I^{2}C Interface Technical Deep Dive

March 2018

TI Training - summary

I2C Summary:

This training will focus on the I2C protocol and the challenges our customer designer's face when using this standard. Topics covered will include basic I2C protocol, challenges/tradeoffs of I2C, and how TI products help overcome these problems.

What you'll learn:

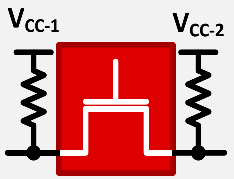
- Learn I2C Protocol
- Understand I2C challenges
- Provide solutions to problems

Detailed Agenda

- Overview of I2C
  - Hardware
  - Protocol
- Devices
  - Translators
  - Switches
  - Buffers
  - I/O Expanders

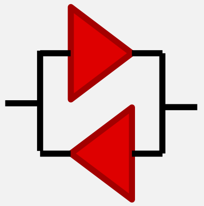
I^{2}C interface products

Translators



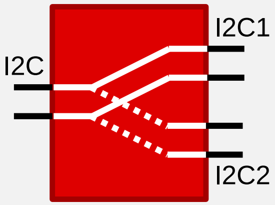
- Level-shifter
- Bus Isolation

Repeaters



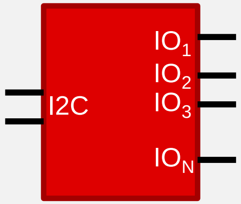
- Static offset buffer
- Hot-swappable buffer
- Bus extender
- Level-shifter

Switches



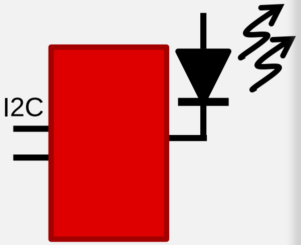
- 1:2, 1:4, 1:8
- Level-shifting switches
- Interrupt switching

IO expanders



- 4-, 8-, 16-, 24-bit
- Level-shifting expanders
- Open drain, push-pull IOs

Special function



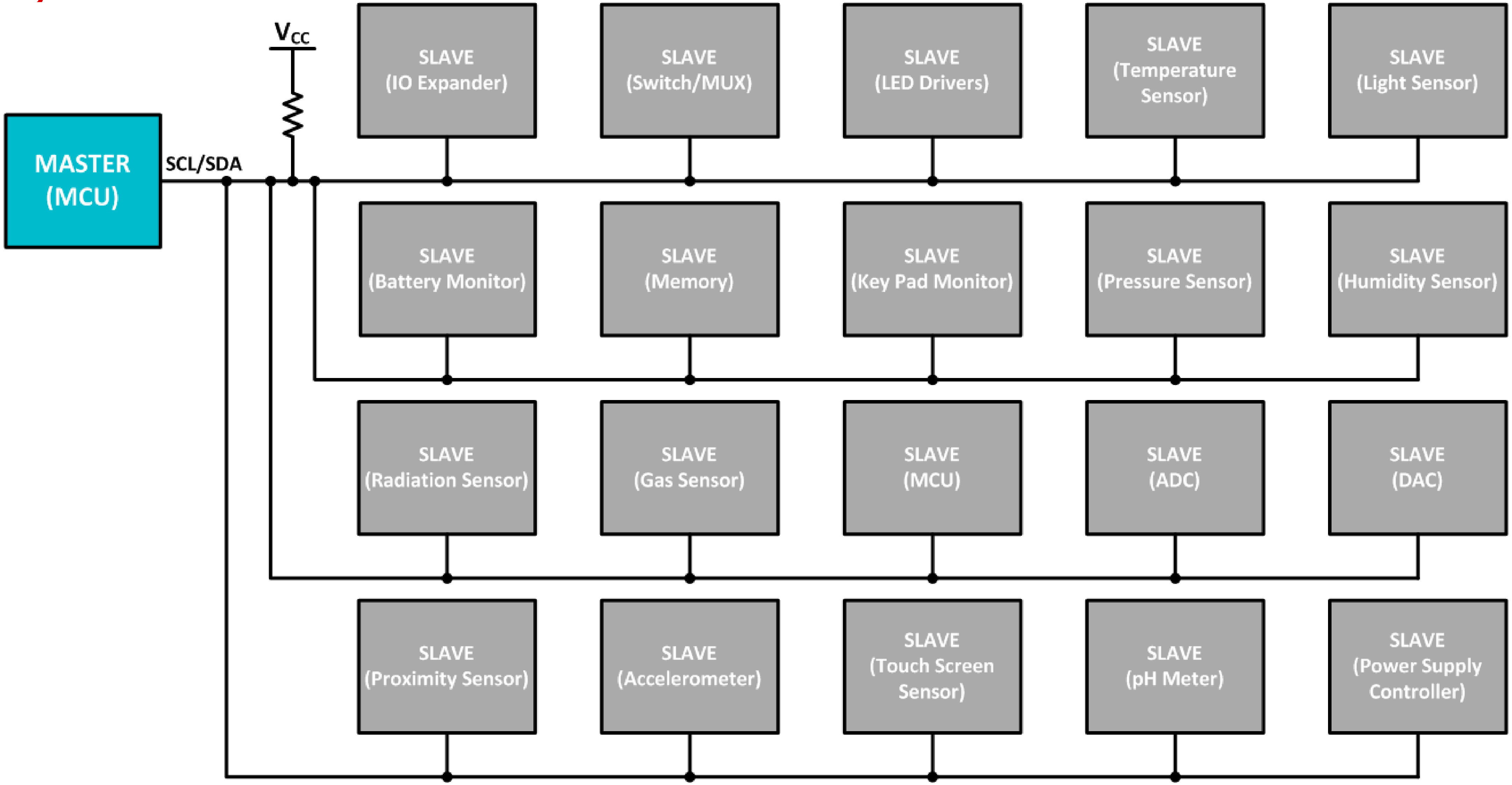
- LED driver

I2C Overview

Hardware

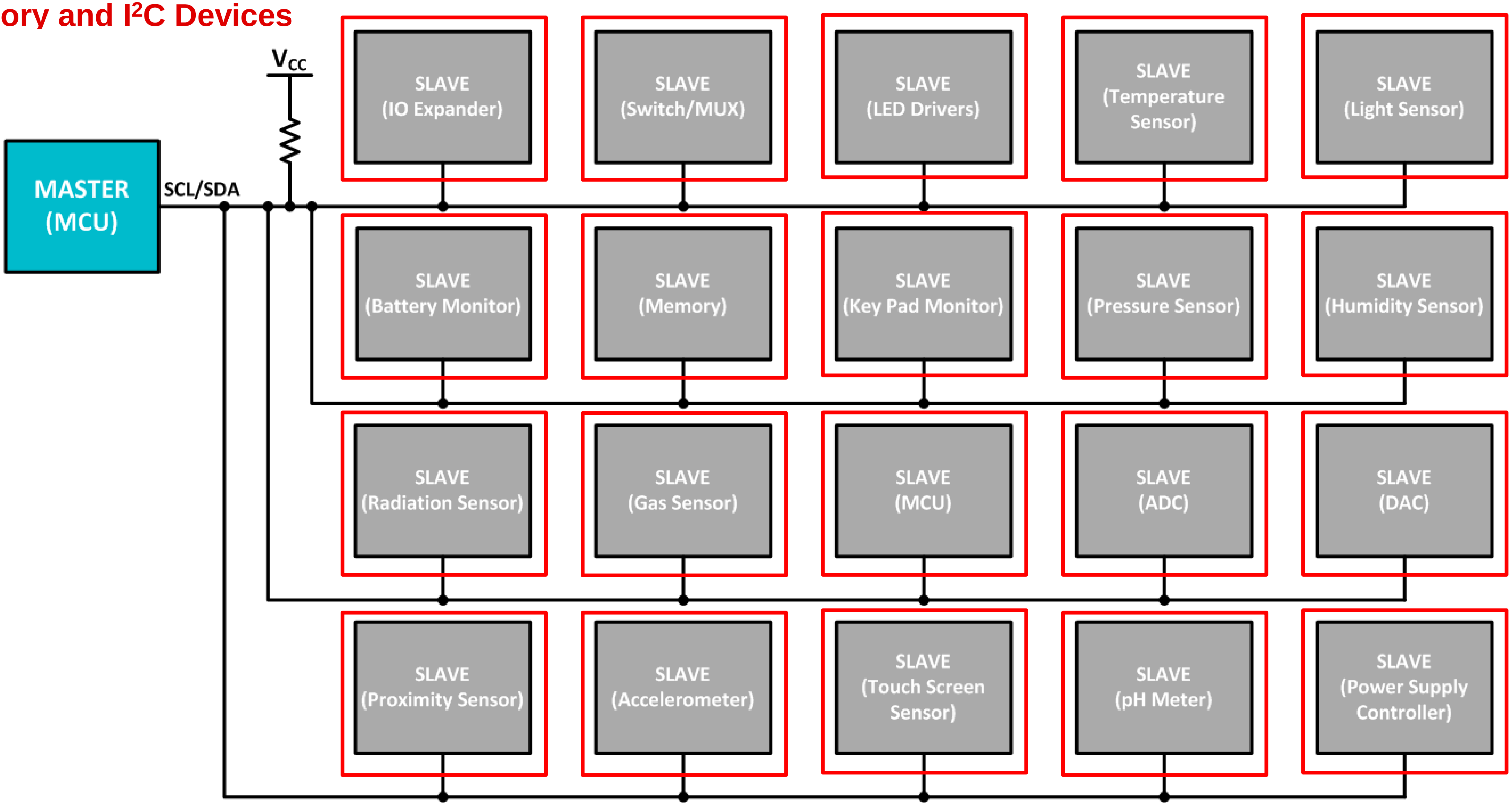
I^2C History and Overview:

History and I^2C Devices



I^{2}C History and Overview:

History and I^{2}C Devices



Features of I^{2}C Interface

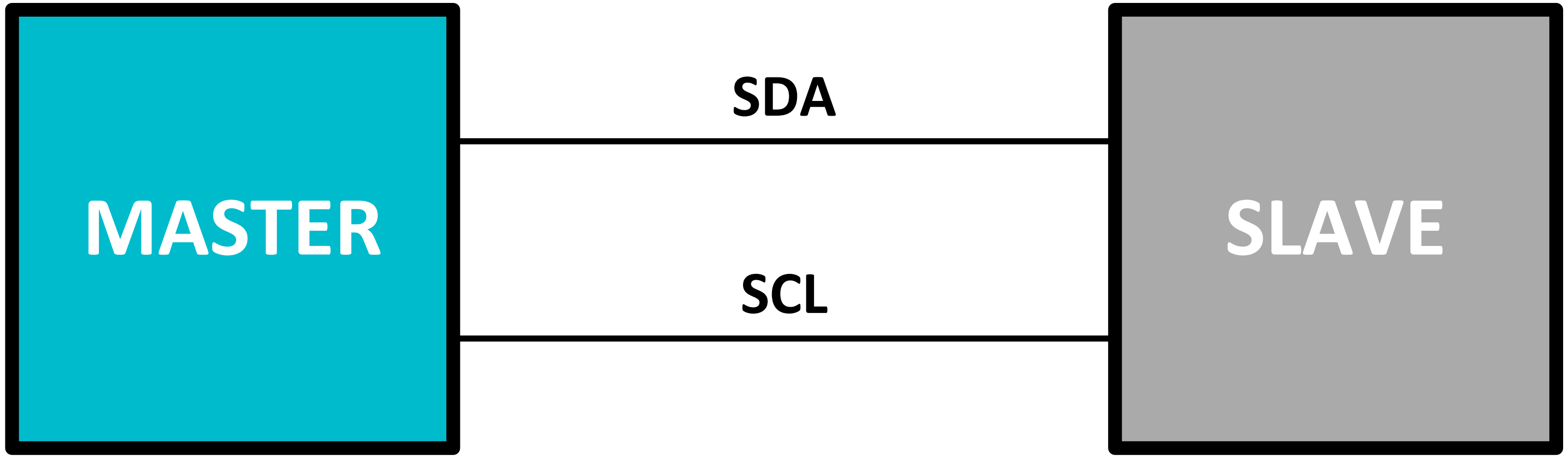
MASTER

SLAVE

Features of I²C Interface

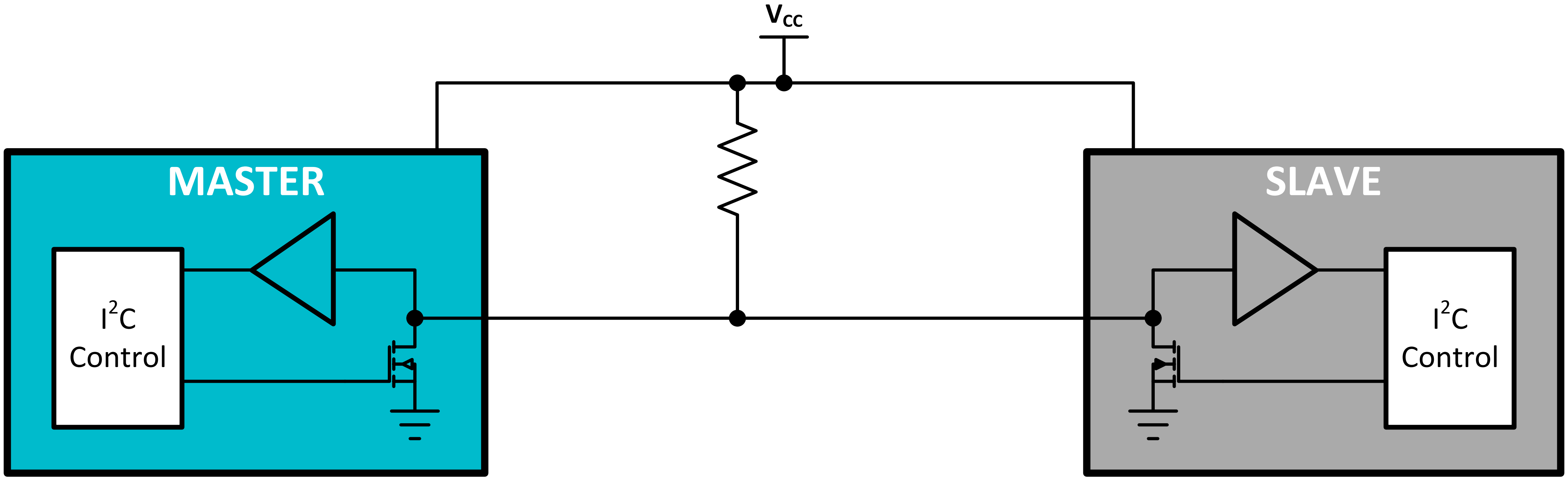
2 wire bus

- SDA (serial data line)
- SCL (serial clock line)



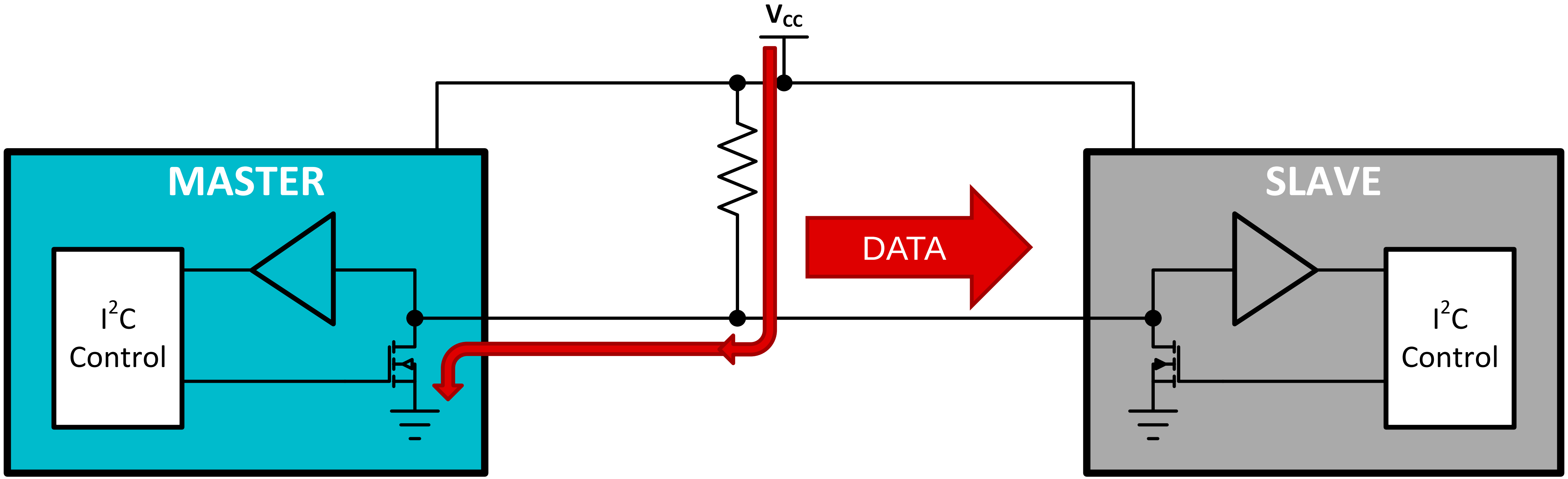
Features of I^{2}C Interface

Open Drain or Collector Driver with Input Buffer that supports Bidirectional Data



