamping

The PCA954X Multiplexers and Switches contain pass transistor, which are very fast and have very low on-resistance. When the pass transistor is enabled and the input voltage is low, the output is also low and the pass transistor has a typical on-resistance around 25 Ω. 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 6 below. The clamping voltage will be somewhat lowered by a load on the output (shown with open output).

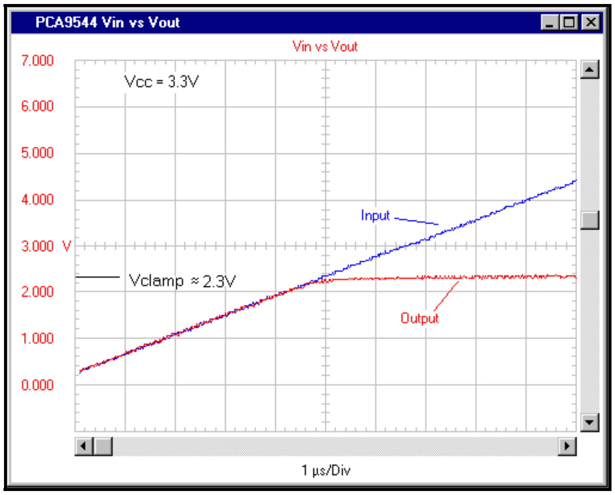


Figure 6. Vout Vs Vin

Voltage Translation

The voltage clamping can be advantageous when there is a 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 of each individual branch to a suitable range. This can be seen in Figure 7.

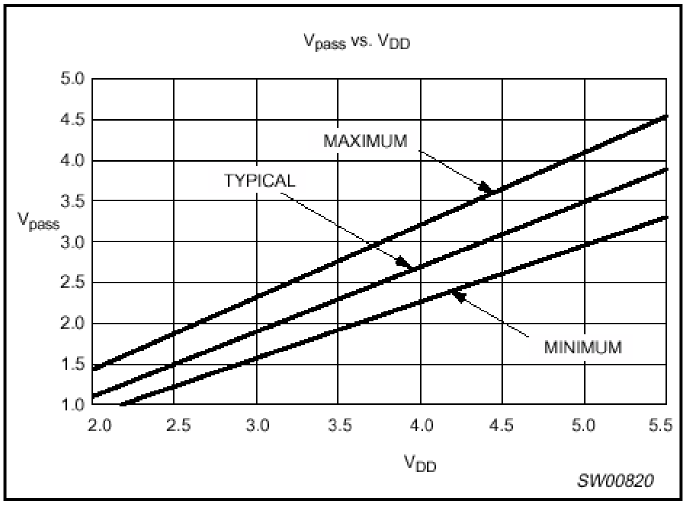


Figure 7. Voltage Translation

The graph shows that the I^{2}C voltage can be translated between the various channels of the PCA954X device. For example, if the upstream channel uses 5 V while the downstream channel uses 3.3 V. If the PCA954X is supplied with 3.3 V, it will clamp the voltage to about 2.3 V so the 5 V will not appear on the 3.3 V side. A pull-up resistor on the 3.3 V side then pulls it all the way up to the 3.3 V 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 multiplexers/switches should be supplied with the lowest I^{2}C voltage needed to ensure proper voltage translation.

How to use the PCA9541 as a 2-channel multiplexer

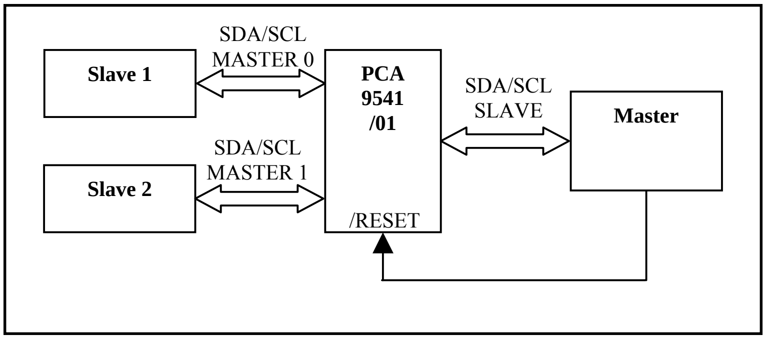


Figure 8. Two-channel multiplexer using a PCA9541

In this application, a PCA9541/01, which defaults with SDA/SCL_MASTER0 enabled (upstream channel 0 connected to downstream channel), is used in a configuration where its 2 upstream I^{2}C channels (SDA/SCL_MST0 and SDA/SCL_MST1) contain the 2 slave devices and it downstream channel (SDA/SCL_SLAVE) contains the master device.

- At power up, PCA9541/01 is configured with SDA/SCL_MASTER0 connected with the downstream slave channel. The master located in the slave side can then directly access Slave 1 device.
- When the master wants to communicate with Slave 2 device, it has to configure PCA9541/01 through its Control register to enable SDA/SCL_MASTER1.
- From there, if the Master wants to communicate again with Slave 1 device, it can access SDA/SCL_MASTER by 2 different methods:
  1. Generate a /RESET strobe causing the PCA9541/01 to go to its default state (SDA/SCL_MASTER0 enabled)
  2. Write to the Control Register in order to "give up" the control of SDA/SCL_MASTER1 and enable SDA/SCL_MASTER0.

