



I2C 2005-1 Demonstration Board

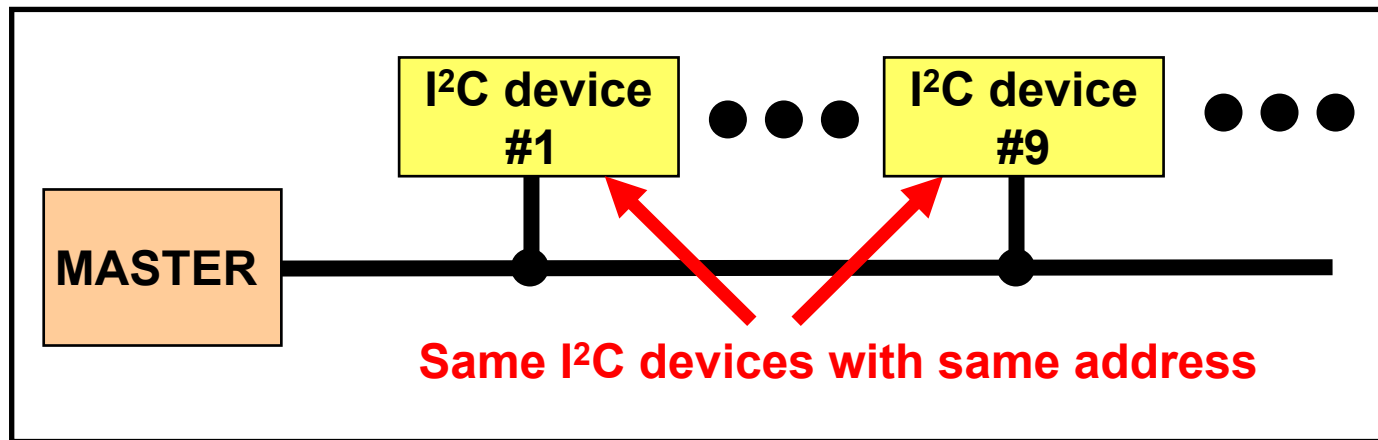
Multiplexer/Switch and Voltage Translation

Oct, 2006



How to solve I²C address conflicts?

I²C protocol limitation: More than one device with the same I²C address is not allowed on the bus. How do we solve this issue for a customer who needs, for example, sixteen PCA9531s in his system when there are only three programmable bits (eight potential addresses)?



How to solve I²C address conflicts?

- I²C protocol limitation: More than one device with the same I²C address is not allowed on the bus

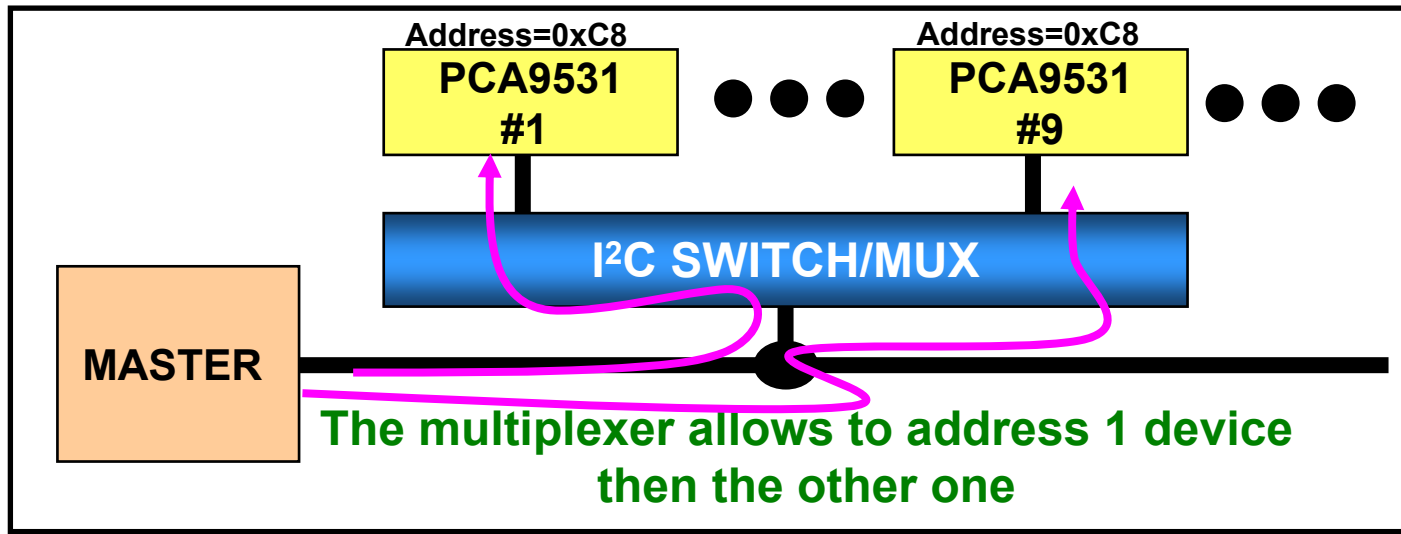
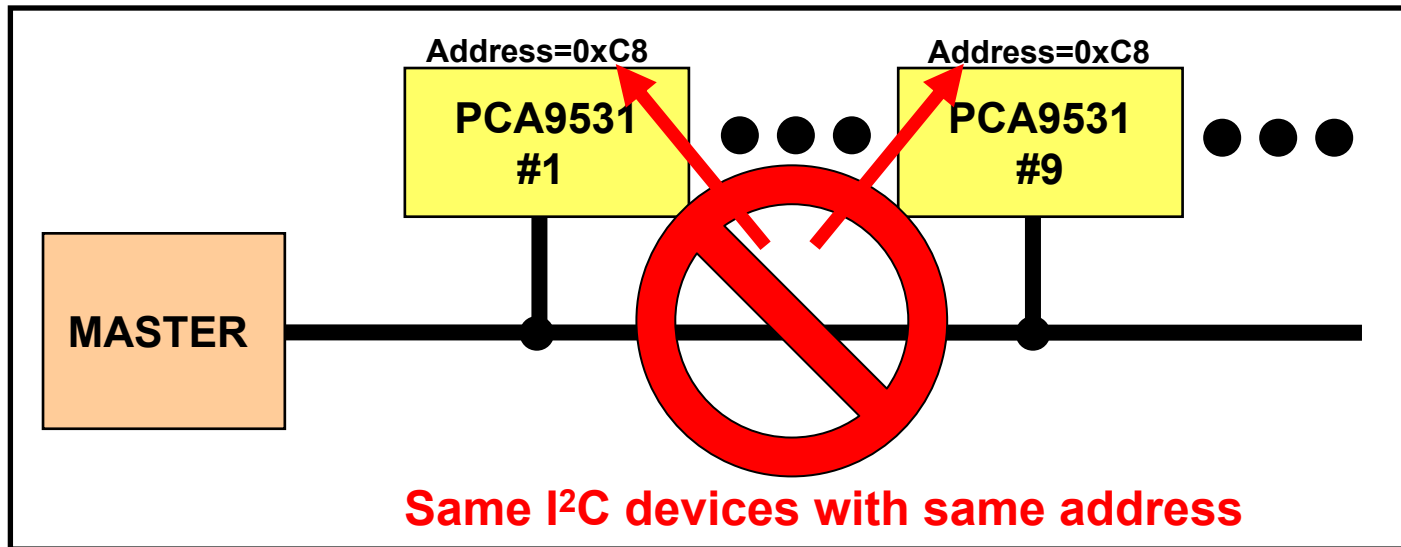
An I²C multiplexer/switch allows more than one device with the same address on the bus