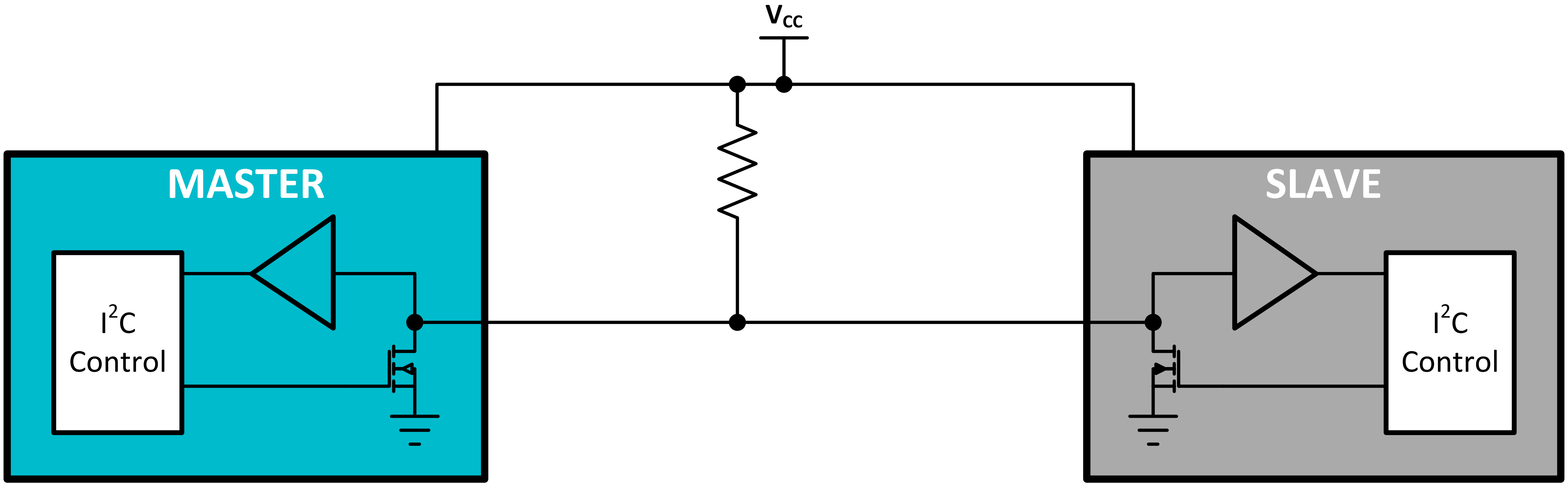
Features of I²C Interface

Open Drain/Collector Driver with Input Buffer that supports Bidirectional Data



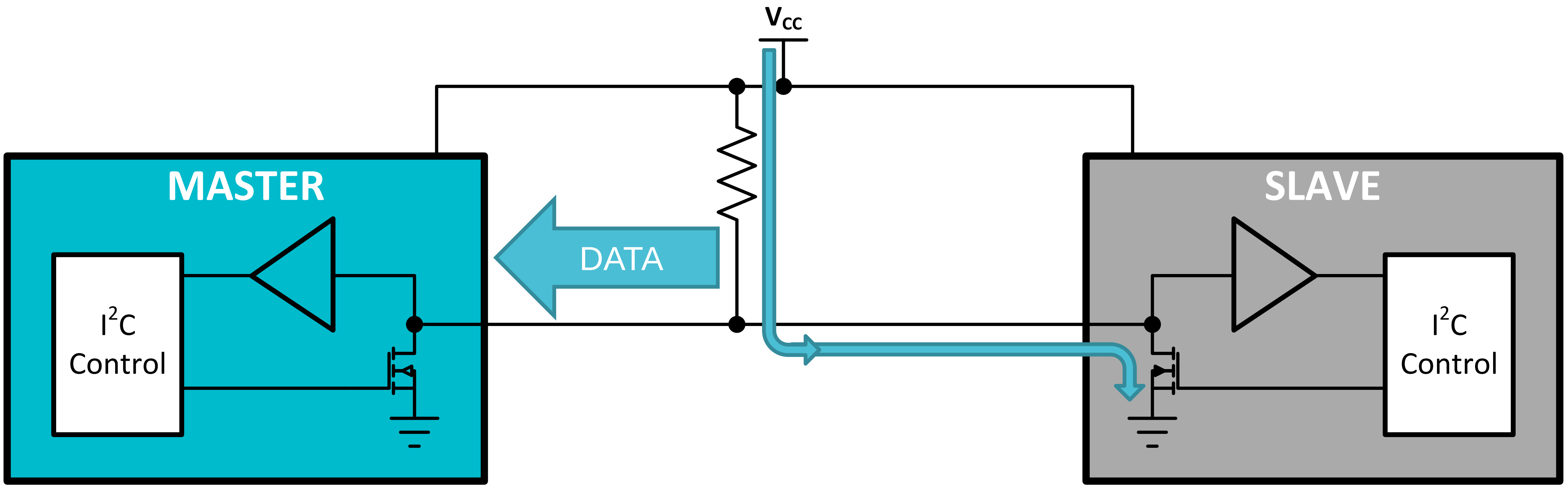
Features of I^{2}C Interface

Open Drain/Collector Driver with Input Buffer that supports Bidirectional Data

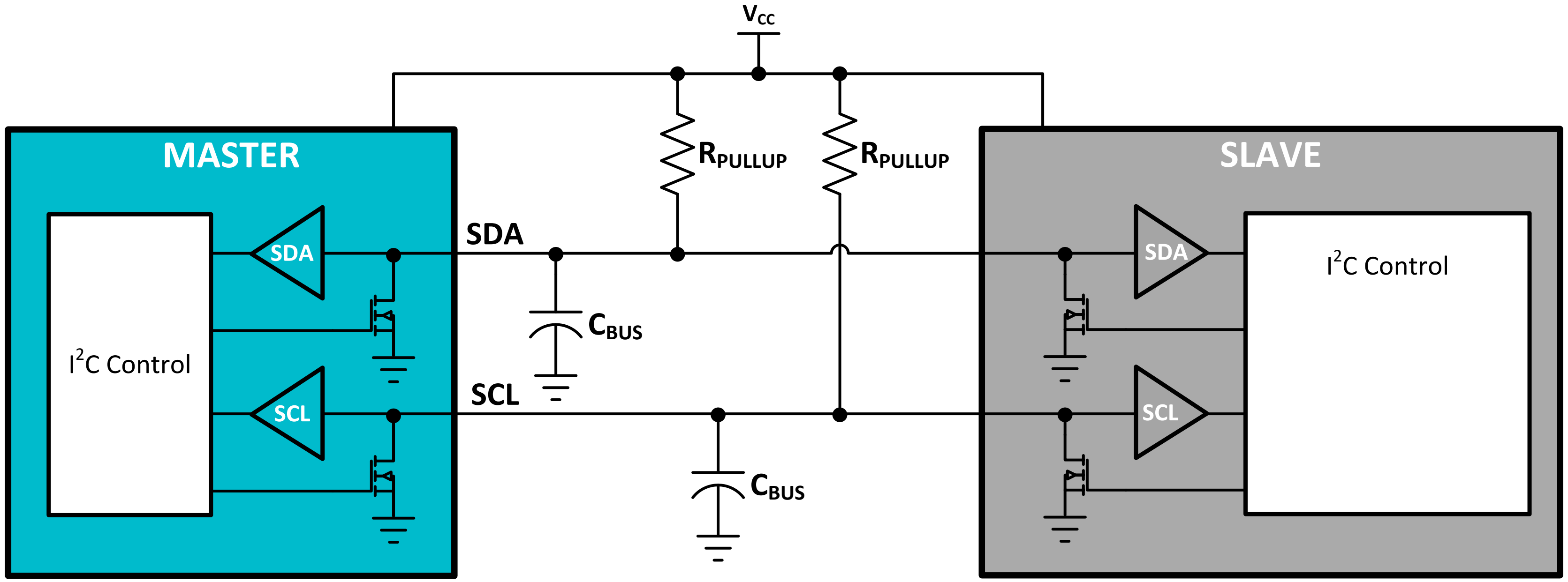


Features of I²C Interface

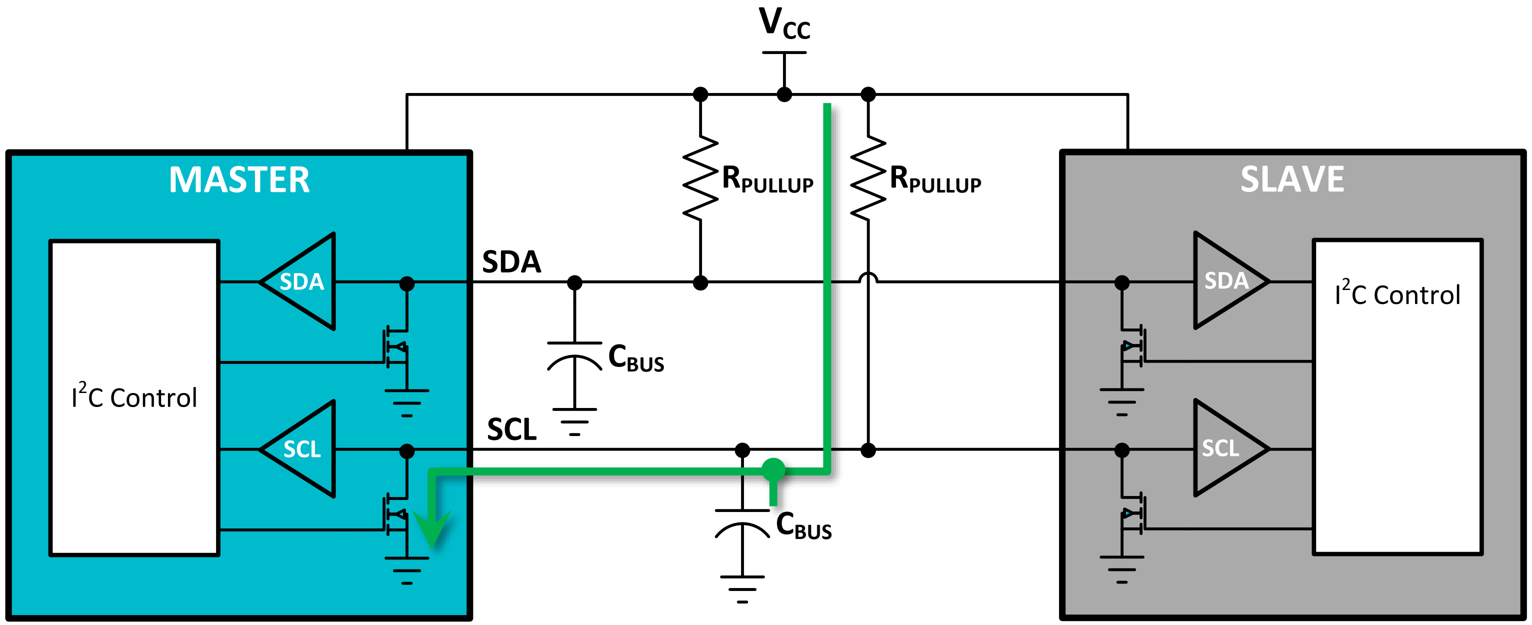
Open Drain/Collector Driver with Input Buffer that supports Bidirectional Data