Determining Pull-Up Resistors values

The PCA954X devices 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 (i.e., bus capacitance and device input loads) must be taken into consideration when choosing the value of a pull-up resistor.

Note that the PCA954X supply should normally be connected to the lowest bus voltage 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:

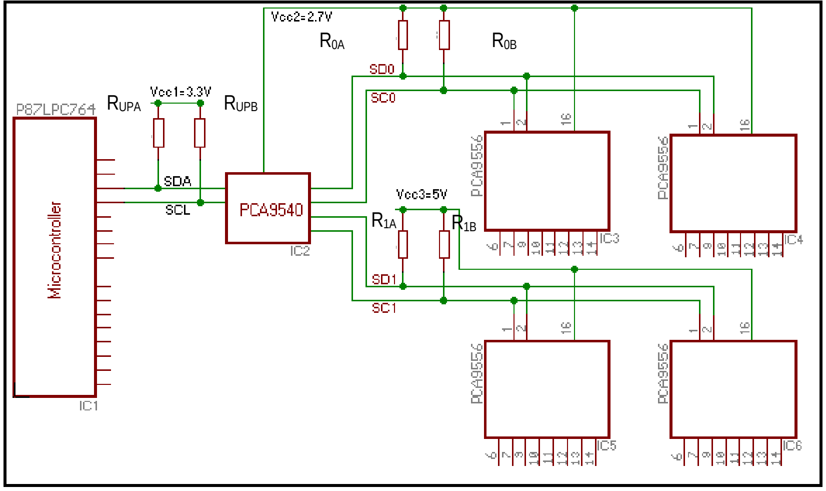


Figure 9. Typical Circuit Example

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} = 3 mA at 0.4 V. This determines the minimum resistor value.
2. Ensuring that the rise time does not exceed 1.0 µs 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). This determines the maximum resistor value.

   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 3 mA 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 = 4.4 kΩ
$R_{0A} = R_{0B}$ = 2.5 V / 2.25 mA = 1.11 kΩ
$R_{1A} = R_{1B}$ = 5.0 V / 2.25 mA = 2.2 kΩ

Additional verification:
Ensure the rise time specification of 1 µs for standard mode I²C is not exceeded. Consider the V_{DD}-related input threshold of $V_{IH} = 0.7 \times V_{DD}$ and $V_{IL} = 0.3 \times V_{DD}$ 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.
$V(t1) = 0.3 \times V_{DD} = V_{DD} (1–1/e^{-t1/RC})$; then t1 = 0.3566749 x RC
$V(t2) = 0.7 \times V_{DD} = V_{DD} (1–1/e^{-t2/RC})$; then t2 = 1.2039729 x RC
T_{rise} = t2 – t1 = 0.8472979 x RC

Scenario 1: No downstream channel enabled

$T_{rise} = 0.8472979 \times R_{UPA}C_{upstream}$
$= 0.8472979 \times 4400 \times 100e\text{-}12$
$= 0.37$ µs

Scenario 2: Channel 0 enabled

$T_{rise} = 0.8472979 \times (R_{UPA} // R_{0A})(C_{upstream} // C_{channel0})$
$$= 0.8472979 \times \frac{4400 \times 1110}{4400+1110} \times (100\text{ e-}12 + 300\text{ e-}12)$$
$= 0.30$ µs

Scenario 3: Channel 1 enabled

$$T_{rise} = 0.8472979 \text{ x } (R_{UPA}// R_{1A})(C_{upstream} // C_{channel1})$$

$$= 0.8472979 \text{ x } \frac{4400\text{x}2200}{4400+2200} \text{ x } (100 \text{ e-}12 + 200 \text{ e-}12)$$

$$= 0.37\ \mu s$$

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

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

The example in Figure 10 is made with the PCA9543A, 2-channel switch. It can easily be extended to the 4 and 8-channel devices.

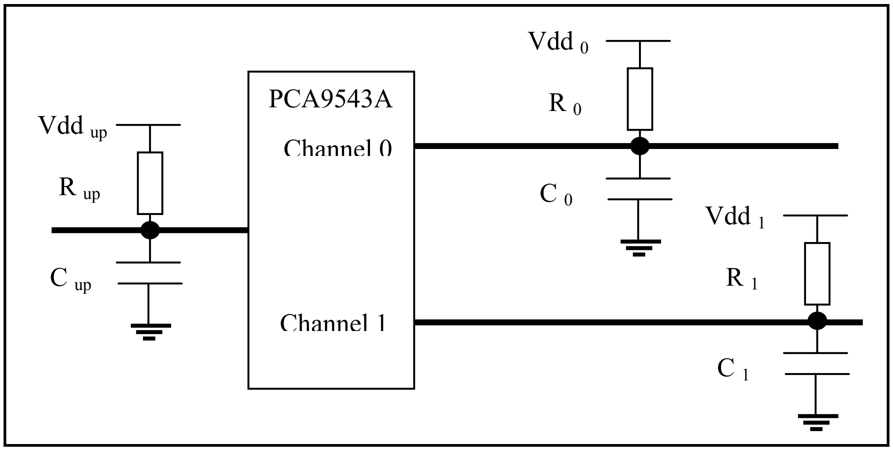


Figure 10. Multiple Channel Example

- 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^{2}C-bus and can be connected to no downstream channels, to channel 0, channel 1 or both channel 1 and 2 at the same time.