- Allows to split dynamically the main I²C-bus into several isolatable sub-branches
- Programmable through I²C so no additional pins are required for control

# of Channels	Standard	With Interrupt Logic
2	PCA9540B	PCA9542A / PCA9543A*
4	PCA9546A*	PCA9544A / PCA9545A*
8	PCA9547* / PCA9548A*	

Parts marked with * also have a reset

I²C Multiplexers: Address Deconflict



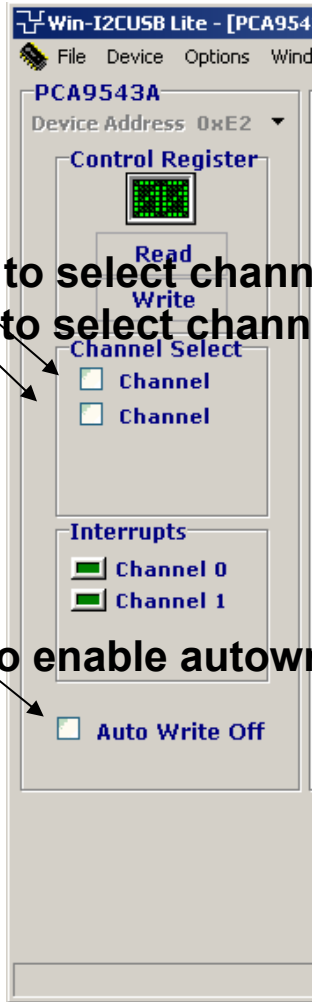
[illegible]