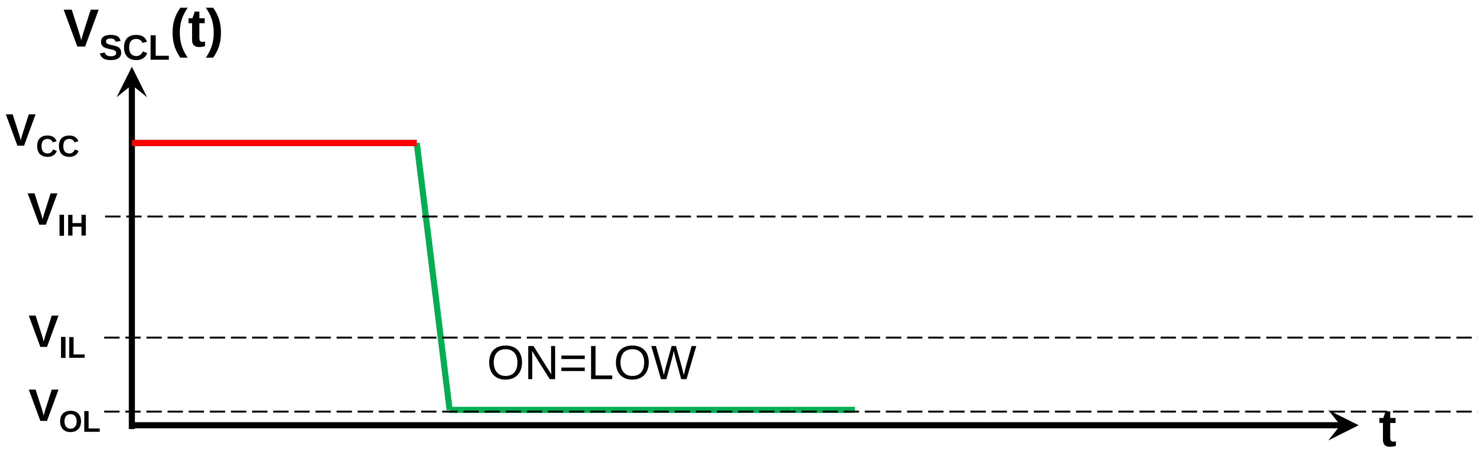


I^{2}C Interface: General Operations

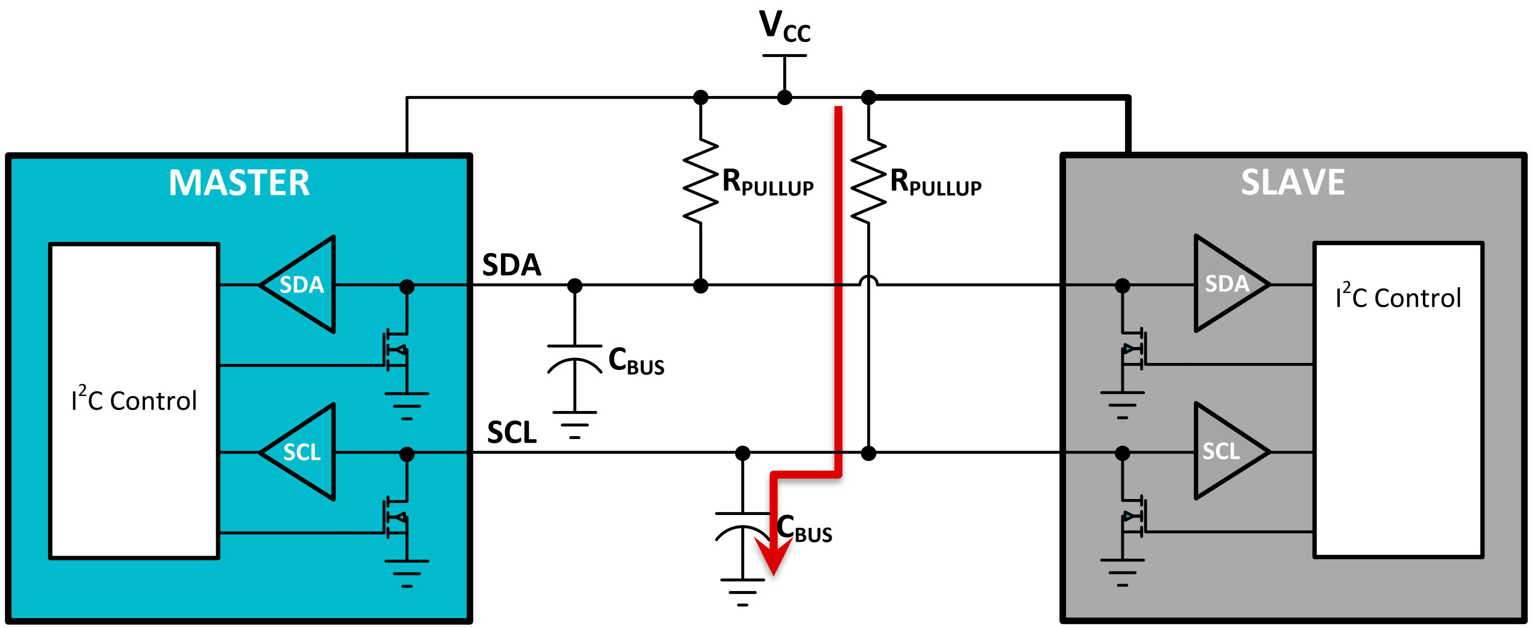


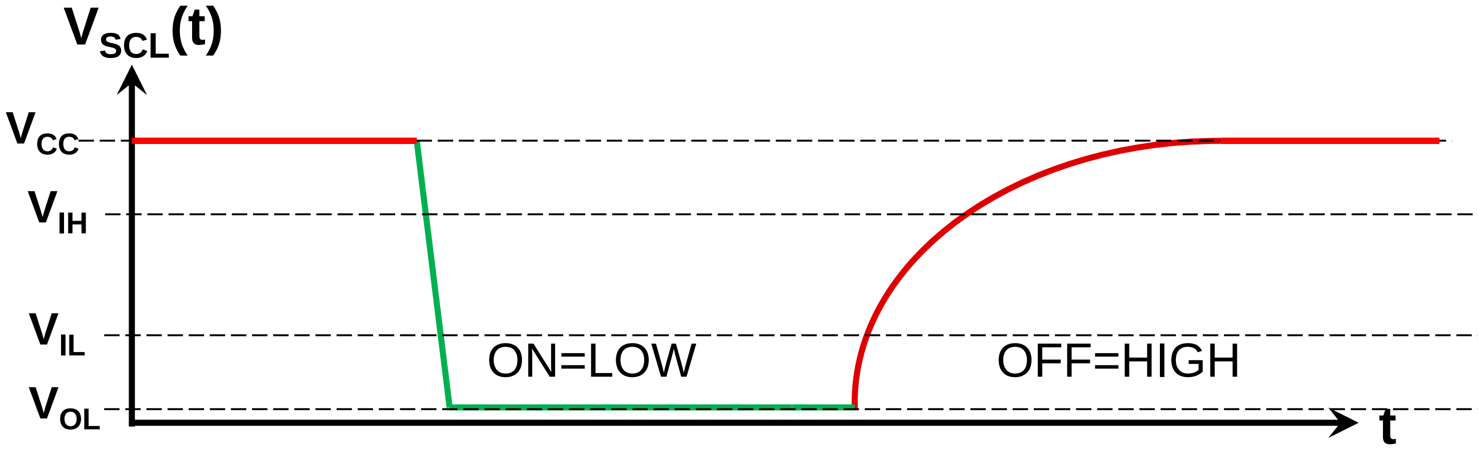
I^{2}C Physical Layer: Hardware



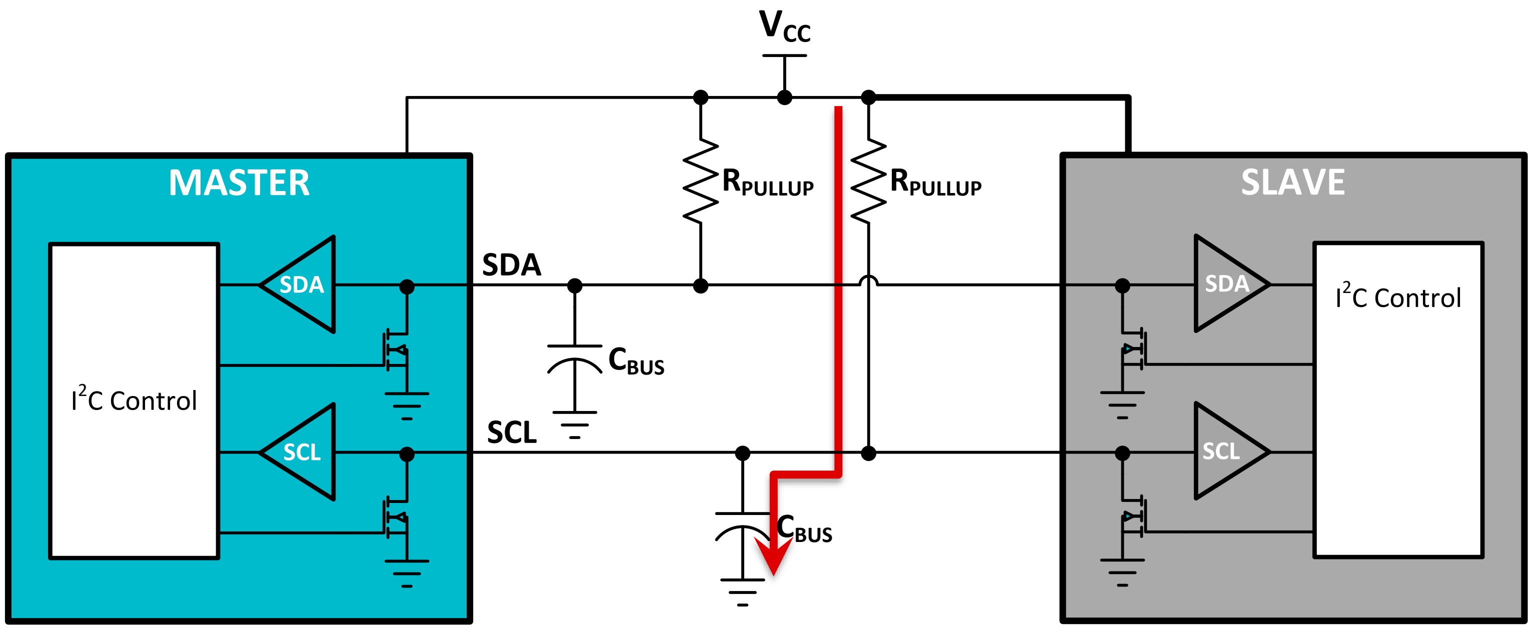


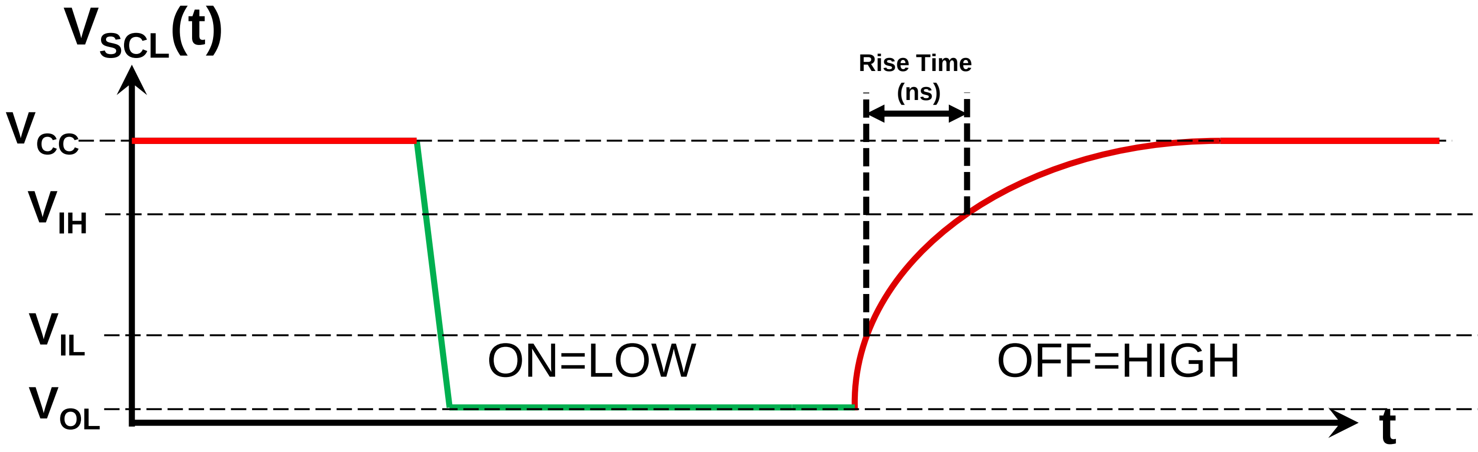
I^2C Physical Layer: Hardware





I^2C Physical Layer: Hardware



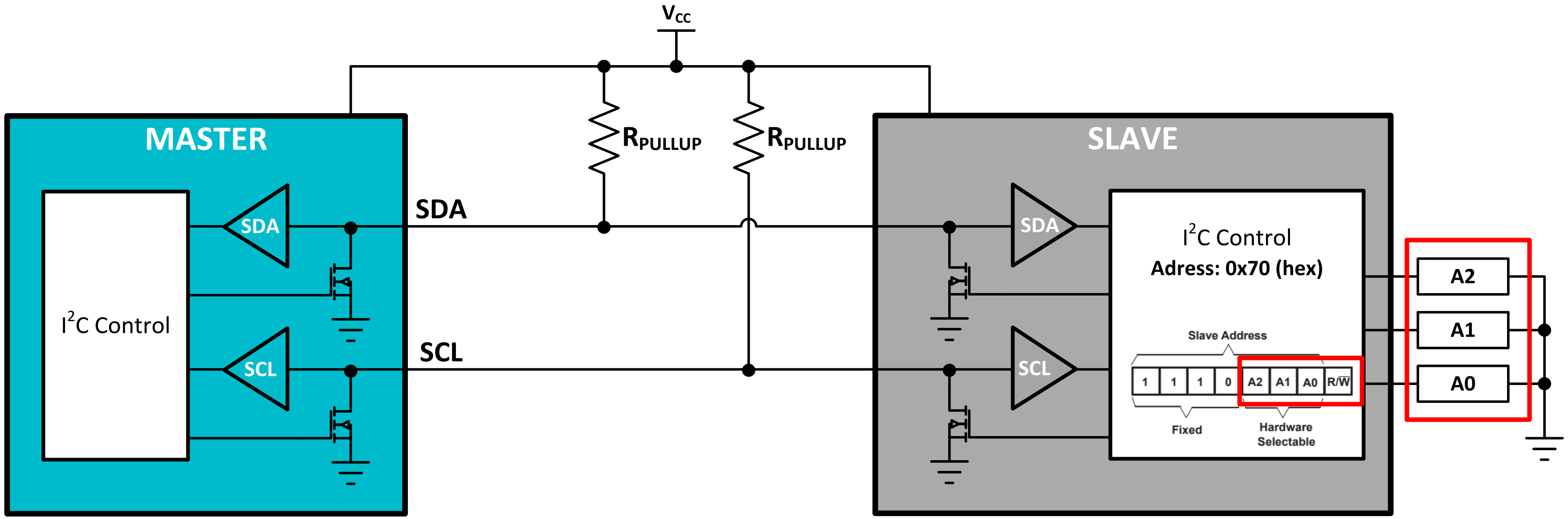


Features of I²C Interface

Addressing is accomplished with the SLAVE's hardware address.

Two Possible addressing modes:

- **7-bit address**
- **10-bit address**



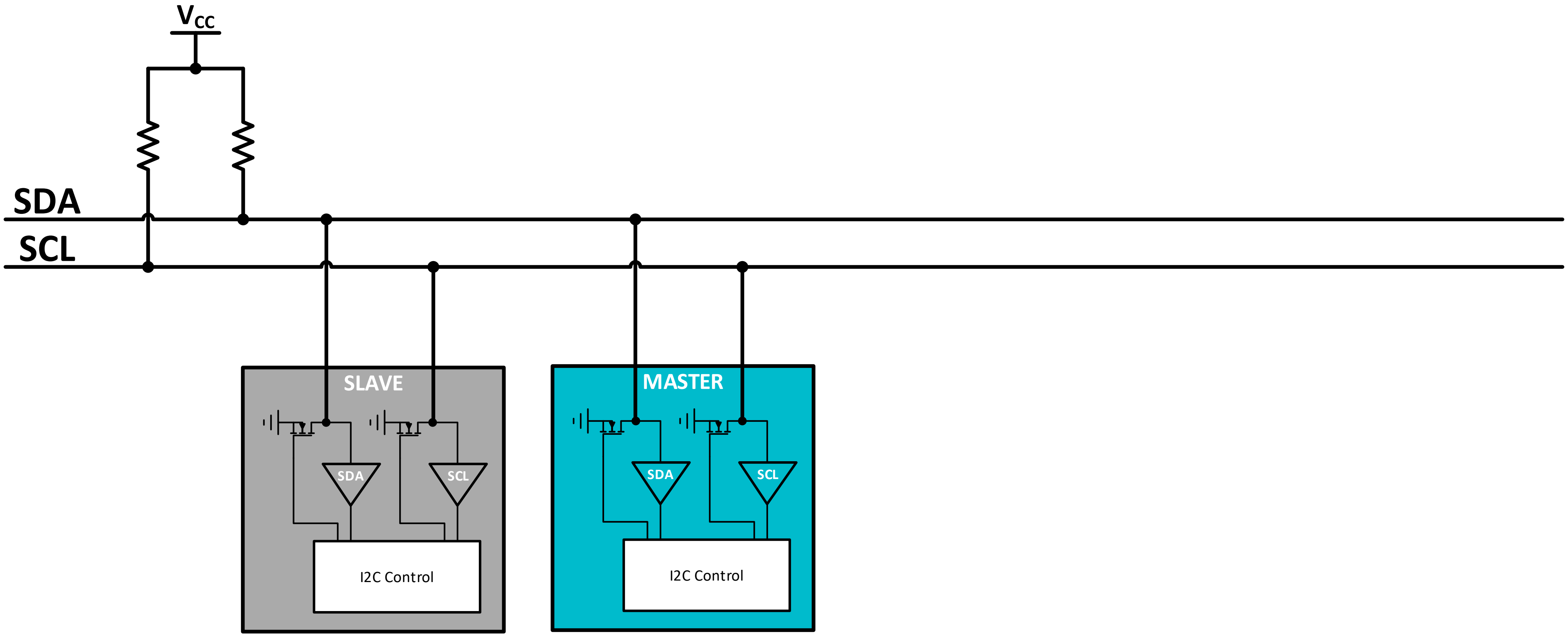
Features of I^2C Interface

Standard Mode	Fast Mode	Fast Mode Plus
0 to 100 kHz	0 to 400 kHz	0 to 1,000 kHz
$C_{BUS\ MAX}$ = 400 pF	$C_{BUS\ MAX}$ = 400 pF	$C_{BUS\ MAX}$ = 550 pF
$t_{RISE\ MAX}$ = 1,000 ns	$t_{RISE\ MAX}$ = 300 ns	$t_{RISE\ MAX}$ = 120 ns

Features of I^{2}C Interface

The I^{2}C bus is a very popular and powerful bus used for communications between a master (or multiple masters) and a single or multiple slave devices (aka ICs)

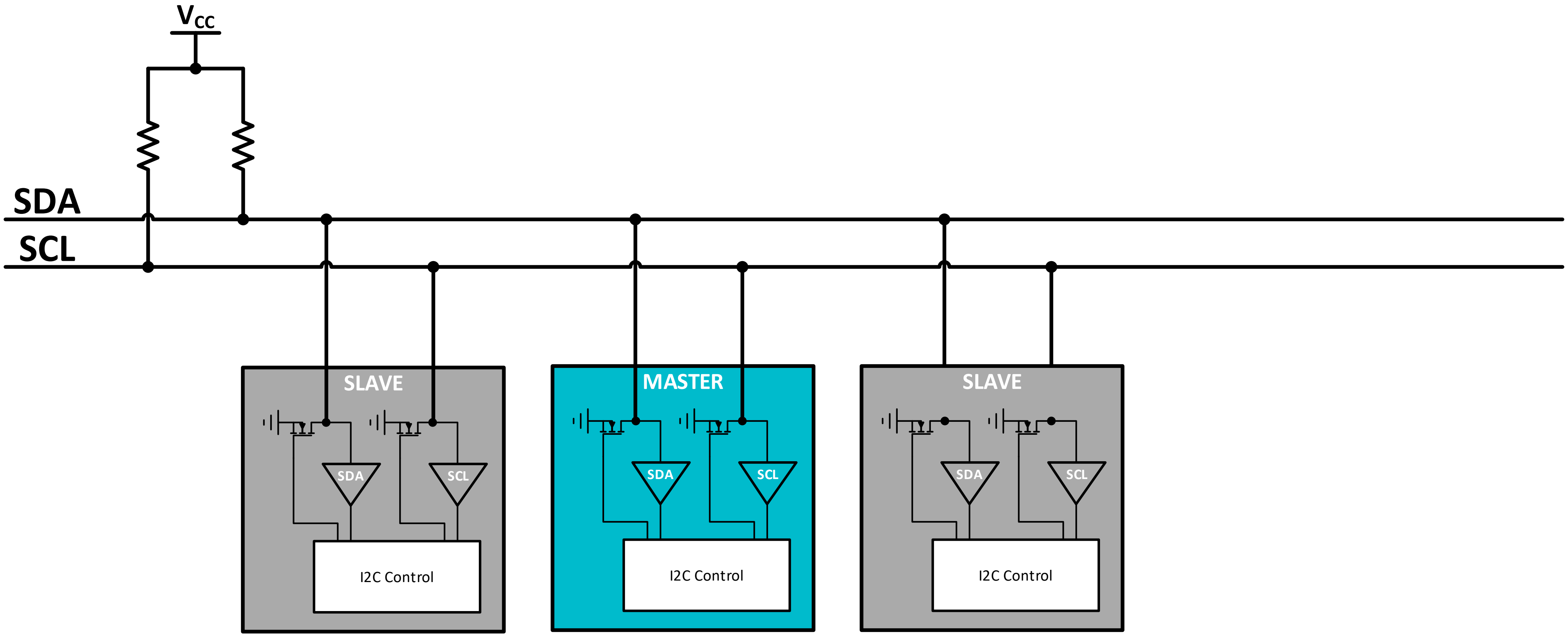
Ability to add more devices in parallel by just connecting to the bus. Ease of expansion.



Features of I²C Interface

The I²C bus is a very popular and powerful bus used for communications between a master (or multiple masters) and a single or multiple slave devices (aka ICs)

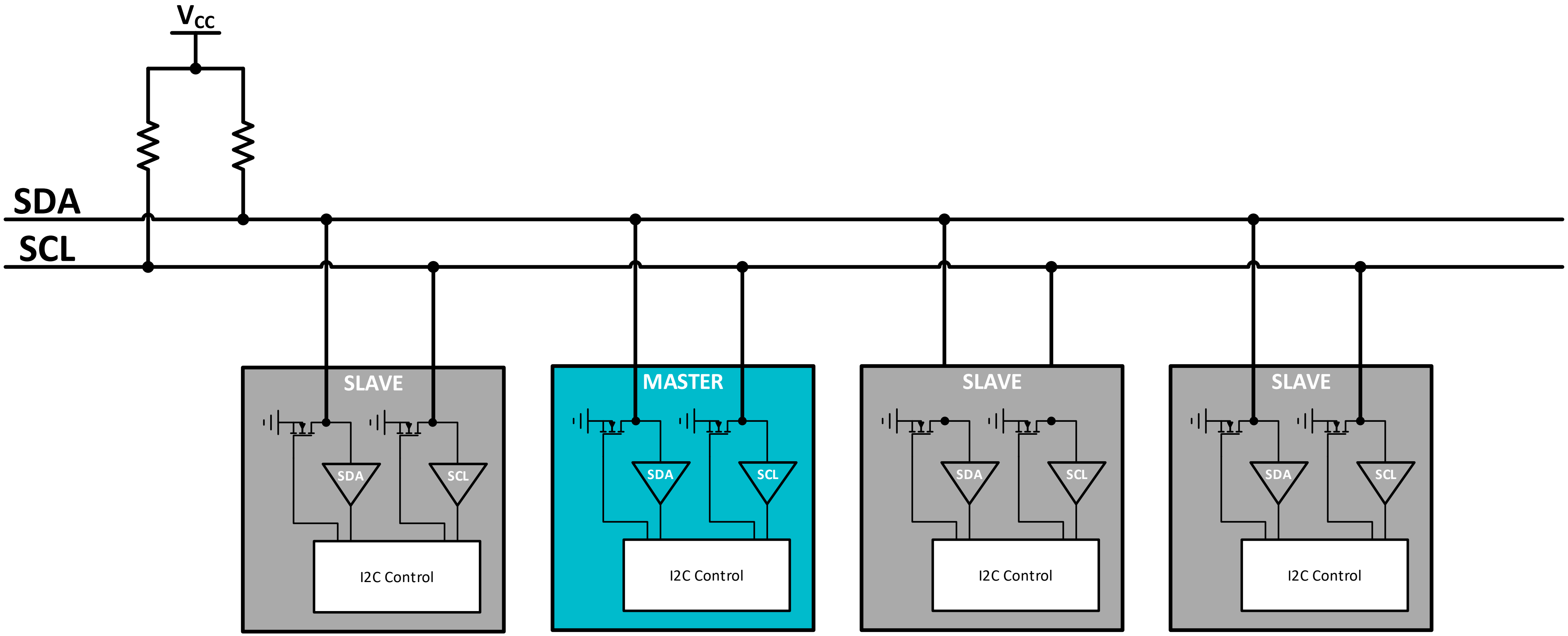
Ability to add more devices in parallel by just connecting to the bus. Ease of expansion.