1. No channel selected:
   - Equivalent resistance of the I^{2}C-bus = R_{up}
   - Equivalent capacitance of the I^{2}C-bus = C_{up}
2. Upstream channel connected to downstream channel 0:
   - Equivalent resistance of the I^{2}C-bus = $R_{up} // R_0 = (R_{up} \text{ x } R_0)/ (R_{up} + R_0)$
   - Equivalent capacitance of the I^{2}C-bus = $C_{up} // C_0 = C_{up} + C_0$

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

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

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

- Quick overview: Maximum and minimum pull-up resistors values

The supply voltage limits the **minimum** value of the pull-up resistor due to a minimum sink current value of 3 mA at V_{OL}max = 0.4 V (see Figure 11).

The bus capacitance is the total capacitance of wire, connections and pins. This capacitance limits the **maximum** value of the pull-up resistors due to the specified rise time (see Figure 12 and Figure 13).

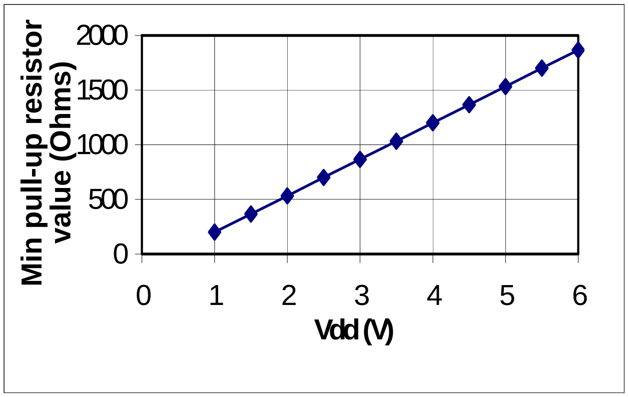


Figure 11. Min pull-up resistor value

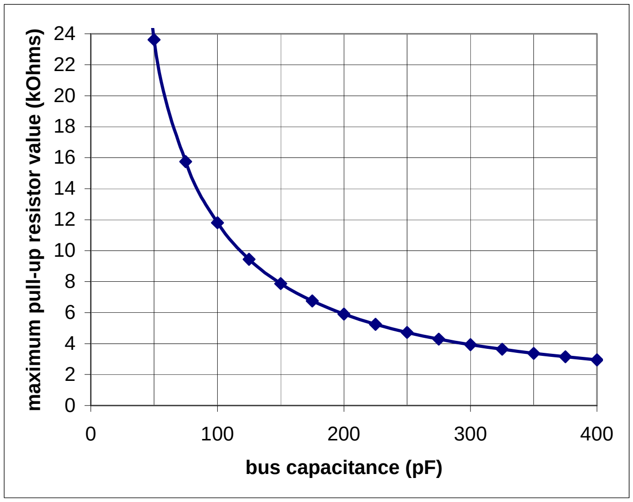


Figure 12. Max pull-up resistor value (standard mode)

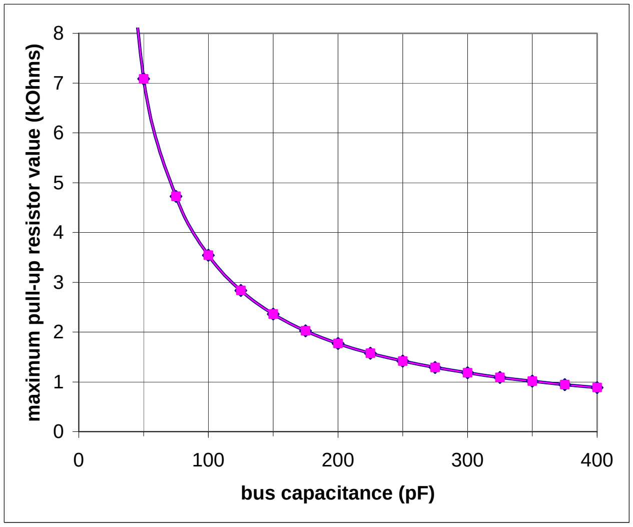


Figure 13. Max pull-up resistor value (Fast-mode)

APPLICATIONS

I²C Multiplexing

Some specialized devices only have one I²C address and sometimes several identical devices are needed in the same system. The multiplexers and switches split the I²C-bus into several sub-branches and allow the I²C master to select and address one of these identical devices at a time, to avoid address conflict issues.