Channel 0



PCA9543A GUI Introduction

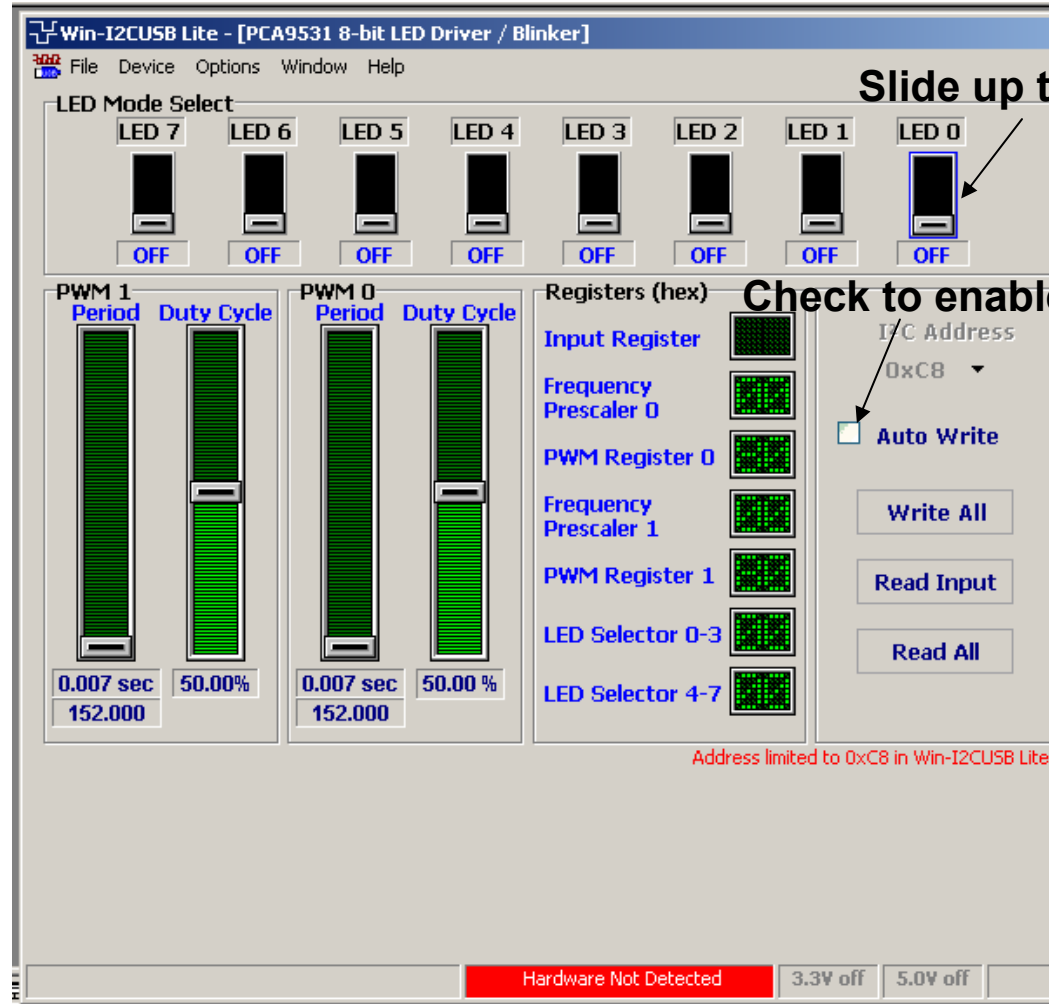
PCA9543A GUI



Check to select channel 0
Check to select channel 1

Check to enable autowrite

Use PCA9531 GUI to turn LEDs On

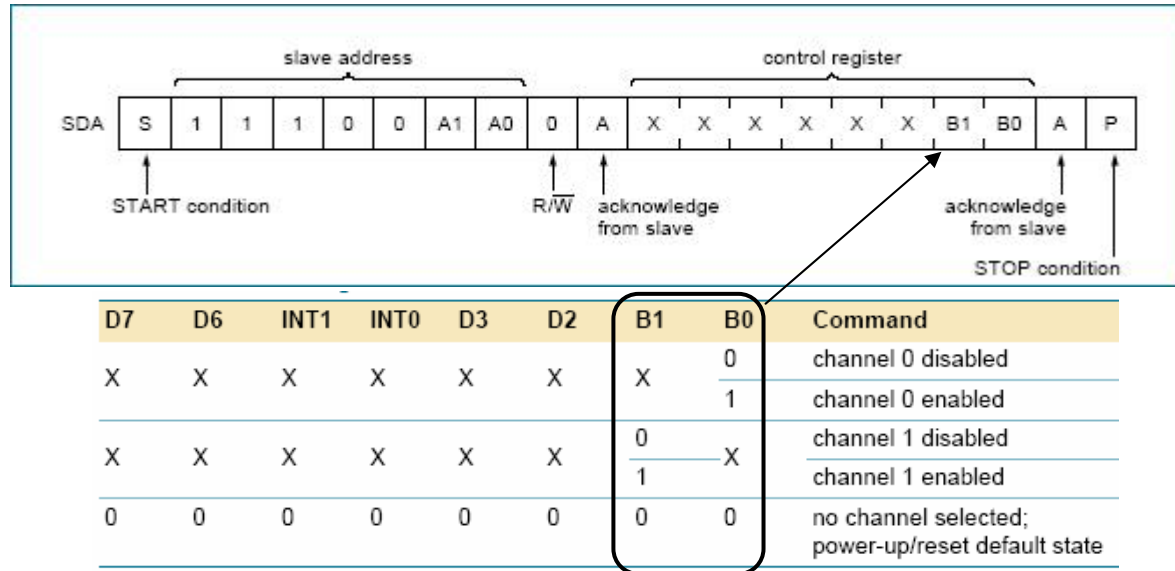


Slide up to turn LED0

Check to enable autowrite

Address limited to 0xC8 in Win-I2CUSB Lite

Exercise: Using a mux/switch to resolve address conflicts

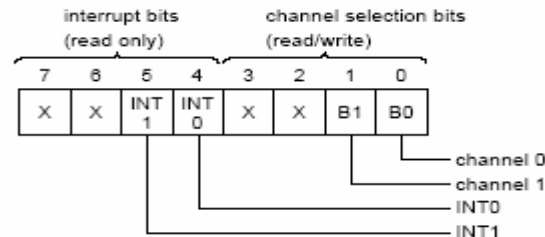


- Using the PCA9543A, enable channel 0
- Using the PCA9531, turn-on LED 0
- Using the PCA9543A, enable channel 1
- Using the PCA9531, turn-on LED 1
- Using the PCA9543A, enable channels 0 and 1
- Using the PCA9531, turn-on LED 3

Exercise: use the PCA9543A interrupt and Reset Features

First power cycle the board, then using the Expert Mode:

1. Enable both channels in the PCA9543A
2. Configure IO1, IO2 and IO3 in the PCA9538 to outputs
3. Set IO2 low
4. Read the PCA9543A and observe bit set to 1 and both channels enabled
5. Set PCA9538 IO1, IO2 and IO3 to 0 and wait 1 ms
6. Set PCA9538 IO1 to 1, IO2 and IO3 to 0
7. Read the PCA9543A and observe additional bit set to 1 and all channels disabled



D7	D6	INT1	INT0	D3	D2	B1	B0	Command
X	X	X	X	X	X	X	0	channel 0 disabled
X	X	X	X	X	X	X	1	channel 0 enabled
X	X	X	X	X	X	0	X	channel 1 disabled
X	X	X	X	X	X	1	X	channel 1 enabled
0	0	0	0	0	0	0	0	no channel selected; power-up/reset default state

How to go beyond I²C max cap load?

- I²C protocol limitation: Maximum capacitive load in a bus is 400 pF. If the load is higher, AC parameters will be violated

Two possible solutions:

- 1. I²C multiplexer or switch**
- 2. I²C buffer**

How to go beyond I²C max cap load?