Features of I²C Interface

The I²C bus is a very popular and powerful bus used for communications between a master (or multiple masters) and a single or multiple slave devices (aka ICs)

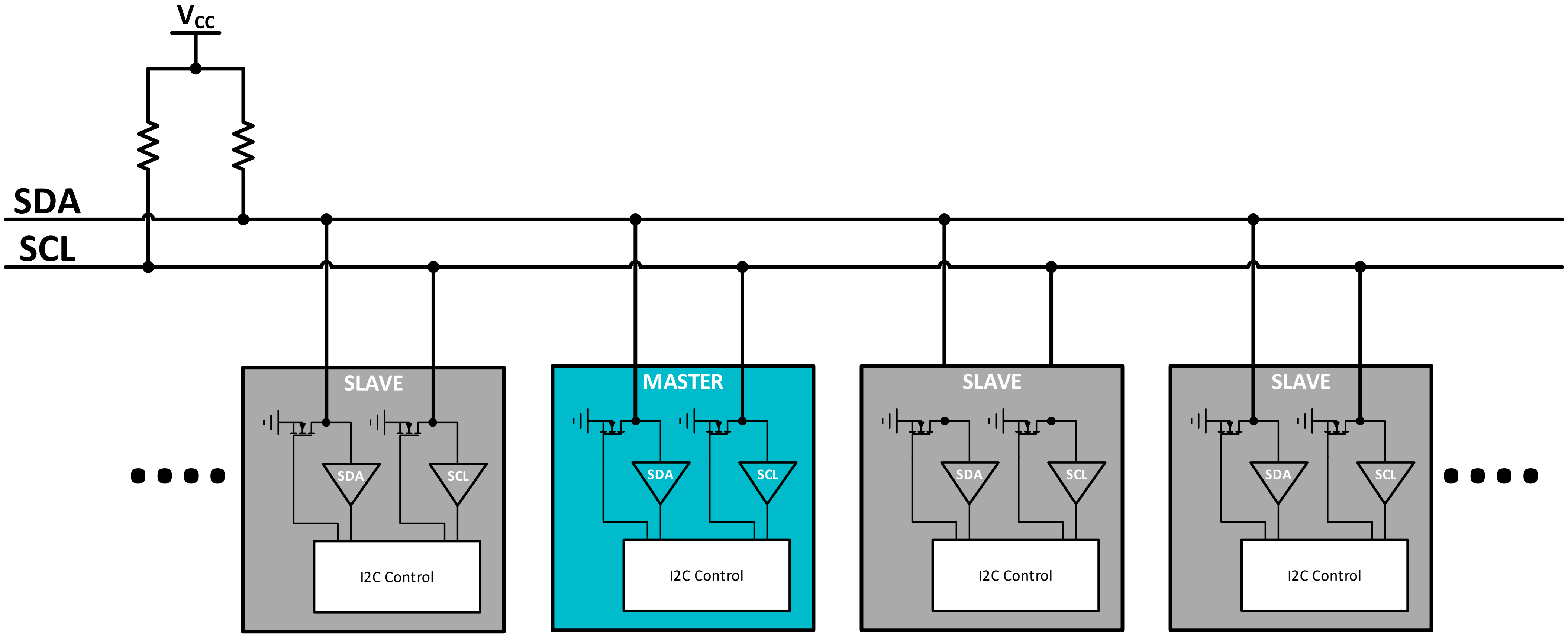
Ability to add more devices in parallel by just connecting to the bus. Ease of expansion.



Features of I²C Interface

The I²C bus is a very popular and powerful bus used for communications between a master (or multiple masters) and a single or multiple slave devices (aka ICs)

Ability to add more devices in parallel by just connecting to the bus. Ease of expansion.



Benefits and Limitations of I^{2}C

Pros

- Simple
- Low Cost
- Robust
- Standardized
- Wide assortment of peripherals
- No need for termination
- Easy Bus Expansion

Cons

- Low speed
- Bus Capacitance Limited
  - Limits Speed
  - Limits distance
- Half Duplex Only
- No suited for ~long distances

Benefits and Limitations of I^{2}C

Pros

- Simple
- Low Cost
- Robust
- Standardized
- Wide assortment of peripherals
- No need for termination
- Easy Bus Expansion

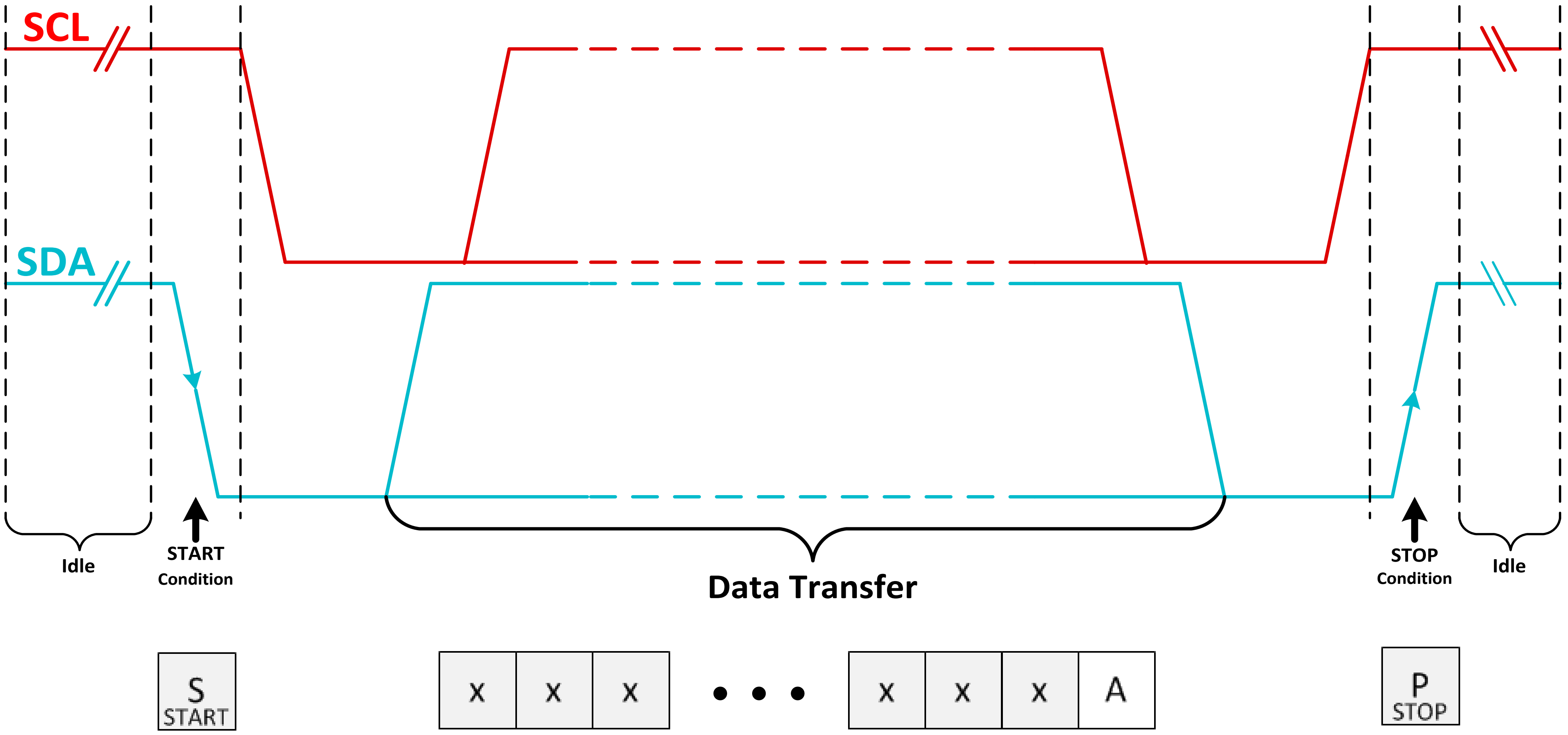
Cons

- Low speed
- Bus Capacitance Limited
    - Limits Speed
    - Limits distance
- Half Duplex Only
- No suited for ~long distances

I2C Overview

Protocol

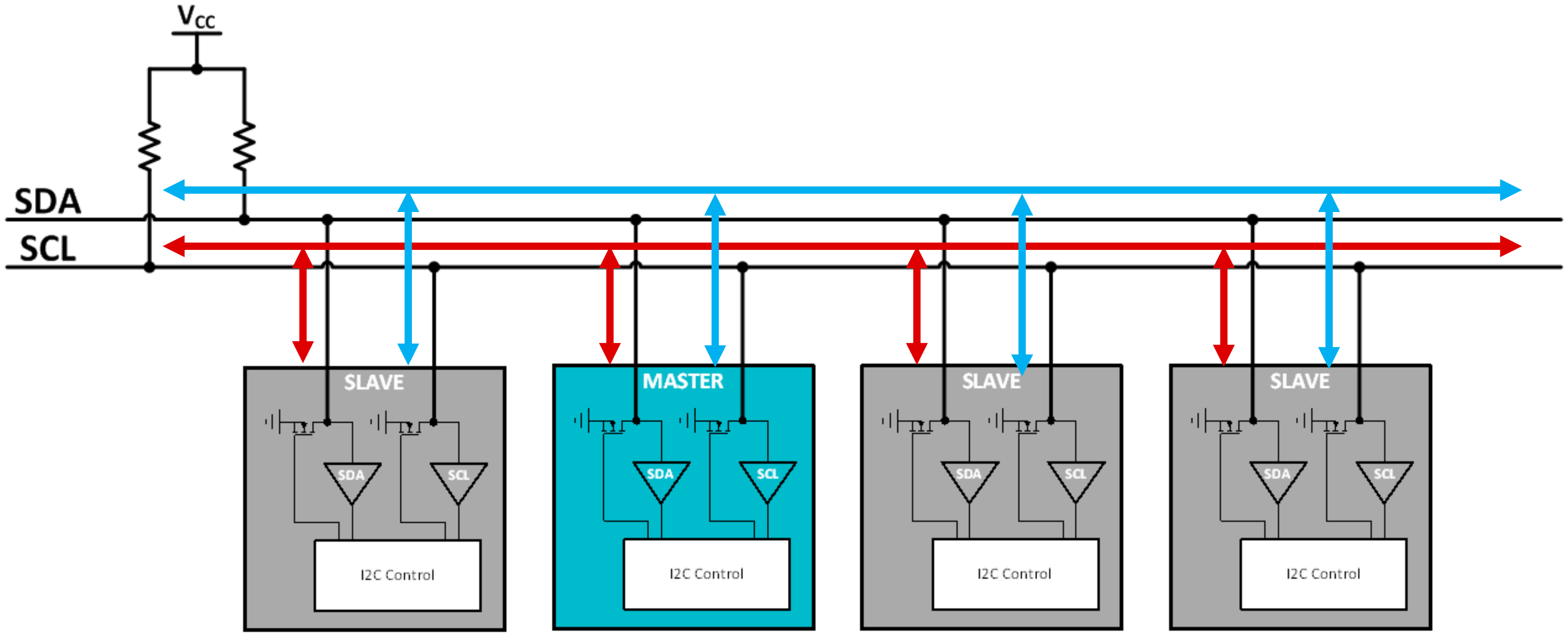
I^{2}C Interface: Bus Control: START and STOP Conditions



A high-to-low transition on the SDA line while the SCL is high defines a START condition.

A low-to-high transition on the SDA line while the SCL is high defines a STOP condition.

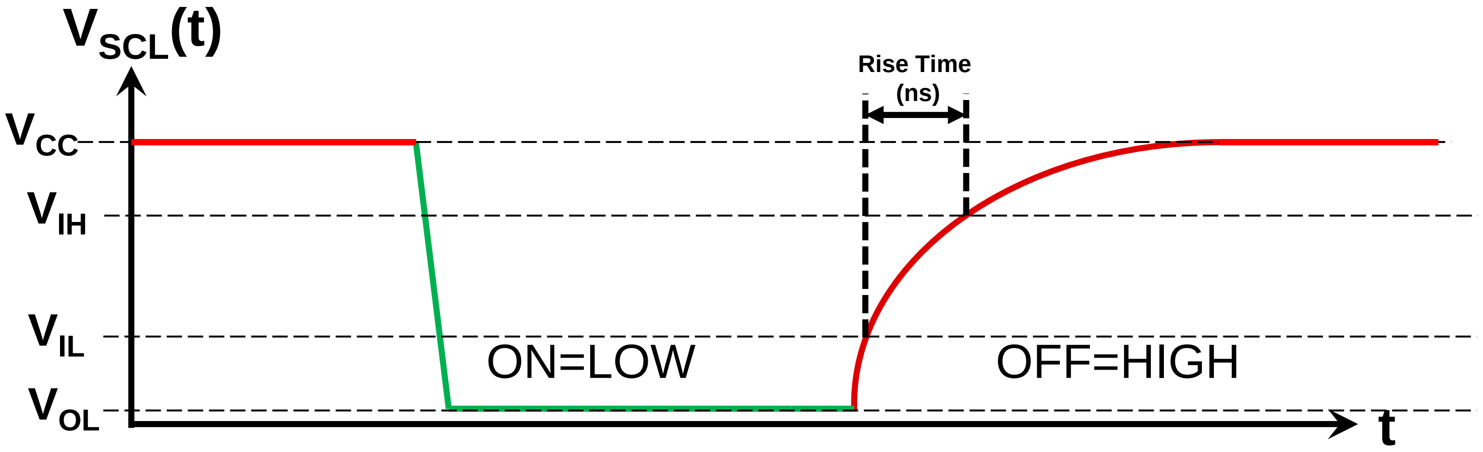
I^{2}C Interface: Signals (SDA and SCL)



I²C Interface: Logic Levels and Timing

The I2C standard specifies that the V_{IH} and V_{IL} be 70% and 30% of VCC respectively.