In many PCs, the Serial Presence Detect (SPD) EEPROM on DIMMs respond to the same I²C address (0xA0). Therefore, if more than one DIMM is installed in the card, a method must be determined to read the data from each EEPROM. This can effectively be done using the PCA9540B as shown in Figure 14.

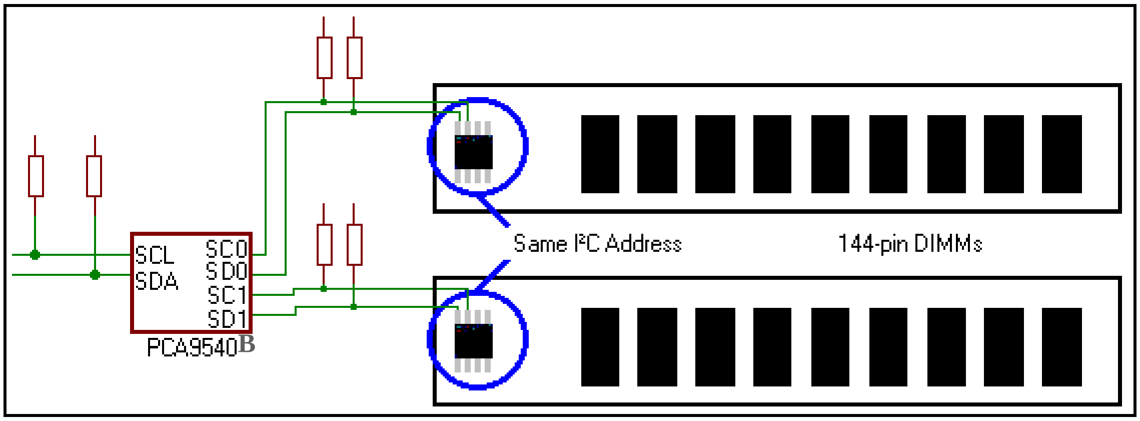


Figure 14. Example of DIMM Serial Presence Detect Application

Generally speaking, any application requiring multiple identical devices with the same I²C address on a single bus can use one or several PCA954X devices to multiplex the I²C-bus and solve any addressing conflict.

Voltage Level Shifting

I²C and SMBus devices can operate at different voltage levels but may need to operate on a common bus. The PCA954X multiplexers and switches allow voltage translation between 1.65 V and 5.5 V and provide a solution to this problem. For example, a 5 V I²C 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) I²C device on channel 1. The device channel pass gates are constructed such that the V_{DD} pin is 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. All PCA954X exhibit this behavior but the PCA954X Switches are best for voltage level shifting application since multiple downstream channels can be active at the same time.

Capacitive Load Sharing

Adding more I²C and SMBus devices on the bus may exceed the 400 pF limitation. A multiplexer or 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 both the upstream channel and all active downstream channels must be considered and maintained below 400 pF. If total active channel system loading must be above 400 pF, then the Philips PCA9510/11/12/13/14/15/16/17/18 I²C bus repeaters and hubs should be considered.

Typical Application

Figure 15 shows the PCA9545A in a typical application that could be used for voltage level translation of multiplexing to one downstream channel at a time.

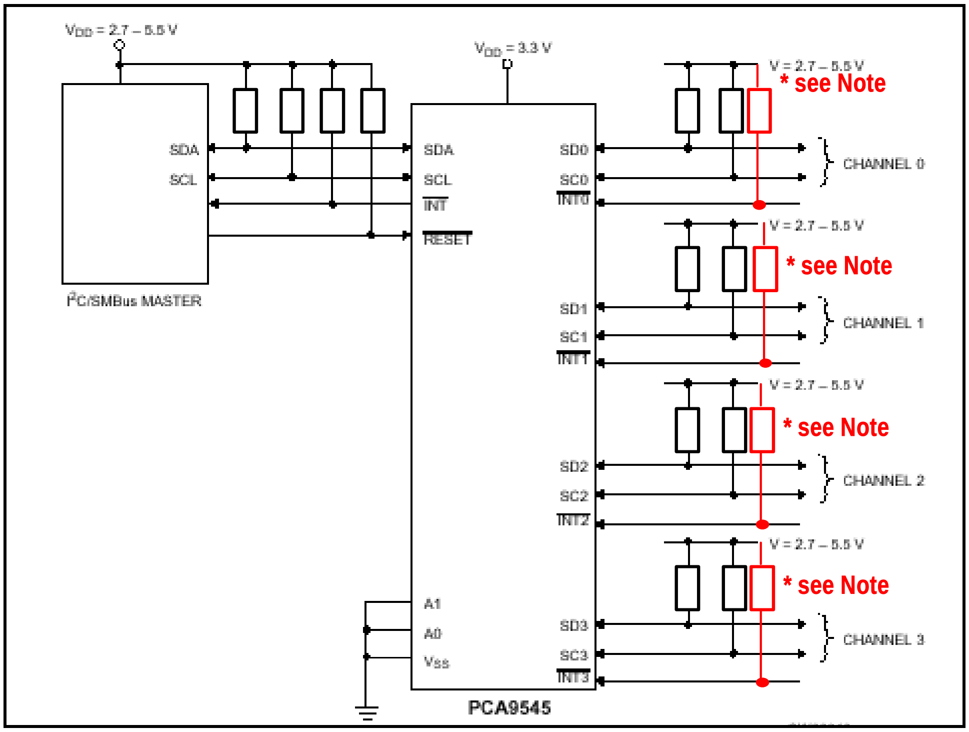


Figure 15. PCA9545A In a Typical Application

Note:
Interrupt input pull-up resistors sizing is discussed on page 9.