- I²C protocol limitation: Maximum capacitive load in a bus is 400 pF. If the load is higher, AC parameters will be violated

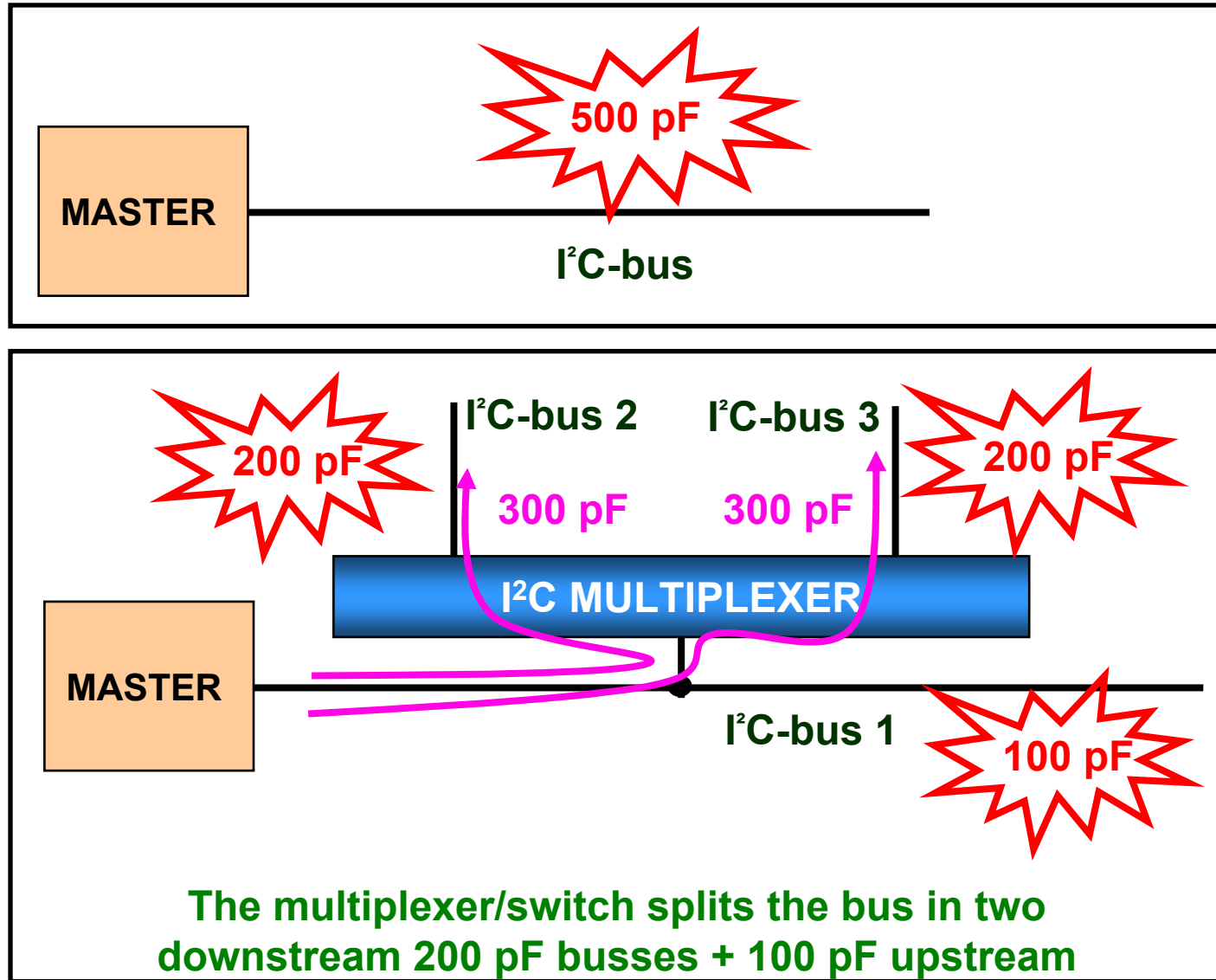
An I²C multiplexer/switch can be used to segment the capacitance

- Allows to split dynamically the main I²C-bus in several sub-branches in order to divide the bus capacitive load
- Programmable through I²C so no additional pins are required for control
- More than one multiplexer can be plugged in the same I²C-bus
- LIMITATION: All the sub-branches cannot be addressed at the same time

# of Channels	Standard	With Interrupt Logic
2	PCA9540B	PCA9542A / PCA9543A*
4	PCA9546A*	PCA9544A / PCA9545A*
8	PCA9547* / PCA9548A*	

Parts marked with * also have a reset

I²C Multiplexers/Switch: Capacitive load split



I²C-bus Voltage Level Shifting

Problem:

A customer has an application where they are using devices with various operating voltages tied to the I²C-bus. Provide details on possible solutions to the level shifting issue.

Level-shifting possibilities:

1. Do not do level shifting
2. PCA954x multiplexer/switches
3. Use FETs
4. GTL2002
5. PCA9306
6. RETs
7. PCA9517/PCA9509 buffer
8. PCA9512 hot swap buffer

**Bus buffering
and translation**



I²C-bus Voltage Level Shifting

Solution #1

Don't do any level shifting.

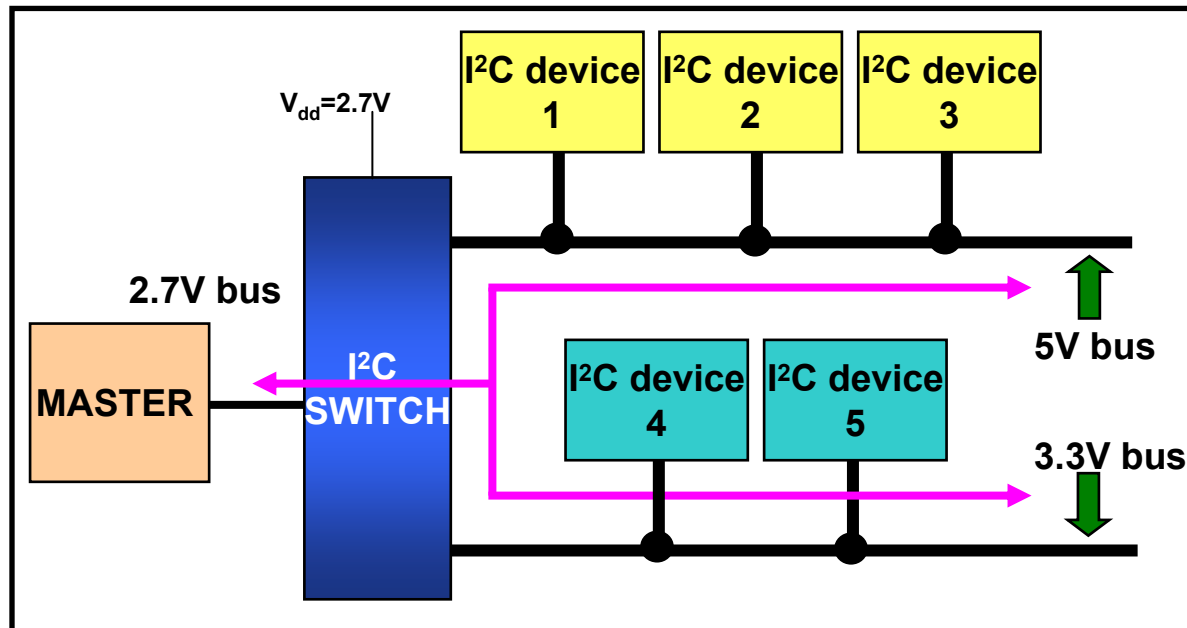
In many situations, the open-drain SDA and SCL pins are over-voltage tolerant (no ESD protection diodes to V_{DD}), so no level-shifting is required.