The Logic level

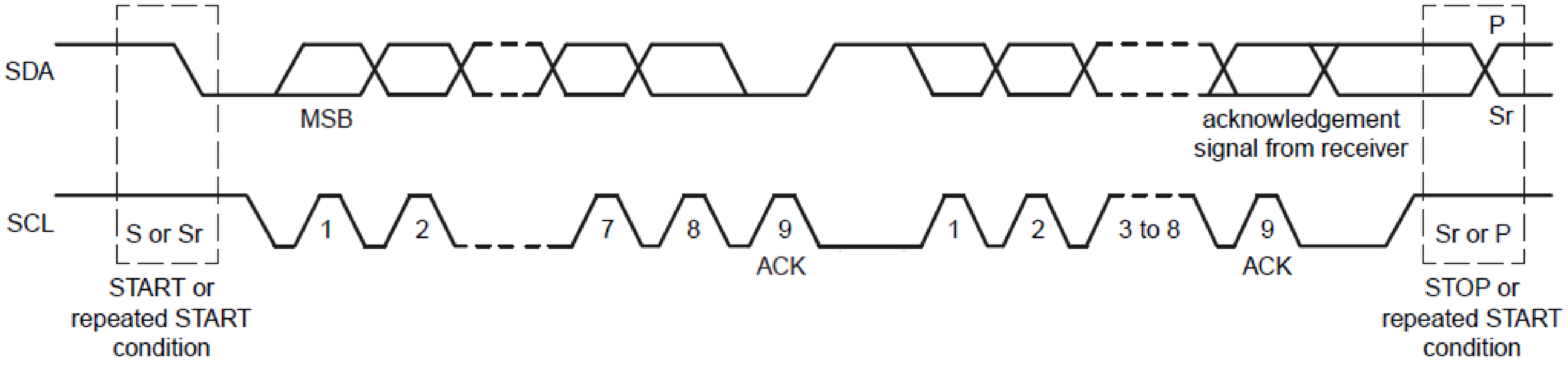


I^{2}C Interface: Byte Format

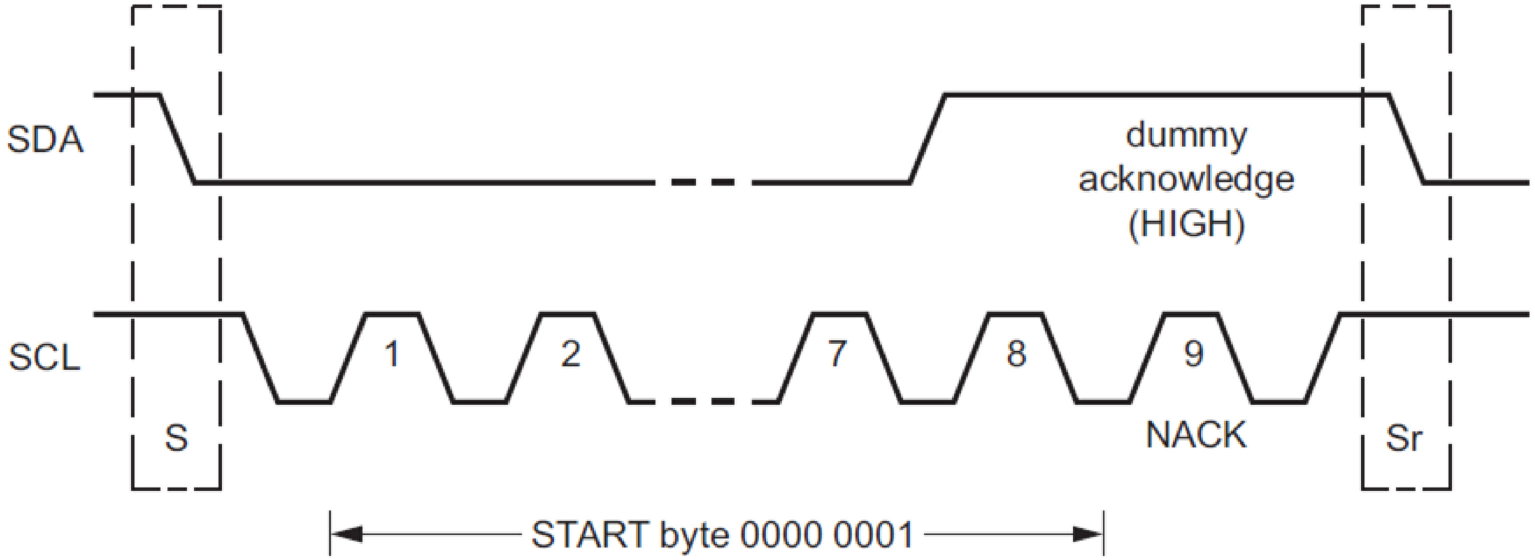
8-bit (byte) format

MSB	-	-	-	-	-	-	LSB	A

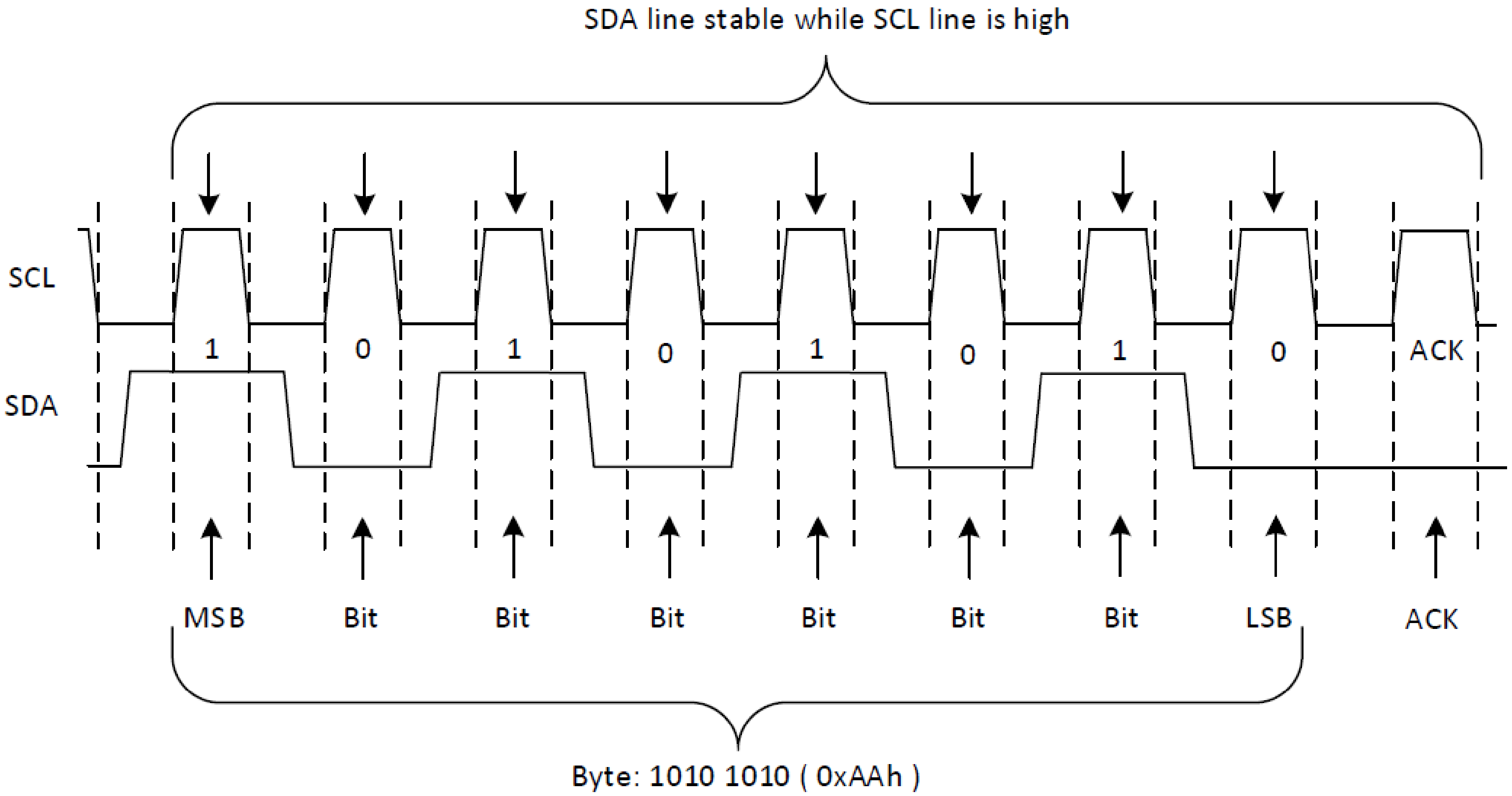
A=ACK
Acknowledge



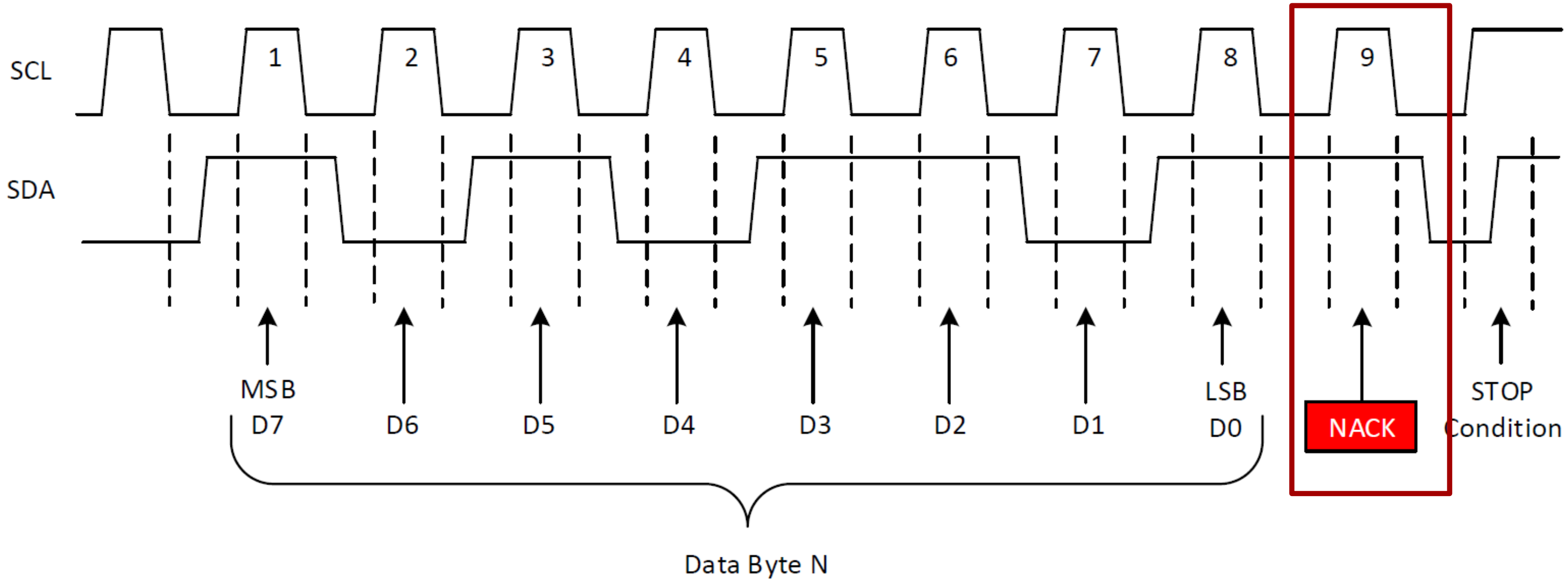
I^{2}C Interface: Repeated START Condition



I^{2}C Interface: Data Validity and Byte Format


SDA line stable while SCL line is high
SCL
SDA
1
0
1
0
1
0
1
0
ACK
MSB
Bit
Bit
Bit
Bit
Bit
Bit
LSB
ACK
Byte: 1010 1010 (0xAAh)

I^{2}C Interface: Acknowledge (ACK) - Not Acknowledge (NACK)



There are several conditions that lead to the generation of a NACK:

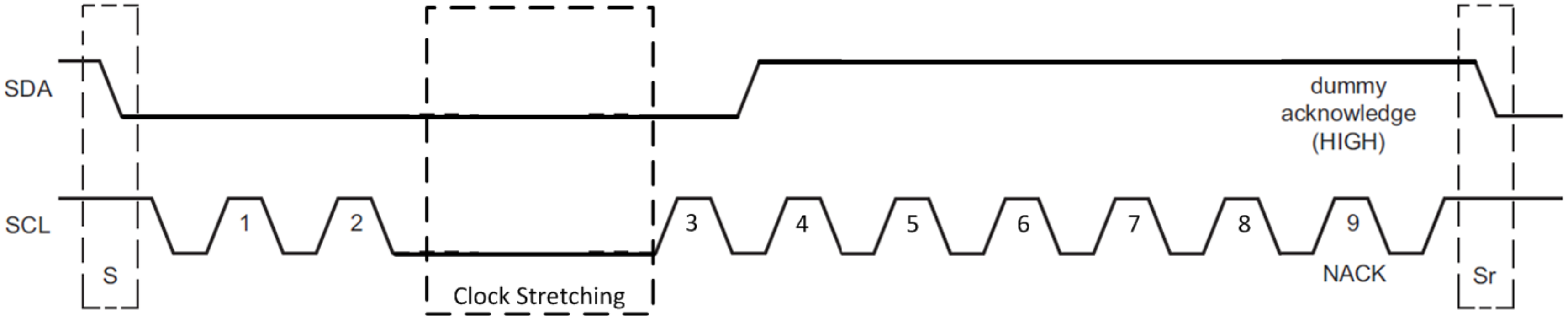
1. The receiver is unable to receive or transmit because it is performing some real-time function and is not ready to start communication with the master.

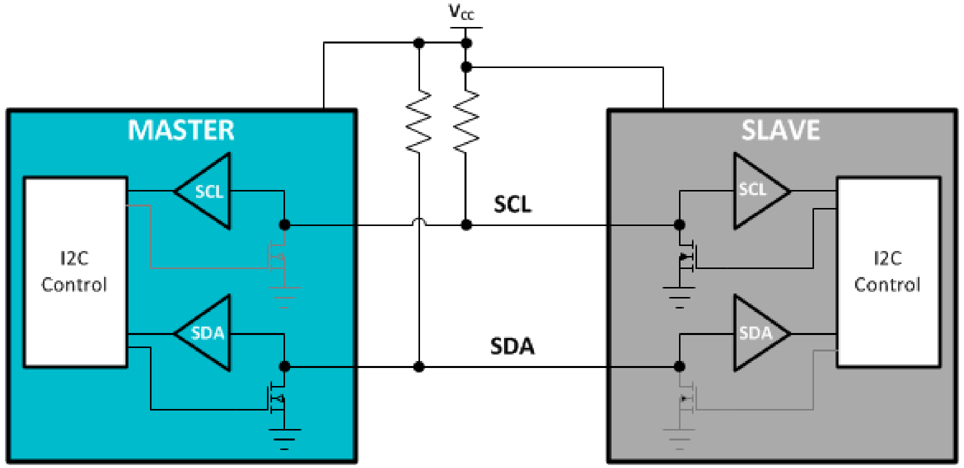
2. During the transfer, the receiver gets data or commands that it does not understand.

3. During the transfer, the receiver cannot receive any more data bytes.

4. A master-receiver is done reading data and indicates this to the slave through a NACK.

I^{2}C Interface: Clock Stretching





I^{2}C Interface: Other I2C Based Protocols

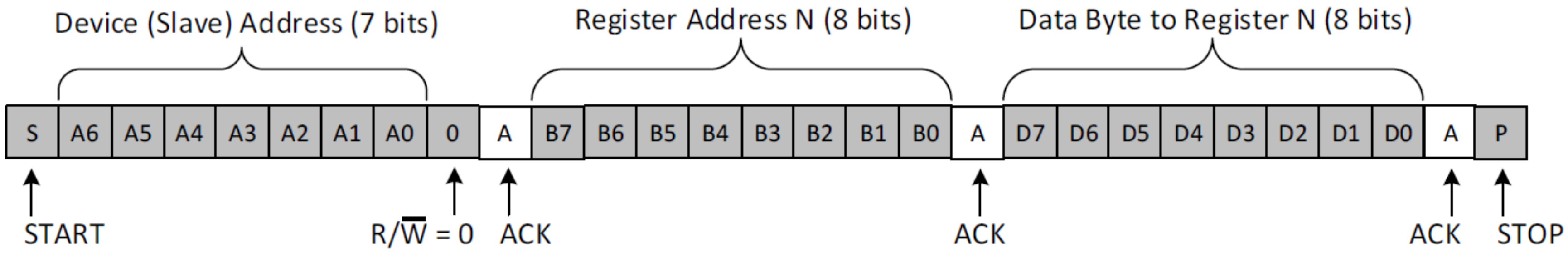
- System Management Bus (SMBUS)
- Power Management Bus (PMBUS)
- Intelligent Platform Management Interface (IPMI)
- Display Data Channel (DDC)
- Advanced Telecom Computing Architecture (ATCA)

I^{2}C Data : Writing to a Slave on the I^{2}C Bus

Example of Writing a single byte to a Slave register:

▨ Master Controls SDA Line

□ Slave Controls SDA Line



Device (Slave) Address (7 bits)

Register Address N (8 bits)

Data Byte to Register N (8 bits)

S	A6	A5	A4	A3	A2	A1	A0	0	A	B7	B6	B5	B4	B3	B2	B1	B0	A	D7	D6	D5	D4	D3	D2	D1	D0	A	P

START

$R/\overline{W} = 0$

ACK

ACK

ACK

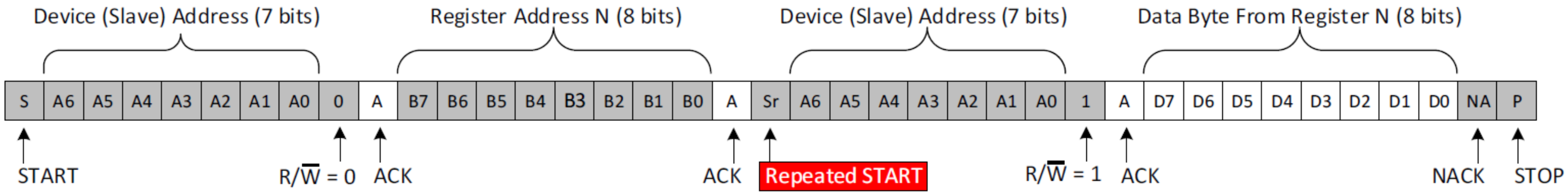
STOP

I^2C Data : Reading from a Slave on the I^2C Bus

Example of Reading a single register byte from a Slave register:

▨ Master Controls SDA Line

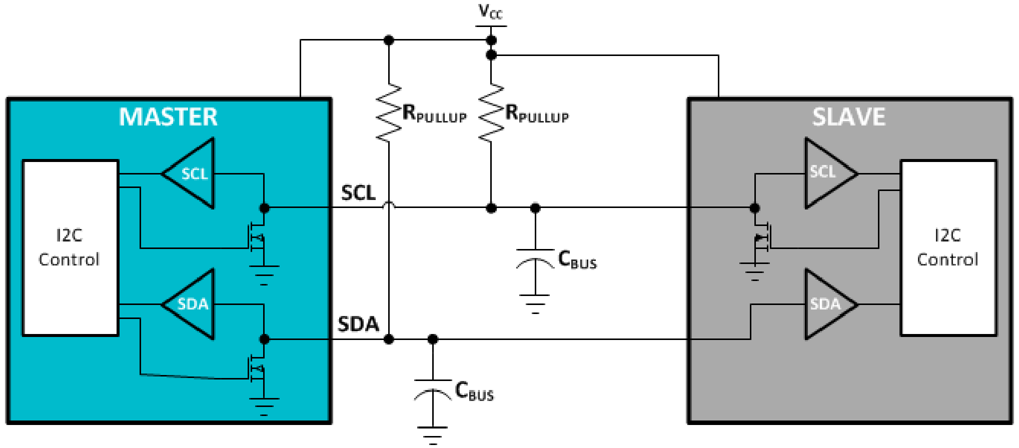
□ Slave Controls SDA Line



Device (Slave) Address (7 bits)
Register Address N (8 bits)
Device (Slave) Address (7 bits)
Data Byte From Register N (8 bits)
START
R/$\overline{W}$ = 0
ACK
ACK
Repeated START
R/$\overline{W}$ = 1
ACK
NACK
STOP