FREQUENTLY ASKED QUESTIONS

1. **Question:** What is the impedance of inputs/outputs when the part is powered down?
   Answer: The I^{2}C channel inputs/outputs will have a high-impedance value if the voltage applied to that I/O 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^{2}C controller. I now see that the PCA954X does not even have an internal timer for that. Another use I thought of is for transition detection for the Sxx line.
   Answer: The way the Interrupt inputs are implemented in these devices is more of an OR function, if one of the interrupts inputs goes low then the interrupt output also goes low. The interrupt is not latched. When the low interrupt input signal goes away, 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 PCA9544A? The supply voltage of the PCA9544A 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: Yes, a different voltage bus can be connected to the I/O of the PCA9544A. This is shown in Figure 7. The PCA9544A powers up with the channels open (high impedance) so if the 2.5 V powers up after the 3.3 V, there is no problem. Also, if the 3.3 V powers-up after the 2.5 V, there will be no problem because the pins of the PCA9544A will be high-impedance when the part is unpowered.

5. **Question:** How do we calculate pull-up resistor values? Do we need to calculate the upstream pull-up values in one side and the downstream pull-up values in the other or are they correlated?
   Answer: The PCA9544A 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. For example, 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 PCA954X, 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 master on 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 to select Channel 1 in order to address a slave device on this downstream channel. I need to access this downstream device several times. Do I have to access the PCA954X 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(s) 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(s). 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 channels are selected.

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 is connected to the upstream bus master channel. The PCA9548A 8-channel switch and any of the 4 channel multiplexers/switches could also be used in any combination

9. **Question:** Does the PCA9544A support 2.5 V I^2C signal at $V_{DD} = 3.3$ V?
   Answer: Figure 7 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 PCA9544A will support downstream channels at 2.5 V, 3.3 V and 5.0 V if powered at 3.3 V V_{DD} but for best results, the PCA9544A should be powered at 2.5 V.

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 will be no latch-up or damage, permanent or otherwise, to the PCA954X device under these conditions. In addition, the device will not cause any disruption to the I^2C-bus.

11. **Question:** Can we use both 3.3 V and 5 V I^2C buses 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 but for bst results, the PCA954X should be powered at the lowest voltage needed.

12. **Question:** Are there any voltage shifts on the I^2C buses when using the PCA954X?
    Answer: Yes, the voltage drop across the switch due to any current through it must be taken into consideration. The PCA954X uses pass transistors that have a specified R_{on} value. If PCA954Xs are used in series, all the series resistance of each device will add to each other. The resistance per device can be as high as 55 ohms over temperature. This shouldn't be an issue in most applications but the customer should be aware of this potential problem if they put a large number of PCA954Xs in series. This parameter can be critical in low voltage I^2C systems.

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 PCA954X devices on the active bus) and capacitive load (400 pF) since the PCA954X devices don't have the ability to isolated the capacitive load. If used in series, the cumlative device resistance should also be considered.

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 using a hardware solution. 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 to restore the upstream bus. The PCA9516 Hub can also be used to isolate downstream channels 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 used only to send information bi-directionally between an upstream bus and one/several downstream buses. The devices can't send any kind of I^2C sequence or initialize/reset some devices on their own. When the device is reset, it 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 effect on the slave (downstream device) when the PCA954X Switch reset pin is asserted?
    Answer: There is no effect on the downstream device. The Reset only disables all the downstream channels.

18. **Question:** If the interrupt pins are not used, can they be left floating?
    Answer: The Interrupt inputs don't have any internal pull-up resistors and must be tied to V_{DD} if they are not used.

19. **Question:** How is it that your part controls the SCL line for data rate transfer if you do not have any control of the SCL line?
    Answer: The PCA9542A is an I^2C controlled NMOS pass transistor multiplexer and the I^2C interface is a simple slave with no SCL driver, so it cannot hold the SCL line low. However when a channel is selected using the I^2C interface, at the next stop condition the NMOS pass transistor is turned on making a slightly resistive connection between the SDA and SD0 or SD1, and a parallel path between the SCL and SC0 or SC1, similar to connecting the upstream and downstream buses together with a wire. This low resistance connection means that a low on one side is passed to the other side in real time with no logic interference from the PCA9542A. If the SCL is stuck low it cannot be the PCA9542A, however a slave on one side with its SCL stuck low will pass through the PCA9542A as a low if the channel is selected. A master with its SCL stuck low will also pass through the PCA9542A as a low to

whichever channel is selected. The PCA9542A is incapable of generating a low on the SCL it only passes the low along. Since the PCA9542A has some resistance checking the low on both sides of the PCA9542A when the system is hung will tell you which side has the offending part because the side with the lowest SCL voltage is the side that is being driven and the higher voltage corresponds to the signal passing through the NMOS pass transistor (switch).