Example, from PCA9552 datasheet:

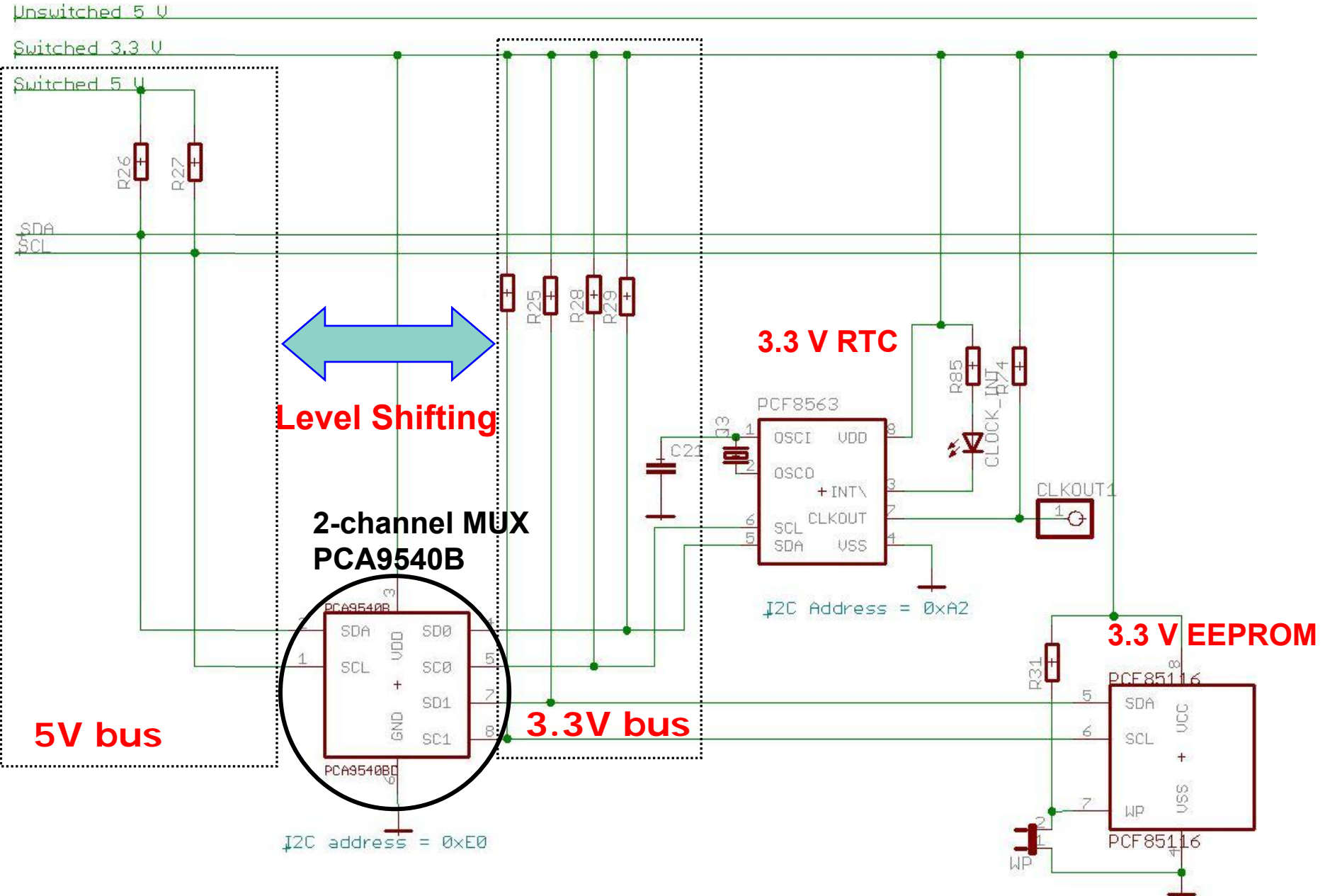
Input SCL; input/output SDA						
V_{IL}	LOW-level input voltage		-0.5	—	$0.3 V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7 V_{DD}$	—	5.5	V