S	A6	A5	A4	A3	A2	A1	A0	0	A	B7	B6	B5	B4	B3	B2	B1	B0	A	Sr	A6	A5	A4	A3	A2	A1	A0	1	A	D7	D6	D5	D4	D3	D2	D1	D0	NA	P

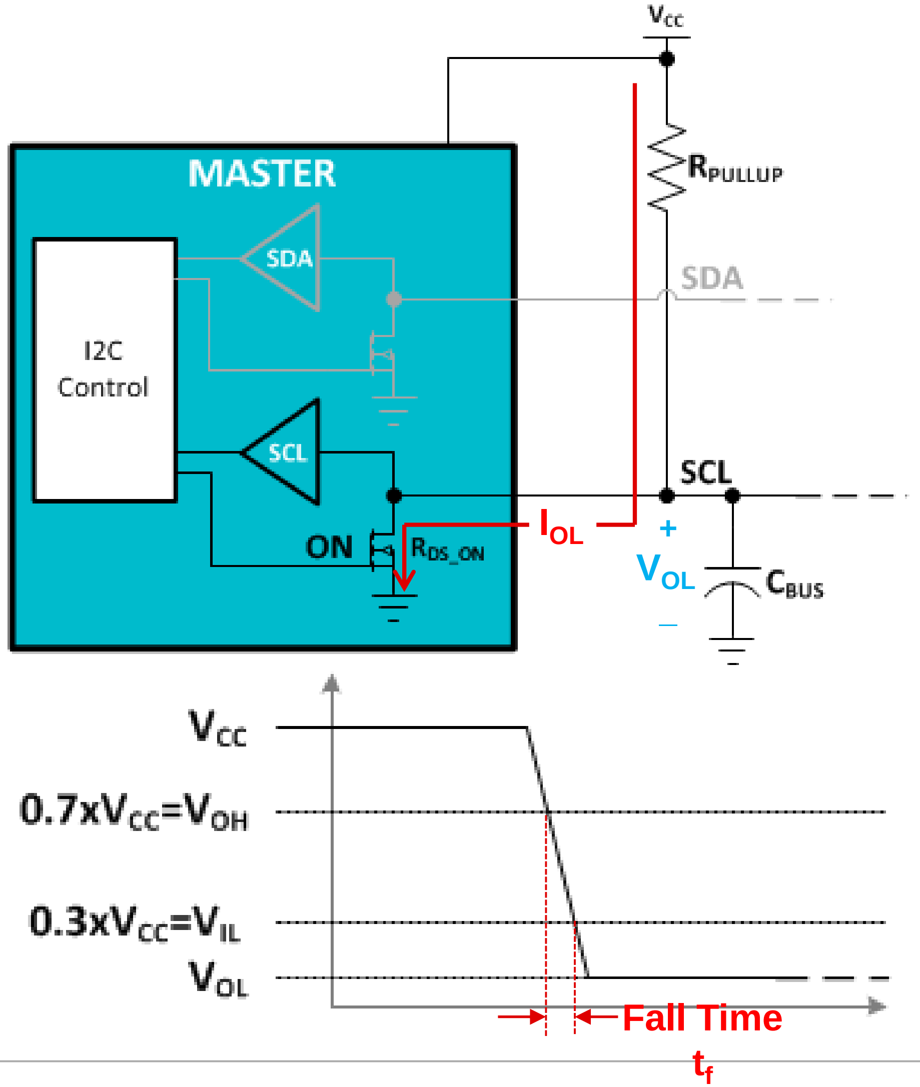
I^{2}C Physical Layer: Hardware



I^{2}C Physical Layer: Hardware (LOW)

Lets look at the physics of the High and Low generation. The SCL and SDA line will be treated as same.

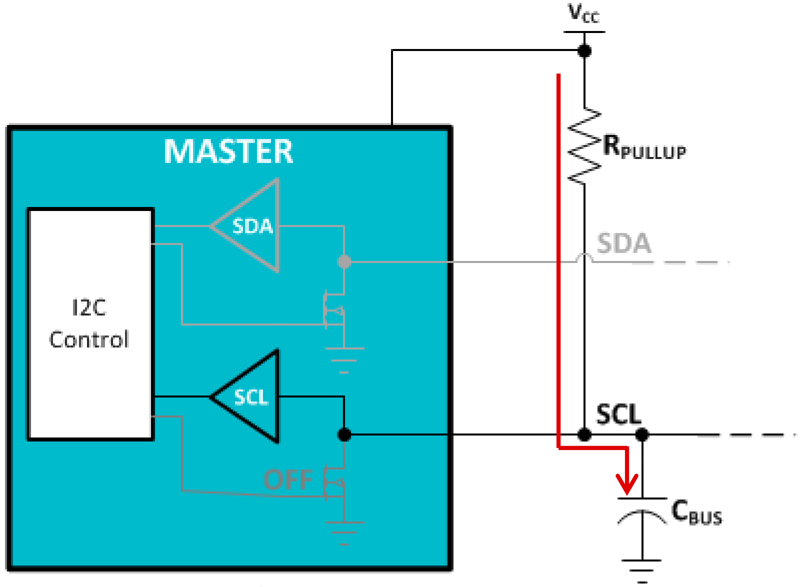
LOW: When Master Pulls SCL to ground for logic low.

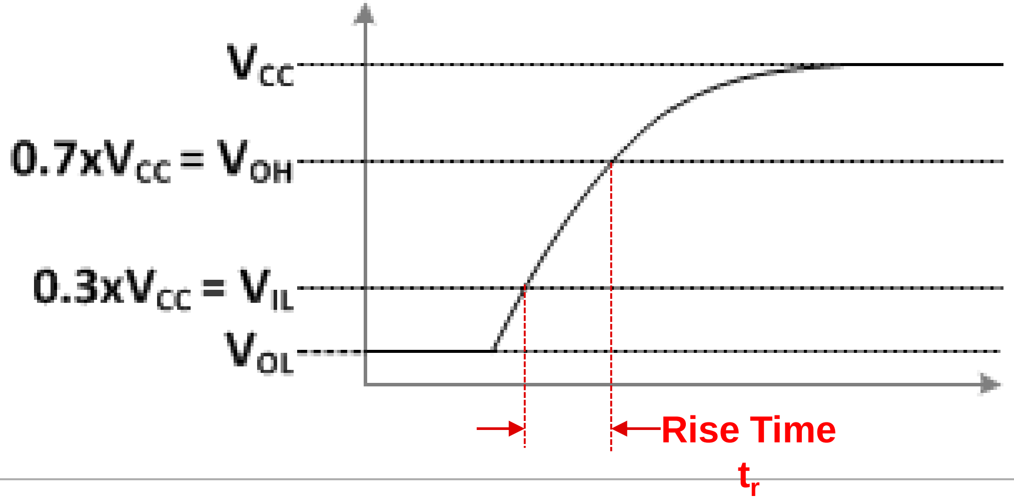


I^{2}C Physical Layer: Hardware (HIGH)

Lets look at the physics of the High and Low generation. The SCL and SDA line will be treated as same.

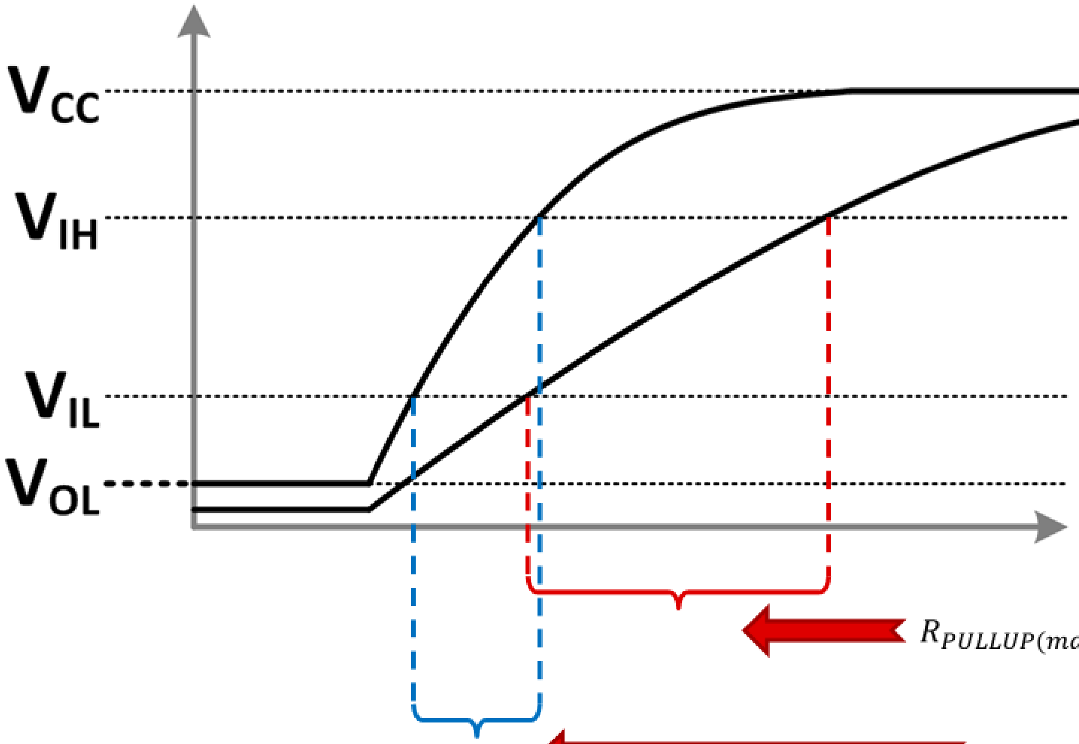
HIGH: When Master releases SCL from ground, R_{PULLUP} raises bus to V_{CC} for logic high.





I^2C Physical Layer: Bus Capacitance & Rise Times

Example: Standard Mode, $C_b = C_{b(max)} = 400pF$, $V_{CC} = 3.3V$



$$R_{PULLUP(max)} = \frac{t_{r(max)}}{0.8473 \times C_{BUS}} = \frac{1{,}000\ ns}{0.8473 \times 400\ pF} = 2{,}950.55\Omega$$

$$R_{PULLUP(min)} = \frac{V_{CC} - V_{OL(max)}}{I_{OL}} = \frac{3.3V - 0.4V}{3mA} = 966.67\Omega$$

$$966.67\Omega < R_{PULLUP} < 2{,}950.55\Omega$$

Standard-Mode:

- $f_{SCL(max)}$ = 100 kHz
- $t_{r(max)}$ = 1,000 ns
- $C_{b(max)}$ = 400 pF
- $V_{OL(max)}$ = 0.4V
- $I_{OL(min)}$ = 3mA

Fast-Mode:

- $f_{SCL(max)}$ = 400 kHz
- $t_{r(max)}$ = 300 ns
- $C_{b(max)}$ = 400 pF
- $V_{OL(max)}$ = 0.4V*
- $I_{OL(min)}$ = 3mA*

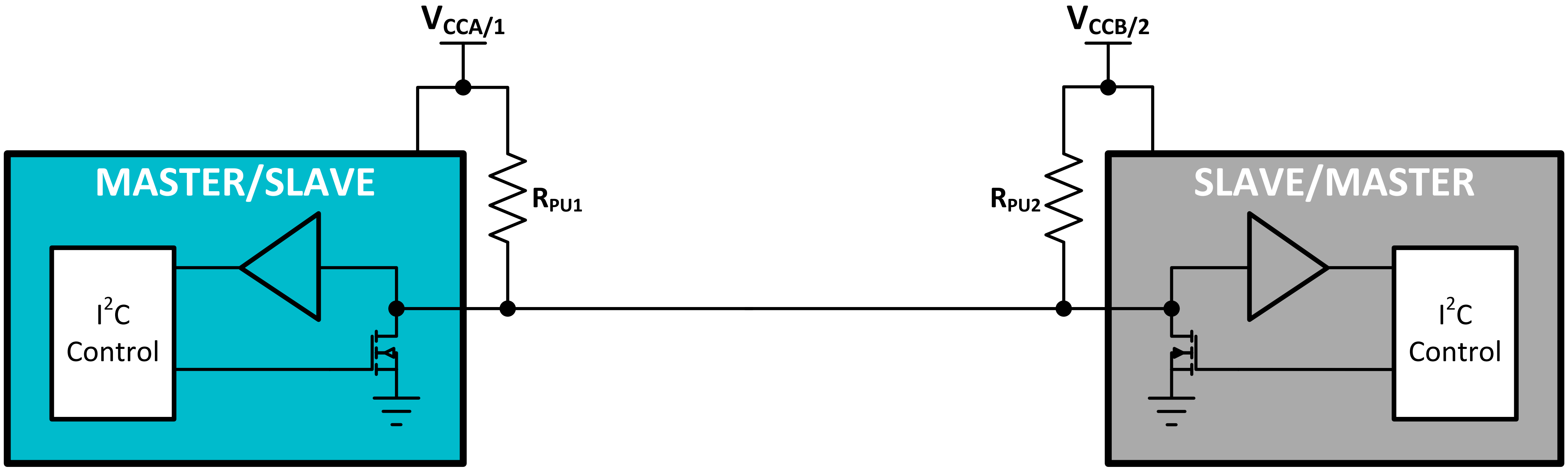
Fast-Mode Plus:

- $f_{SCL(max)}$ = 1,000 kHz
- $t_{r(max)}$ = 120 ns
- $C_{b(max)}$ = 550 pF
- $V_{OL(max)}$ = 0.4V*
- $I_{OL(min)}$ = 20mA

Translators

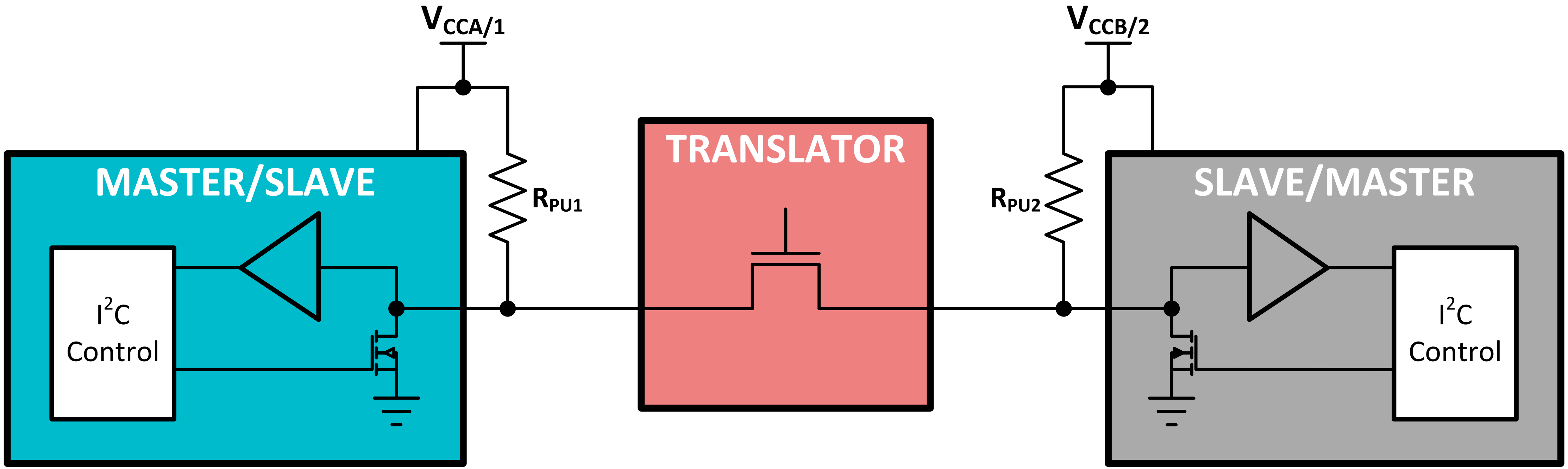
Translation of I²C Bus

What happens when two devices need to operate at different V_{CC} values?