20. **Question:** The SCL line is being held low, which is hanging the I^2C-bus. Is it possible that the PCA9542A could hold this line low? All of the devices on the 9542A's output are slave devices and all of the devices on the 9542A's input are masters. Is it possible that the 9542A is holding the SCL line low because a slave device is not generating the SCL signal?
Answer: The PCA9542A does not have an output driver on the SCL pin so it could not hold the SCL line low. Another device on the bus must be holding the SCL line low that can be passed through the PCA9542A.

21. **Question:** If I wanted to test the Slave components on the SC0 and SC1 side of the MUX for this issue as being the culprit, how would you modify the PCA9542A to isolate SC0/SC1 line to isolate the slave components? Would you lift the SC0 pin and let it float or pull it high?
Answer: The simplest answer is to not select either SC0 or SC1, this should leave the slaves isolated. Lifting the SC0 / SC1 and letting it float would also isolate the slaves. Any pull-up on the PCA9542A would need to be considered part of the SCL (or SDA) total pull-up current when selected, so any pull-up would need to be resistive. All three methods should make the SCL isolated. SC0 and SC1 should be pulled up by their own resistors.

22. **Question:** What size pull-up resistors should I use in this application and what should I do with the unused channels? Is there a difference if I operate at 100 kHz or 400 kHz? The capacitance of the upstream bus between the master and PCA9548A is 22 pF per device or about 100 pF total. Each downstream bus has an identical device with a fixed I^2C address so only one channel will be activated at a time. The slave device and downstream bus trace have a capacitance of 10 pF.

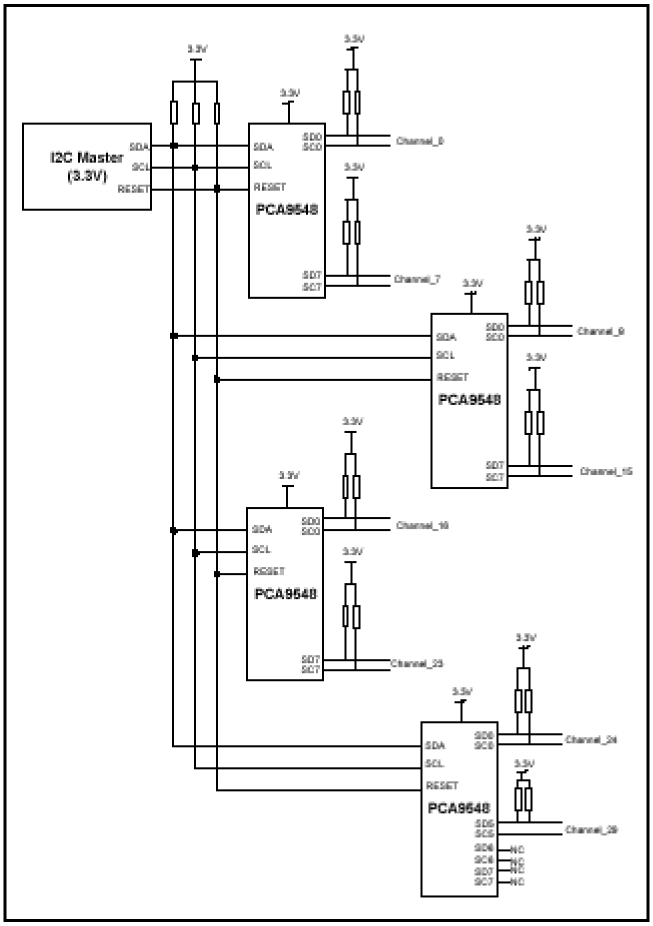


Figure 16. Typical Application using the PCA9548A

Answer: Pull-up resistor calculations

- 100 kHz mode:

  - Static load:

    Maximum load current specified = 3 mA
    $V_{ol\ max}$ = 0.4 V at 3 mA
    A 3.3 V bus is used: R_{min} = (3.6 -0.4)/3 mA = 1.1 kΩ.
    Minimum value for the bus is then 1.1 kΩ.
    This value is the total resistance of the I^2C-bus.
    Since the bus is divided in 2 parts with the PCA9548A, upstream and a downstream pull-up resistors need to be taken into account.
    $R_{up\ mini}$ = 2.2 kΩ
    $R_{down\ mini}$ = 2.2 kΩ
    R_{total} is then 1.1 kΩ

  - Dynamic load:

    The maximum value of the rise time is 1µs
    C_{up} = 4x22 pF
    C_{down} = 10 pF
    C_{total} is about 100 pF
    T_{rise} = 0.847 x R_{total} x C_{total} = 0.1 µs

- 400 kHz mode:

  If the same resistor values need to be used at 400 kHz, it's important to determine that they will not violate the max rise time value is 0.3 µs. According the calculations above, the resistor values look fine for a 400 kHz bus.