I²C-bus Voltage Level Shifting

Solution #8 Multiplexer / Switch

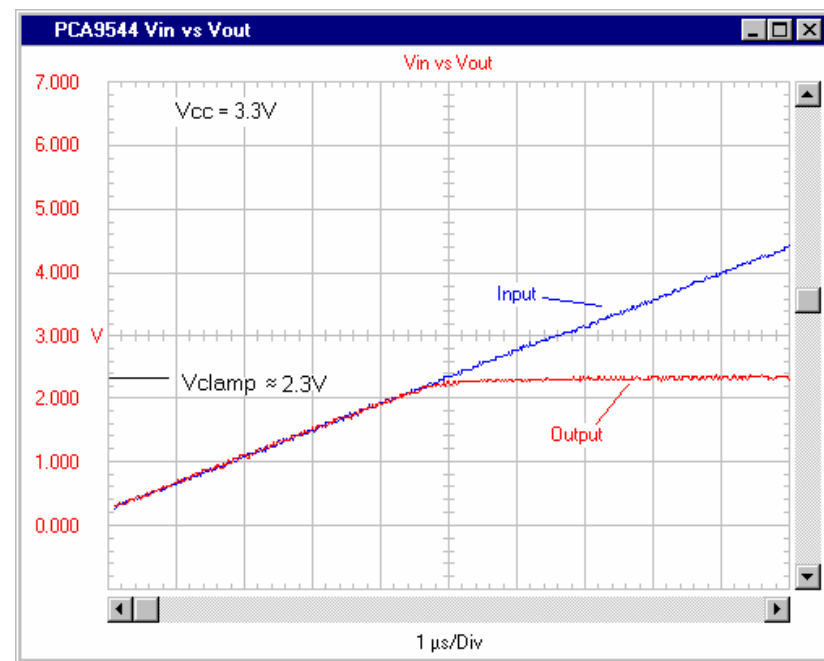


- The voltage through the device is limited by V_{DD} of the Switch
- Each branch must be pulled up to its respective bus voltage



How Voltage Translation Works In MUX/Switches

- 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})
- The clamping voltage will be somewhat lowered by a load on the output (shown with open output).



How to isolate failing slave devices?

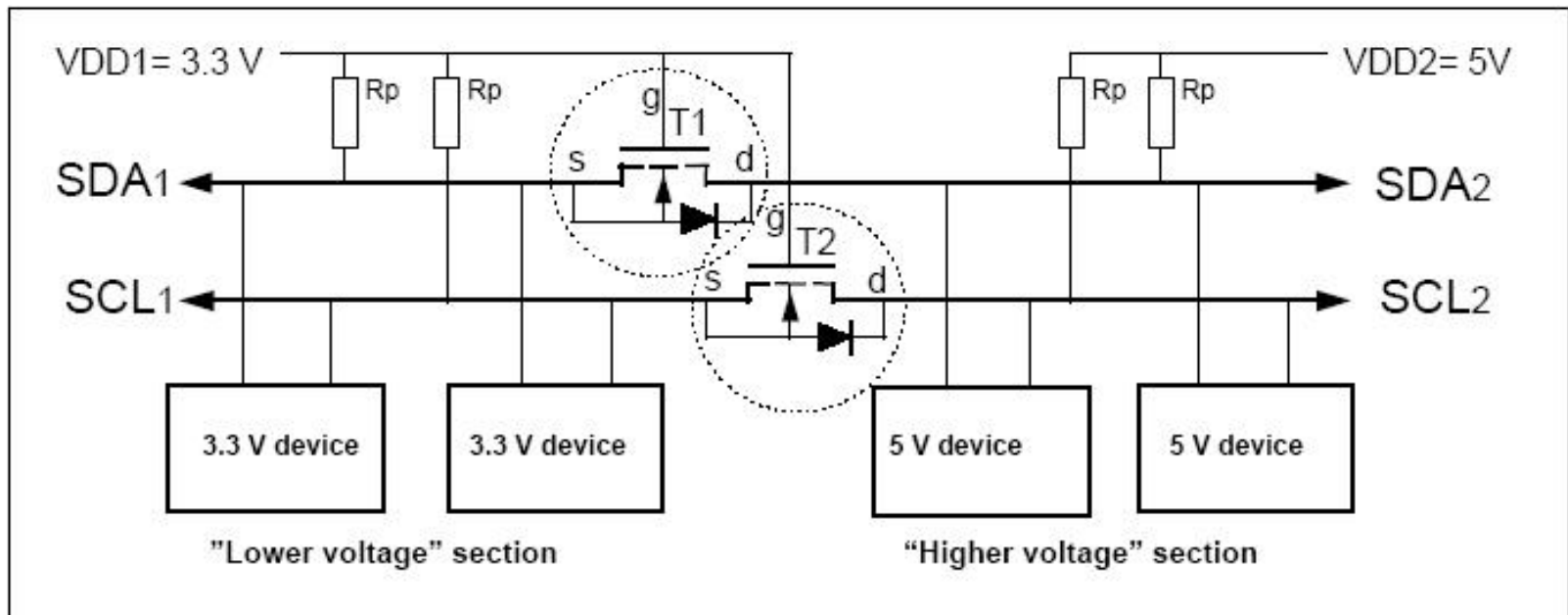
- I²C protocol: If one device does not work properly and hangs the bus, then no device can be addressed anymore until the rogue device is separated from the bus or reset

An I²C switch can split the I²C-bus in several branches that can be isolated if the bus hangs up

- Switches allow the main I²C to be split dynamically in several sub-branches that can be:
 - active all the time
 - deactivated if one device of a particular branch hangs the bus
- When a malfunctioning sub-branch has been isolated, the other sub-branches are still available
- Programmable through I²C so no additional pin is required to control it
- More than one switch can be plugged in the same I²C-bus