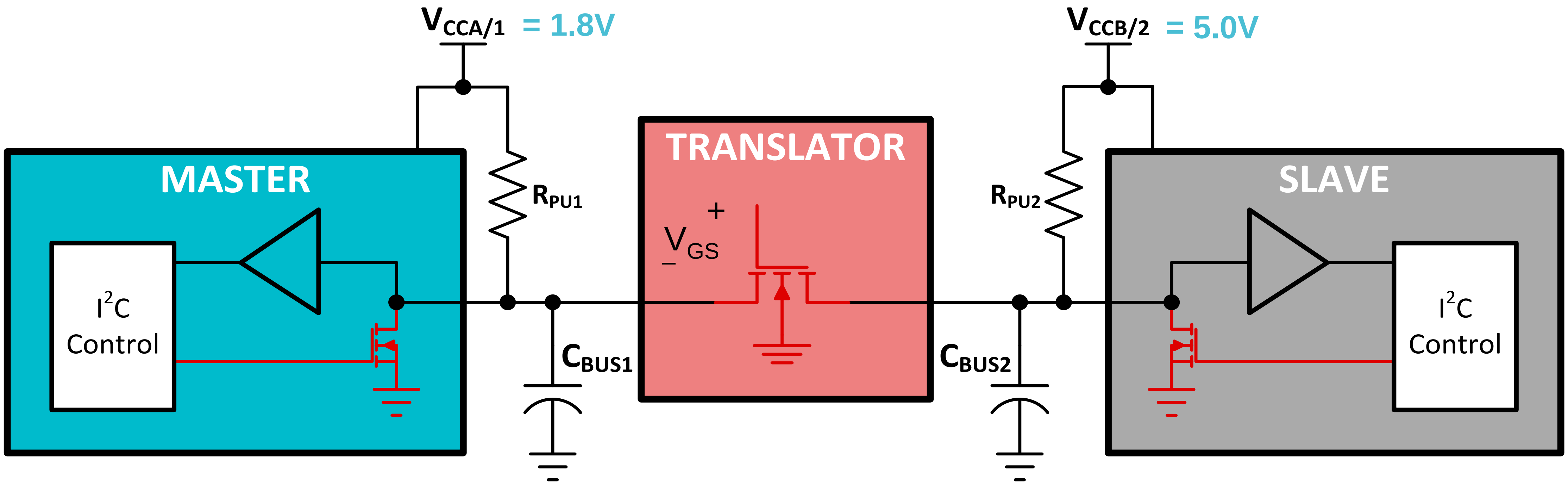
Translation of I^{2}C Bus

Pass Elements Based Devices



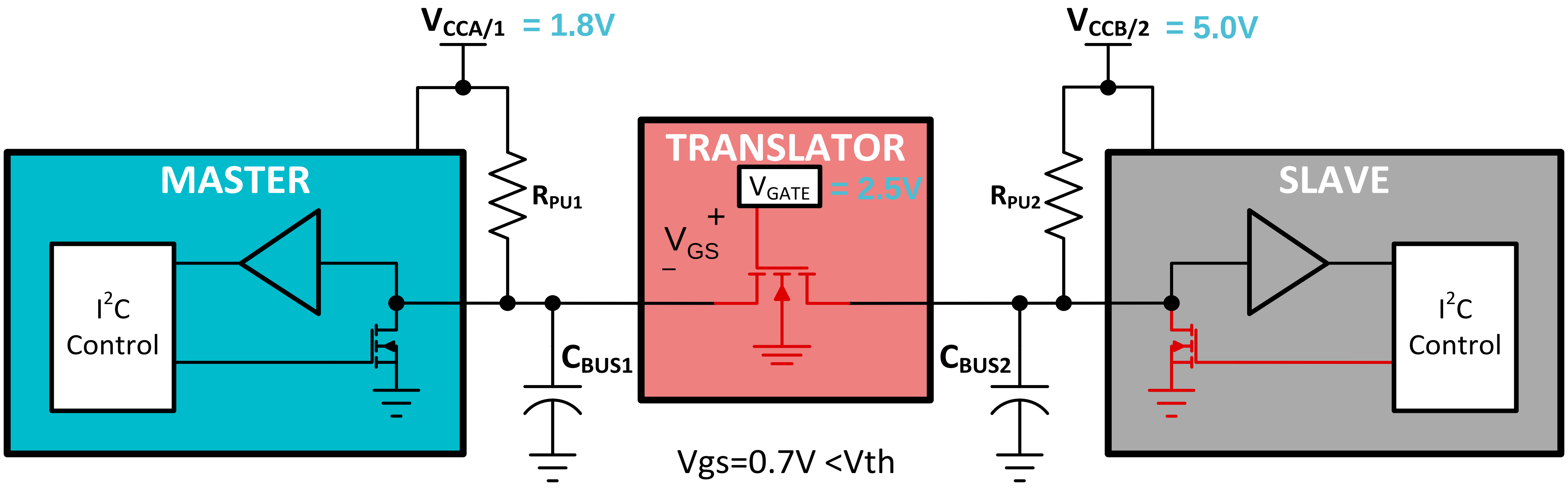
Translation of I^{2}C Bus

PCA9306 – Pass FET Architecture



Translation of I^{2}C Bus

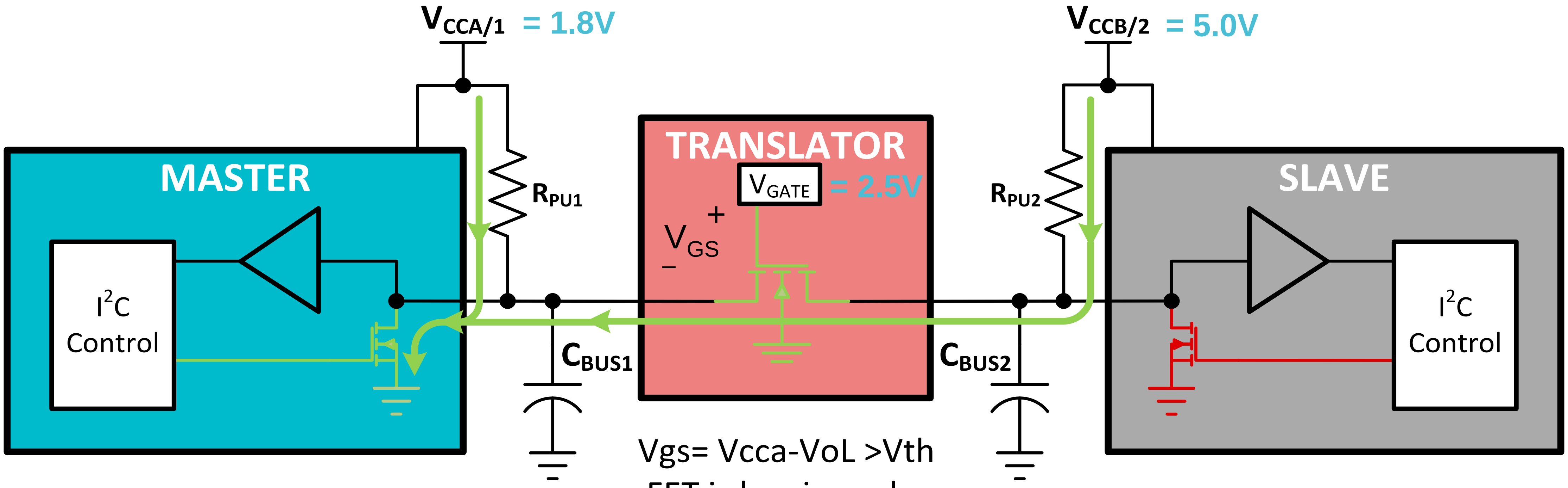
PCA9306 – Pass FET Architecture



Vgs=0.7V <Vth
so FET is high impedance

Translation of I²C Bus

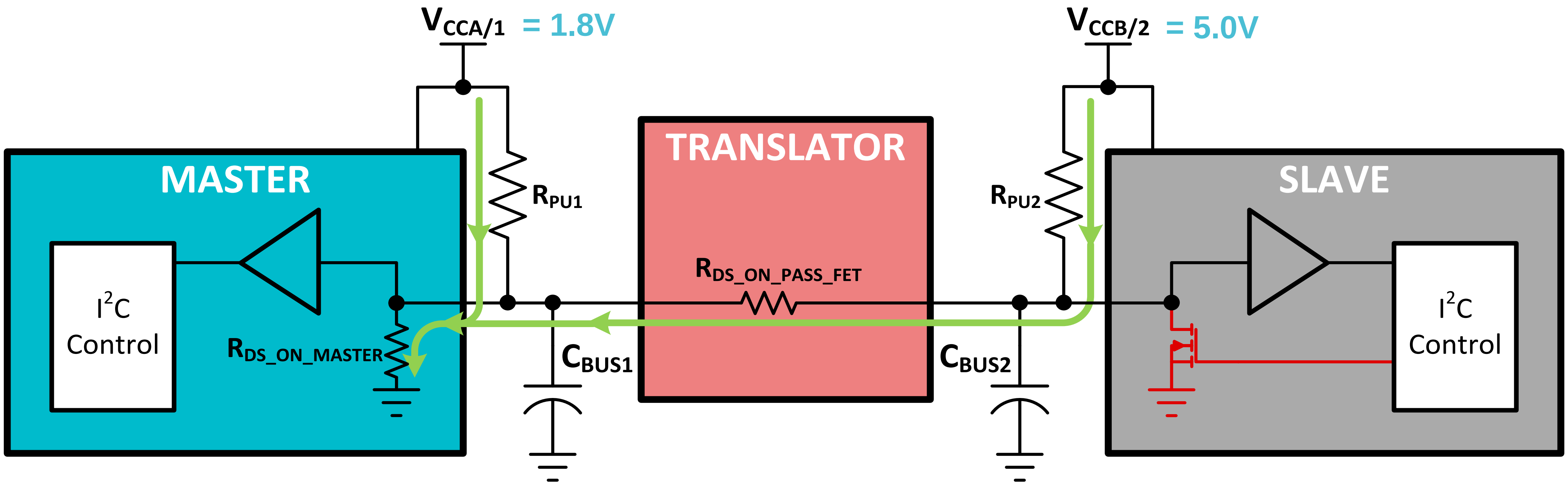
PCA9306 – Pass FET Architecture



Vgs= Vcca-VoL >Vth
so FET is low impedance

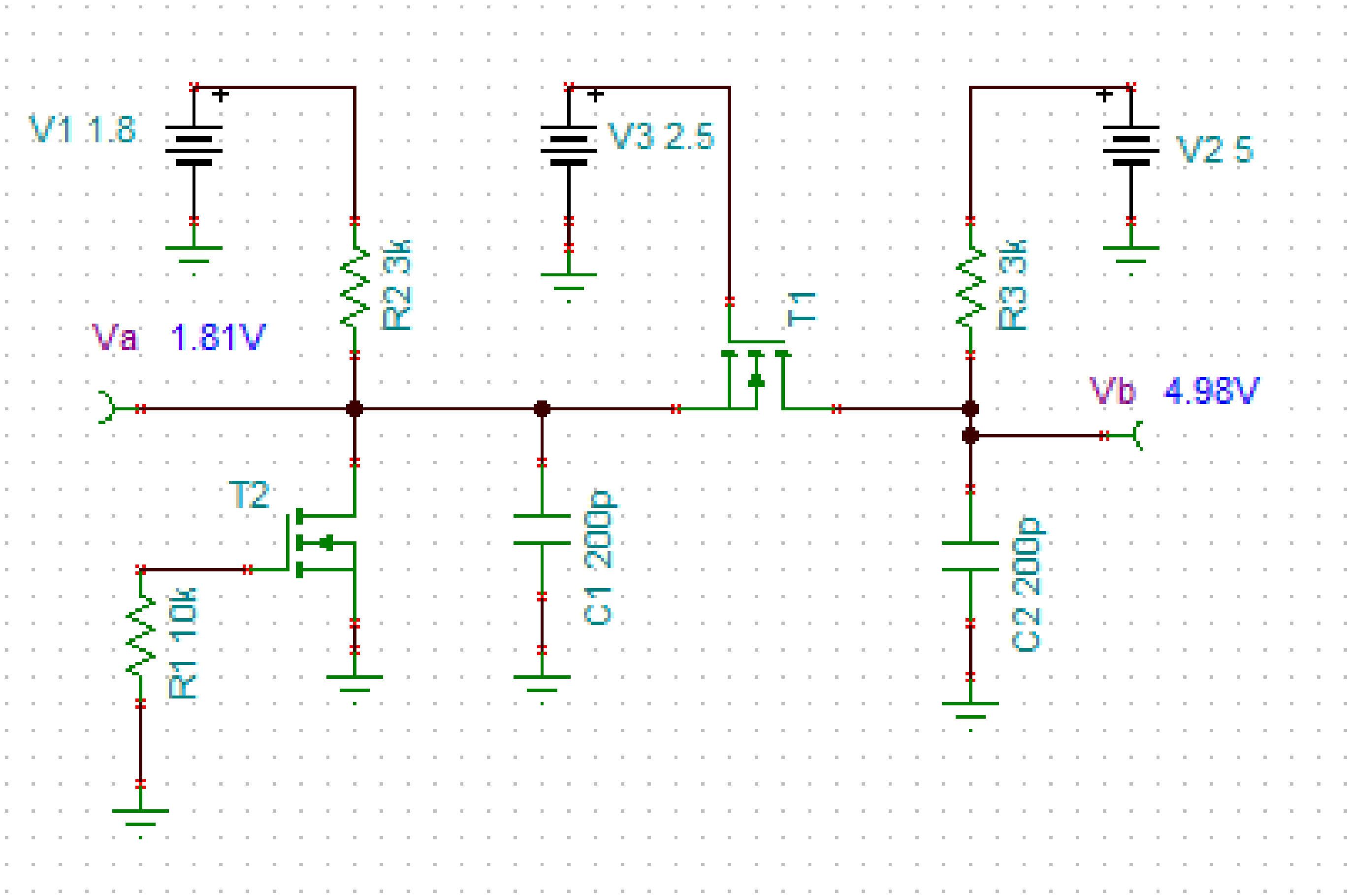
Translation of I²C Bus

PCA9306 – Pass FET Architecture



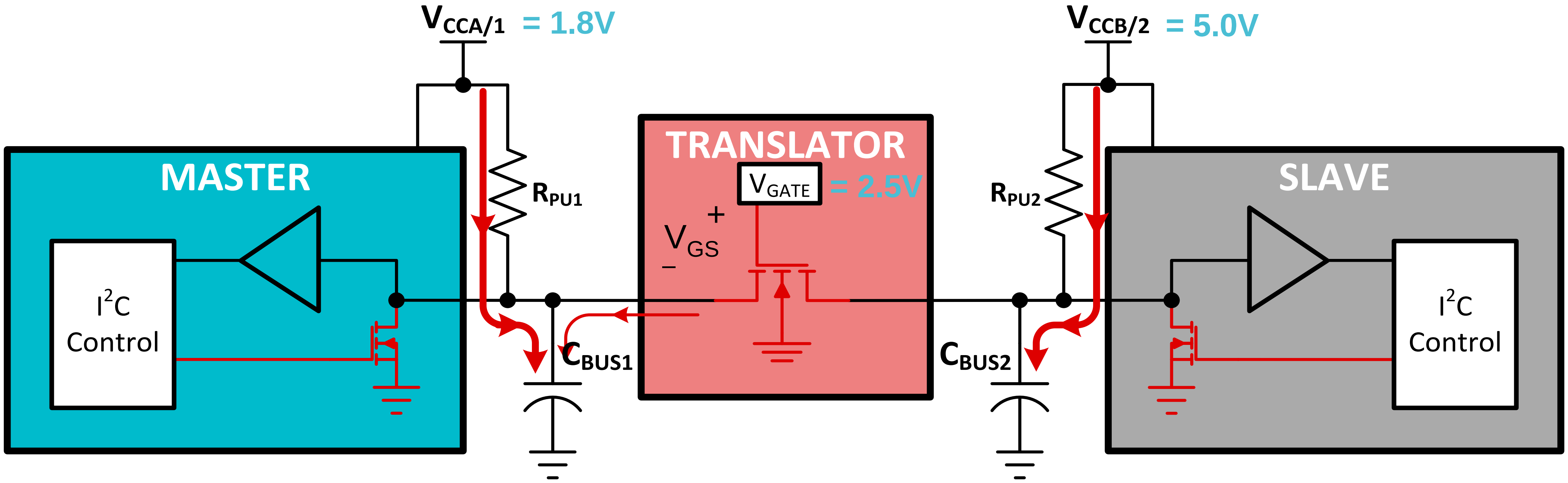
Translation of I^{2}C Bus

PCA9306 – TI TINA equivalent circuit for resistance network



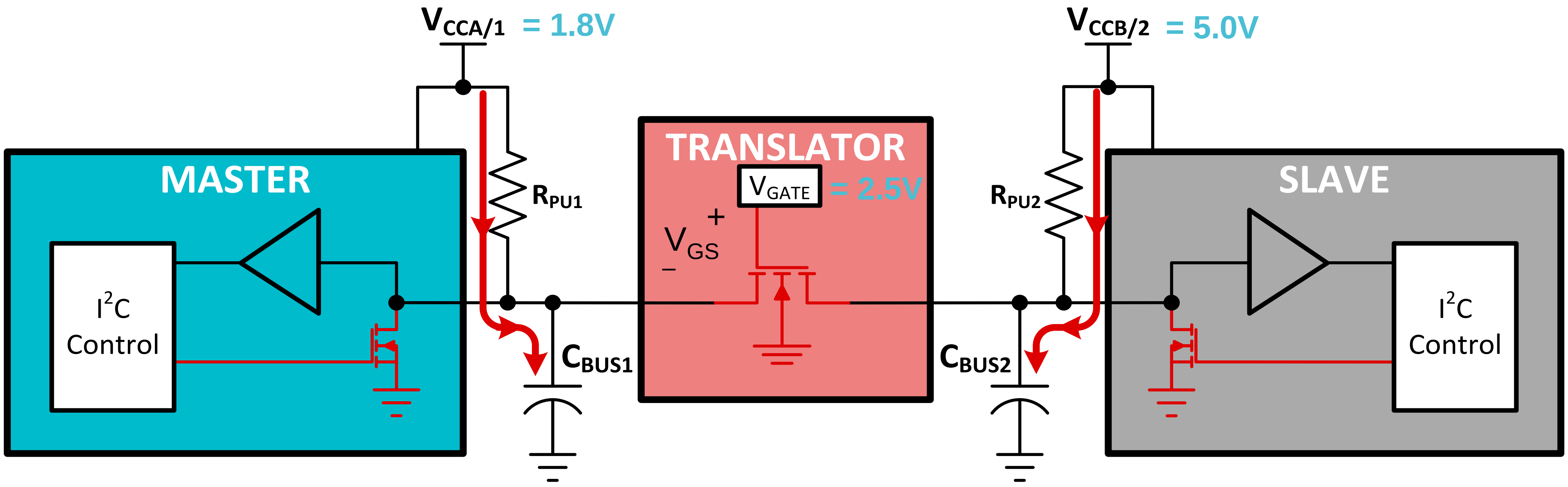
Translation of I²C Bus

PCA9306 – Pass FET Architecture



Translation of I²C Bus

PCA9306 – Pass FET Architecture



Translation/Level Shifting of I^{2}C Bus

Design Considerations:

- Translators introduce propagation delays
  - Delays become longer the further Vref1 and Vref2 are from each other
    - Example: if Vref1 and Vref2 difference is 3V, propagation delays will be lower if Vref1 and Vref2 at 1V
  - Placing Translators in series will further increase propagation delays and could cause timing issues
  - Higher gate voltages will lower propagation delays
- Translators will not buffer capacitance on both sides of the translator.