  Margin:

  Since the bus is not fully loaded, it's possible to take some margin: 2.7 kΩ upstream and 2.7 kΩ downstream are also fine for both 100 kHz and 400 kHz.

- Unused Channel:

  The unused channels should have the same size pull-up resistors as used for the other channels.

23. **Question:** are the addresses located on the downstream side of the PCA954X devices "isolated" from the addresses on the upstream side or do all the addresses in the entire system have to be factored into the determination of available addressing space?

Answer: you can't have more than one I^2C multiplexer/switch with the same slave address in your system. An I^2C multiplexer/switch behaves like any other I^2C slave device: if several devices have the same address and are present in the same bus then they will all respond to the master sending this slave address and will perform the exact same operation (assuming a write operation). Figure 17 illustrates the different conflicts that can occur.

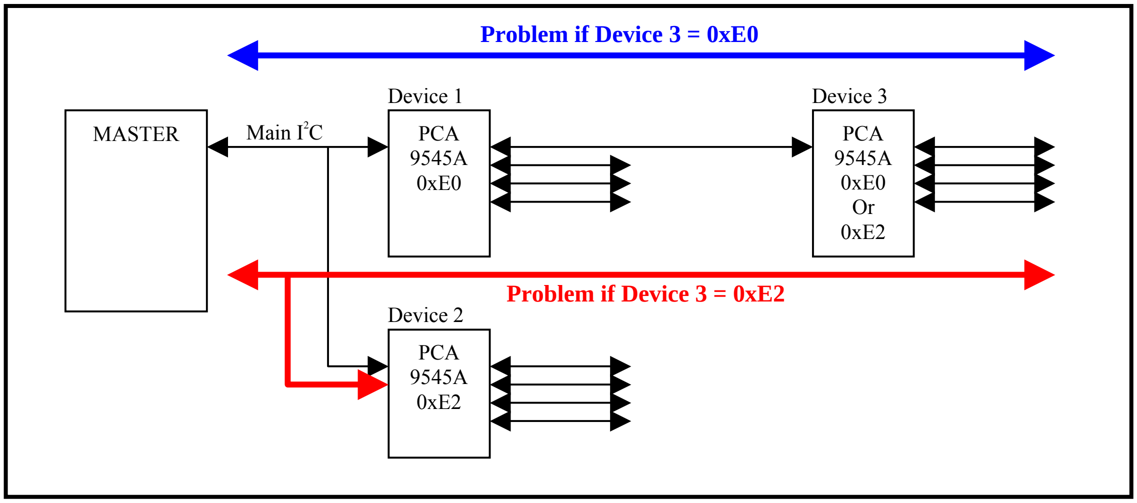


Figure 17. Address conflicts using several PCA9545A

If Device 3 = 0xE0: there is a problem when Device 1 is programmed to enable the downstream channel where Device 3 is connected. When the master now addresses Device 3 to program it, then Device 1 will also be programmed to the same value (and can actually disconnect Device 3 depending on the data sent by the master).

If Device 3 = 0xE2: there is a problem when Device 1 is programmed to enable the downstream channel where Device 3 is connected. When the master addresses Device 3 to program it, then Device 2 will also be programmed to the same value

Note: PCA9545A has 2 programmable address pins then the 3 devices can be programmed to different values to get rid of that problem.

24. **Question:** We are using a PCA9546A (V_{DD} = 3.3 V) where 2 downstream channels are connected to 5 V (one of them has a secondary master connected to it), the 3rd channel is connected to a NVRAM (3.3 V). The 4th channel is not used. When the main master (upstream) is connected to the NVRAM (only this one, the other channels are disabled), we noticed some glitches (causing actually errors during read back from the NVRAM) when the secondary master is sending I^2C commands at the same time (to some other devices in the downstream channel where it is located. Since the secondary master is disconnected from the main PCA9546A communication channel, we would not expect any glitches with the active channel. What could cause this crosstalk to happen?
Answer: I agree that as long as the secondary master is disconnected there should be no cross talk. There is however a special case where if the secondary master is driving very fast edges, so that the low level goes significantly below the ground of the PCA9546A (i.e. low level = -1 V) then the signal can couple and degrade the high level on the main channel. Since the I^2C specification limits the minimum edge rate this should not happen. If however the secondary master violates the edge minimum, it is then possible to use a series resistor on the output of the master to slow down the edge and prevent the problem to happen.