I²C-bus Voltage Level Shifting

Solution #2 Using FETs

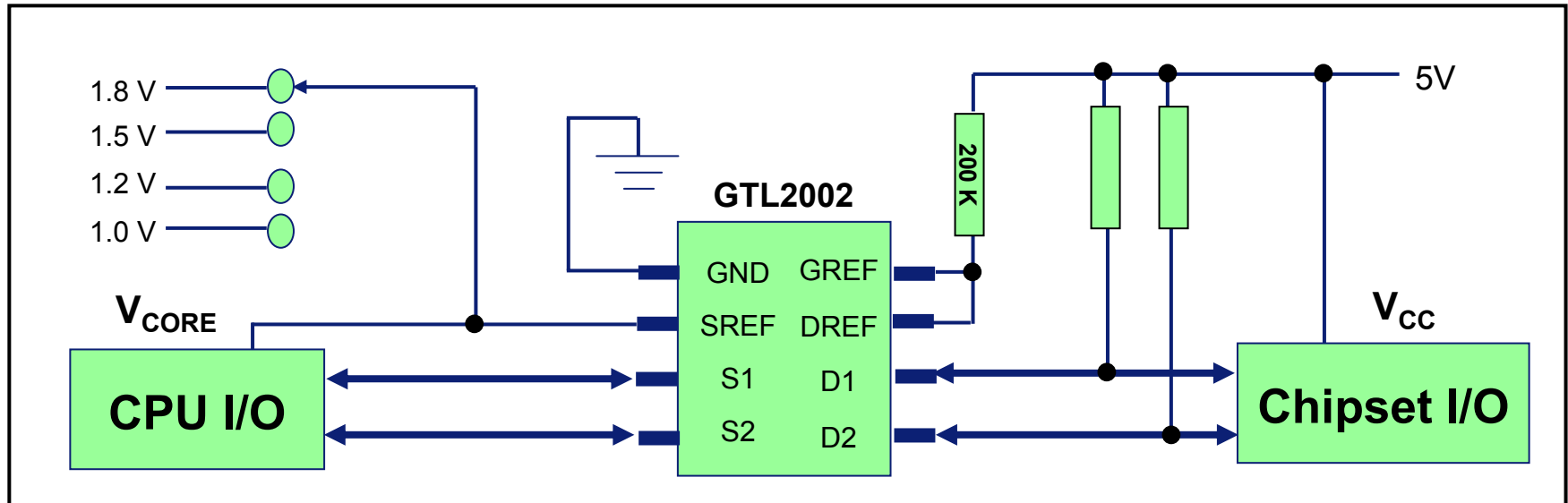


Application Note AN97055

I²C-bus Voltage Level Shifting

Solution #3 Use the GTL20xx

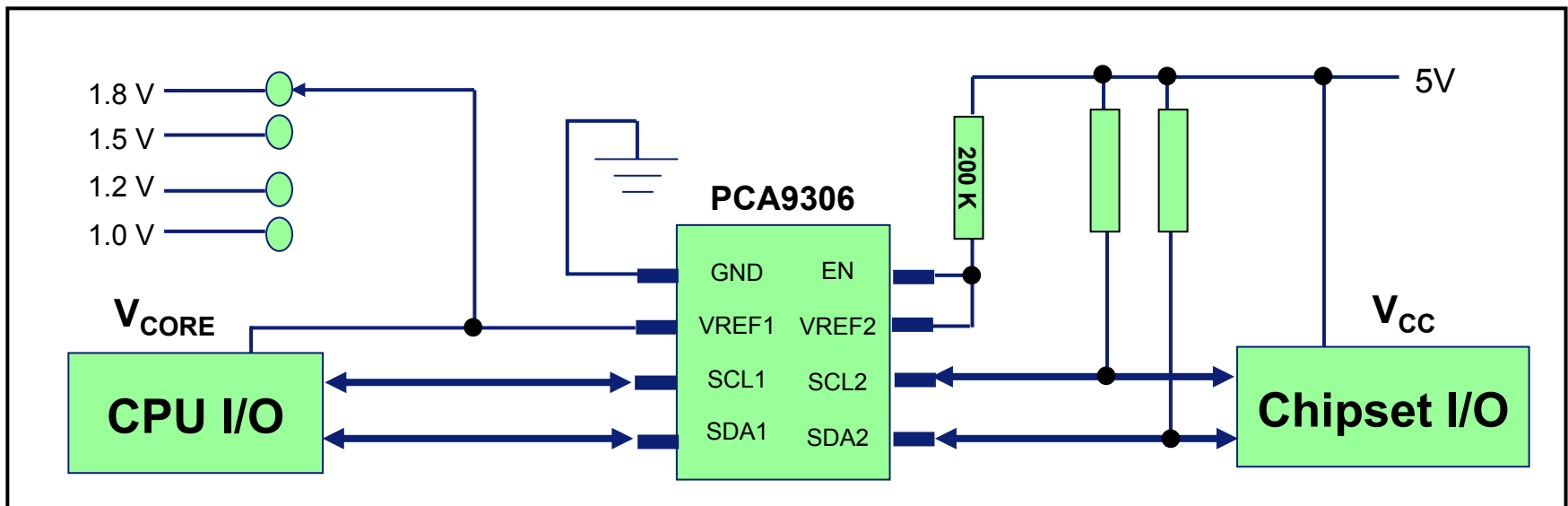
Application Note AN10145



- Voltage translation between any voltage from 1.0V to 5.0V
- Bi-directional with no direction pin
- Reference voltage clamps the input voltage with low propagation delay
- Supported packages: VSSOP8, TSSOP8, SO8