25. **Question**: I want to insert an I^2C switch in my system where the I^2C devices are supplied with 2.5 V (going down to 2.3 V) and controlled by a 1.8 V I^2C-bus. Would that work?
Answer: I cannot think of any reason why this would not work. Our part is simply an NMOS pass gate, so it passes a low well and a high “poorly”. The supply voltage is used to charge the gate and turn the transistor on (or close the switch). We use the characteristic of passing a high “poorly” to allow the part to act as a voltage translation stage. This is because when the voltage on the I/O's is above our parts supply voltage less a threshold, the pass gate is now off. This is why it is important for all busses to have individual pull-up resistors.

26. **Question:** Should the reset line on the PCA9546A be used only as a last resort, meaning the I^2C-bus got hung after making a connection? What happens if the reset is used as a disconnect means instead of writing to the control register to disconnect? It seems to me that the reset should only be used as a failure recovery and not a standard means of disconnecting from a channel. The software Engineer saw this as an opportunity to write less code. When disconnecting it was easier to hit the reset line versus sending the disconnect information.
Answer: Difference between the 2 methods is:
    - The Reset will put the device in its Power-up state, in a No Channel select mode, with all the registers and state machine reset.
    - The "No channel select" by writing the control register will just disconnect all the downstream channels but will not initialize the device.

    So to answer the question, the reset was not envisioned to be the normal method to open the switch channels but it will not damage the device and since the device does not have to be addressed and command byte/stop sent, it will be faster. Both methods can be used to disconnect the I^2C-bus.

27. **Question:** I'm developing a system that requires a master to control more than 70 nodes. Using devices like PCA9548A below the master limits the system to 64 nodes (8x8: the PCA9548A is a 8-channel device and up to 8 of them can be used thanks to the 3 programmable address pins). In addition, each node has seven I^2C slaves that should be connected only during boot, or during intial burning of the system (EEPROMs). I would like to isolate them during the regular operation. How can I form such a topology using Philips devices?
Answer: The switches and multiplexers can also be placed in series. The only requirements for such a configuration is to be sure that:
    - the total capacitance is less than 400 pF on the active buses
    - there is no address conflicts between the devices.

    Several configurations are then possible. For example:
    - One 4-channel device, like the PCA9546A (4 segments) with three PCA9548A on each downstream channel (24 channels per segment) thus allowing up to 8x3x4 = 96 nodes.
    - One 4-channel device with two PCA9548A and one PCA9546A thus allowing up to (8+8+4)x4 = 80 nodes.

Then from the chosed configuration above, a 3rd level eliminating the address conflicts between the EEPROM's can be performed with a PCA9548A (8 nodes of which you need 7). Therefore you would have only five of the switches active at a time:

- 1st level: one PCA9546A connected to the master device
- 2nd level:three PCA9548A or two PCA9548A plus one PCA9546A connected to each downstream of the 1st level PCA9548A
- 3rd level: one PCA9548A active and no address conflicts to each downstream channel of the 2nd level devices.

28. **Question:** I am using a PCA9544A. If there is a short on one of the output buses (say to GND), can that also short out the input bus, therefore preventing any further communication with that host device?
Answer: Once the main and sub-channel (outputs) are connected internally via I^{2}C command, the device acts as a wire and if you ground one side then the other side will be grounded also. If the sub-channel is not active (not hooked to the main channel) then nothing done to that sub channel will be seen on the main channel since the wire is not connected.

29. **Question:** Can each of the 4 output buses on the PCA9545A have 400pf of load each?
Answer: The PCA9545A does not isolate capacitance. The PCA9545A can have 400 pF on each output only if the input has no capacitance and only one output channel can be active at one time. If you have 100 pF in the input bus then you can have 300 pF on each output channel if you have only one of the four channels active at one time or 75 pF on each channel if all four channels are active at the same time.
A different device, the PCA9516 hub can have 400 pF on each segment and all segments can be active at the same time. There are hardware enable pins on the PCA9516 that individually activate/deactivate the 4 output channels. These enable pins can be controlled by the ucontroller GPIO.

ADDITIONAL INFORMATION

The latest datasheets for the PCA954X family of products and other SMBus/I^{2}C products can be found at the Philips Semiconductors website:
http://www.philips.com/i2clogic

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: I2C.support@philips.com

REVISION HISTORY

Revision	Date	Description
_2	20041027	Application note, second version (9397 750 14252). Supersedes data of 10 October 2003 Modifications: • Added PCA9541, PCA9547 and PCA9549 devices • Modified part numbers: PCA954X to PCA9540B, PCA9542A, PCA9543A, PCA9544A, PCA9545A, PCA9546A, PCA9548A due to ABQ fab closure • Added paragraph "Power-up / Reset default state" (page 13) • Added HVQFN package for the PCA9544A, PCA9545A, PCA9546A and PCA9548A (Pinout and ordering information) • Added Q&A 23 to 29.
_1	20031003	Application note, initial version .

Disclaimers

Application information – Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Life support – These products are not designed for use in life support appliances, devices or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes – Philips Semiconductors reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Contact information
For additional information please visit
http://www.semiconductors.philips.com

Fax: +31 40 27 24825

For sales offices addresses send e-mail to:
sales.addresses@www.semiconductors.philips.com

Date of release: 10-04
Document number: 9397 750 14252