I²C-bus Voltage Level Shifting

Solution #4 Use the PCA9306

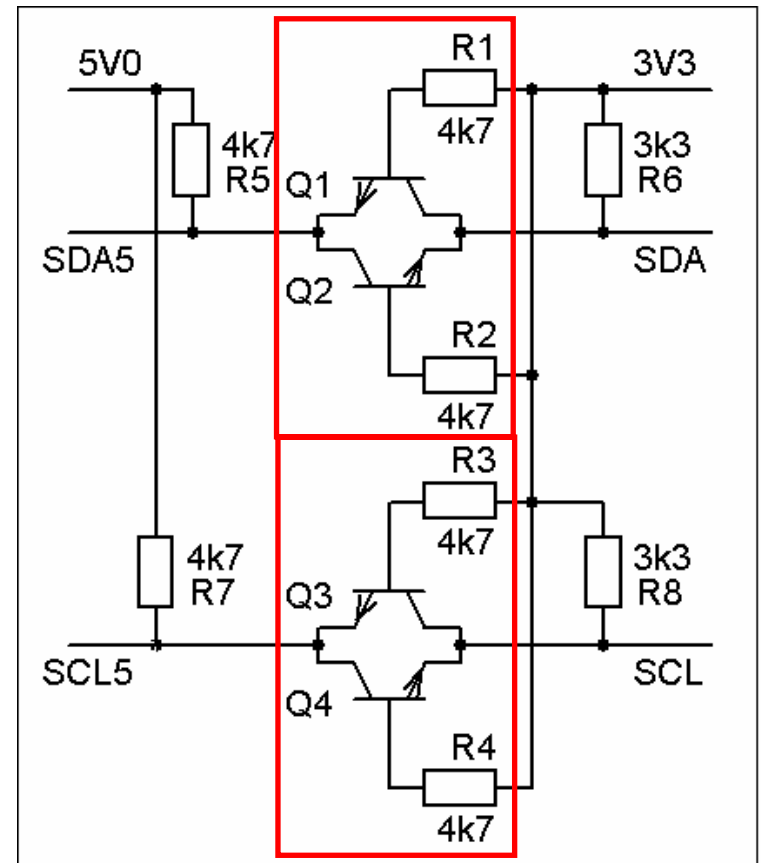


- Voltage translation between any voltage from 1.0V to 5.0V
- Bi-directional with no direction pin
- Reference voltage clamps the input voltage with low propagation delay
- Supported packages: VSSOP8, XQFN8, TSSOP8, SO8

I²C-bus Voltage Level Shifting

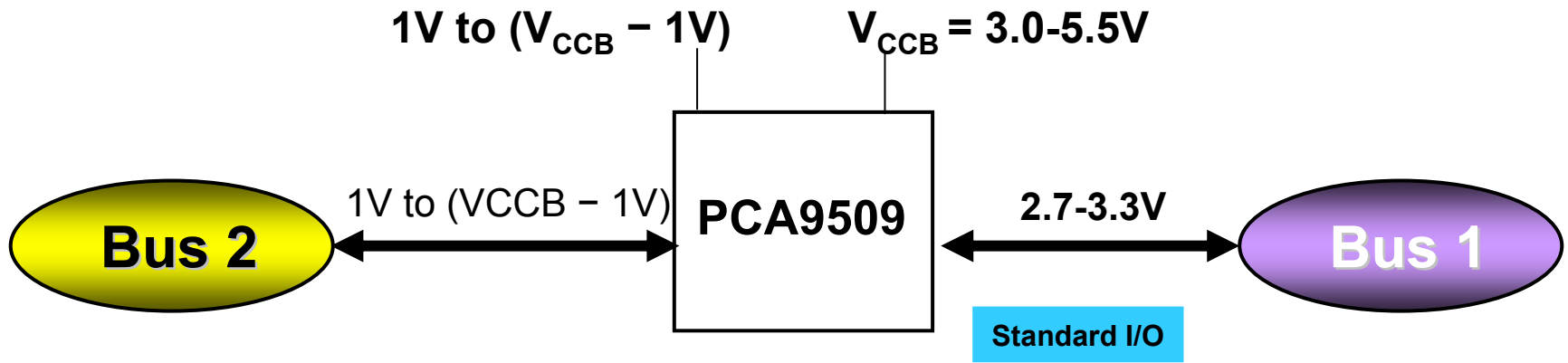
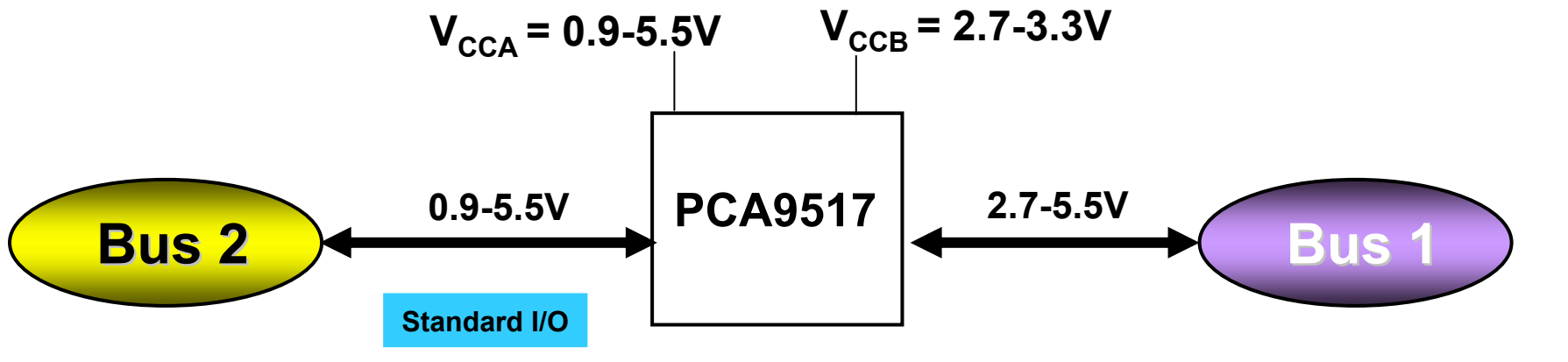
Solution #5: Low cost I²C bus level shifter just needs two double RETs

- ▶ Saves about 25% cost compared to MOSFET solutions
- ▶ Available in very small SOT363 (SC-88) and ultra small SOT666 (PUMH- and PEMH-series)



I²C-bus Voltage Level Shifting

Solution #6 PCA9517/PCA9509 Translating Buffer



I²C-bus Voltage Level Shifting

Solution #7 PCA9512A Hot Swap Buffer

- In an application where the I²C-bus is active and/or power supply is on, plug in of new devices may corrupt data or damage devices

An I²C hot swap bus buffer can be used to detect bus idle condition isolate capacitance, and prevent glitching SDA & SCL when inserting new cards into an active backplane

- Repeaters work with the same logic level on each side except the PCA9512A which works with 3.3 V and 5 V logic voltage levels at the same time

Device	# of repeaters	# of Enables
PCA9510A/11A/13A/14A	1	1
PCA9512A	1	0

Isolate I²C hanging segment(s)