Translation/Level Shifting of I^2C Bus

PCA9306	PCA9306-Q1	TCA9406

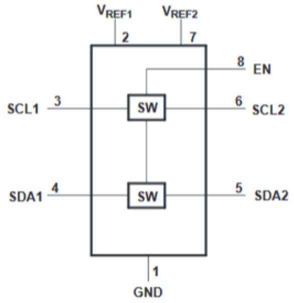


Figure 6. Block Diagram of PCA9306

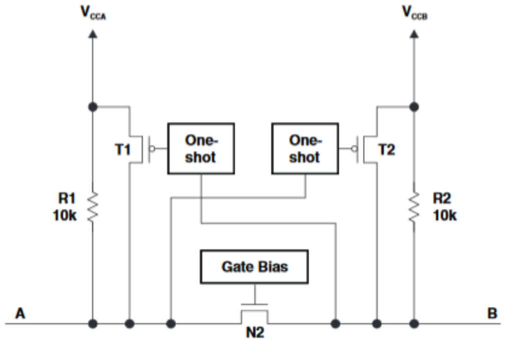


Figure 5. Architecture of a TCA9406 Cell

PCA9306

- Features
  - Could be used as a switch via EN pin
  - Small Package Available
- Pros:
  - More configurable
- Cons:
  - Wrong configuration can be damaged

TCA9406

- Features:
  - Internal 10k Pull ups
  - Small Package Available
  - Rise Time Accelerators support larger bus capacitance.
- Pros:
  - Supports up to 1MHz
  - Can use larger pull ups to get lower V_{OL}
- Cons:
  - RTA can cause glitches in some conditions
  - V_{CCA} supports max of 3.6V

Translators

Common questions

I^{2}C Devices: Translators

Common Question:

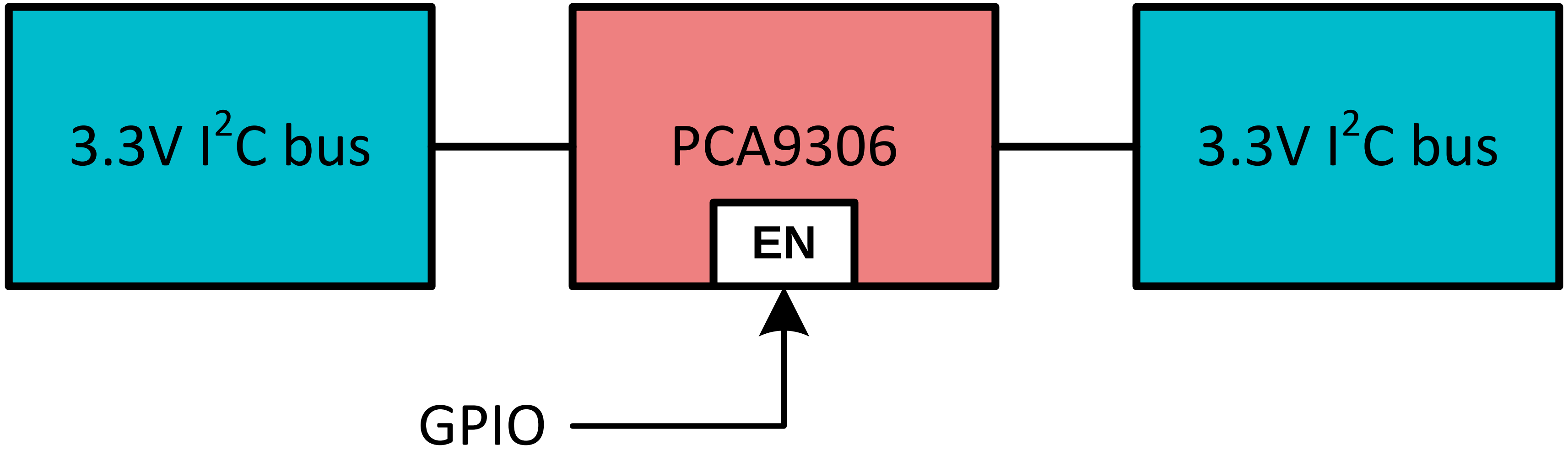
- Difference between PCA9306/TCA9406 and <<insert buffer w/ translation here>>?
  - PCA9306/TCA9406 do not provide active buffering because it is a pass FET architecture.
  - Buffers **typically** have an offset on one side and PCA9306/TCA9406 pass VoL to the opposite side with little voltage increase due to voltage drop due to pass FET resistance and pull down current.

I^2C Devices: Translators

Common Questions:

- Difference between PCA9306/TCA9406 and <<insert buffer w/ translation here>>?

- Can I use PCA9306/TCA9406 to isolate an I2C line at the same voltage on both sides?

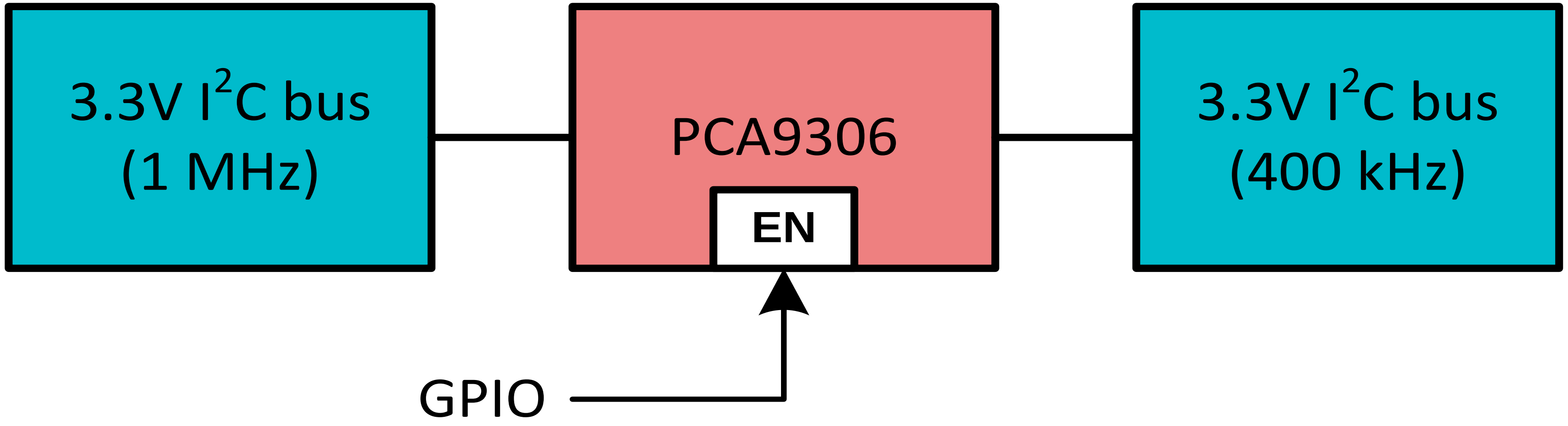


  – Yes, but why would you want to do this?

I^2C Devices: Translators

Common Questions:

Can I use PCA9306/TCA9406 to isolate an I2C line at the same voltage on both sides?



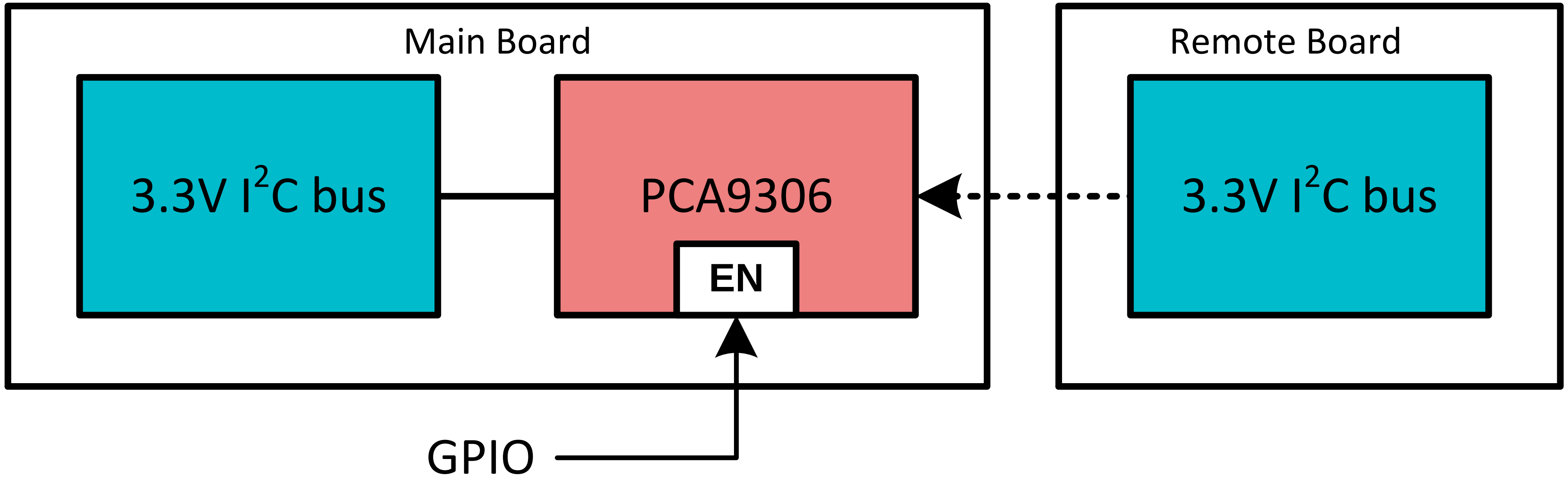
Why?

1. Could be used to disable communication with slaves who don’t support 1MHz

I^{2}C Devices: Translators

Common Questions:

Can I use PCA9306/TCA9406 to isolate an I2C line at the same voltage on both sides?



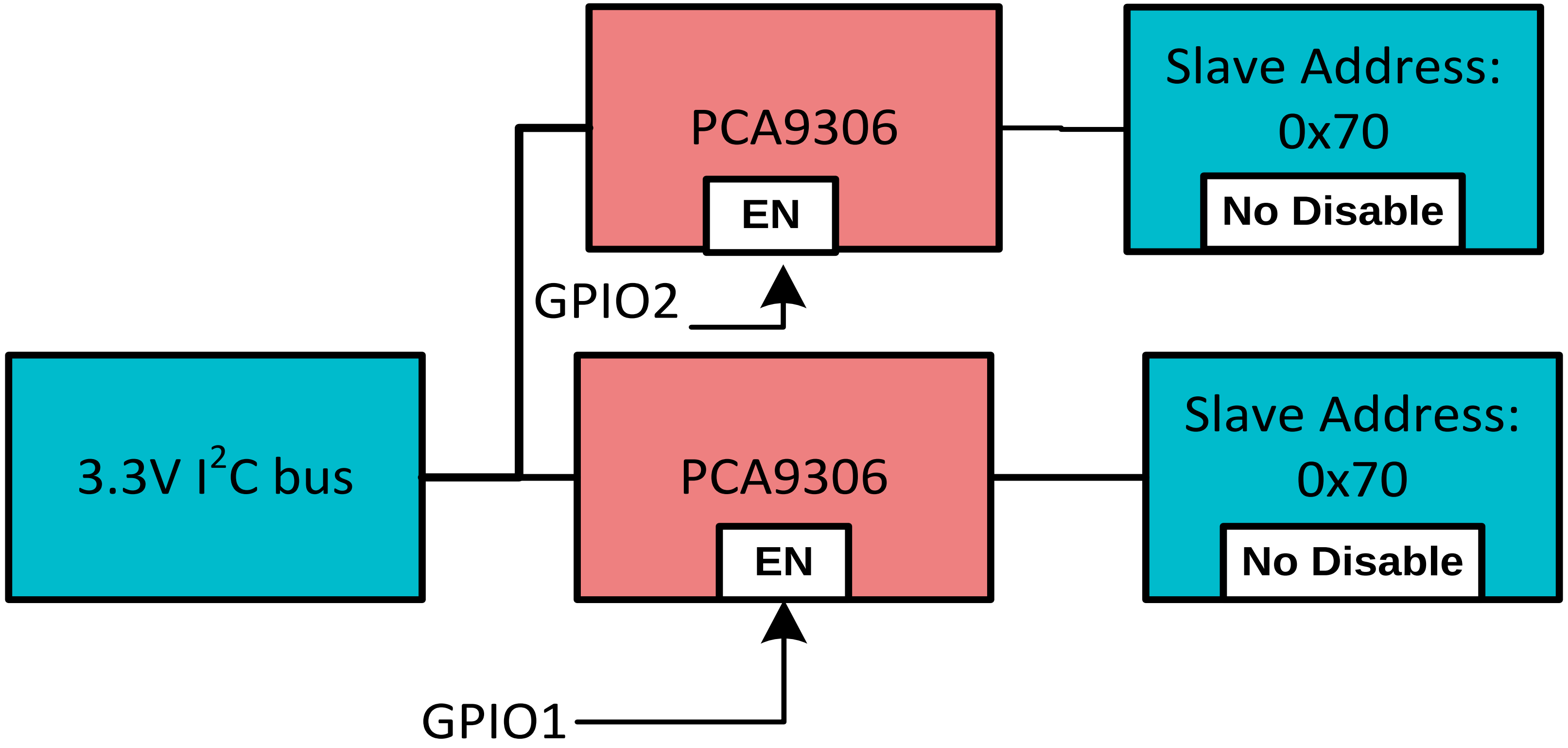
Why?

2. One bus may plug into another: PCA9306 could be disabled during insertion and enabled when I2C line is idle to support “hot insertion”

I^2C Devices: Translators

Common Questions:

Can I use PCA9306/TCA9406 to isolate an I2C line at the same voltage on both sides?



Why?

3. A switch to disable an I2C device which has address conflicts

I²C Devices: Translators

Common Question:

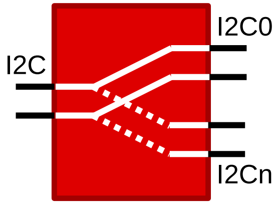
- Difference between PCA9306/TCA9406 and <<insert buffer w/ translation here>>?

- Can I use PCA9306/TCA9406 to isolate a I2C line at the same voltage on both sides?

- Does master need to be on side 1 of the device? (The datasheet pictures show master only on side 1)
  - Both PCA9306 and TCA9406 are Bidirectional and do not know/care which side master is on

Switches

When to use… a switch



Switch vs. multiplexer (Mux)

- Switch = 1 or many channels can be enabled at a time
- Mux = only 1 channel can be enabled at a time
- Switches can be used as muxes but muxes cannot be used as switches

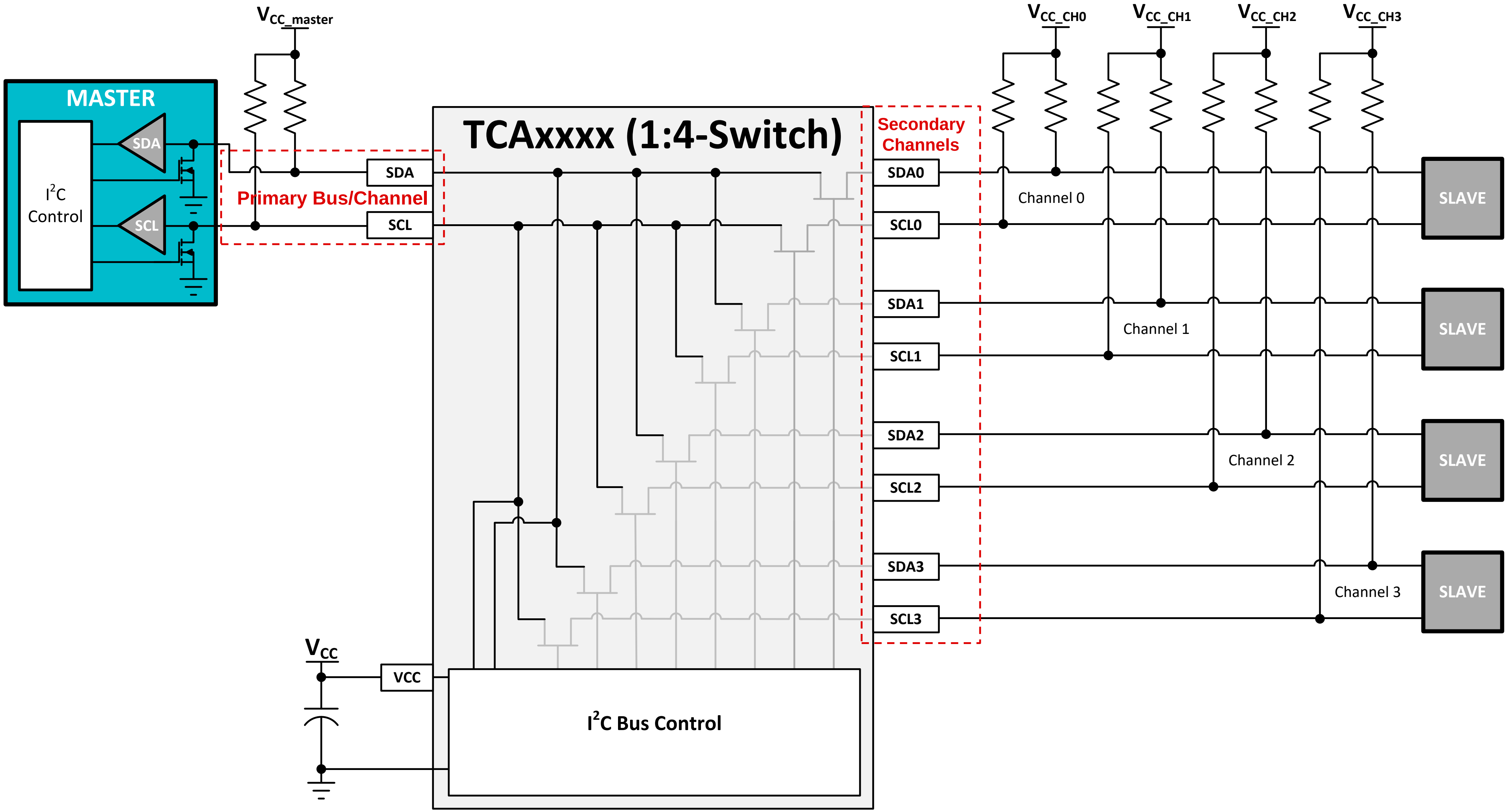
Slave address conflicts → Muxing/Switching is needed

- No address pin limits to 1 device per channel,
- Single ADDR pin limits to 2 devices per channel,
- Two address pins (A0, A1) limits to 4 devices per channel, etc.

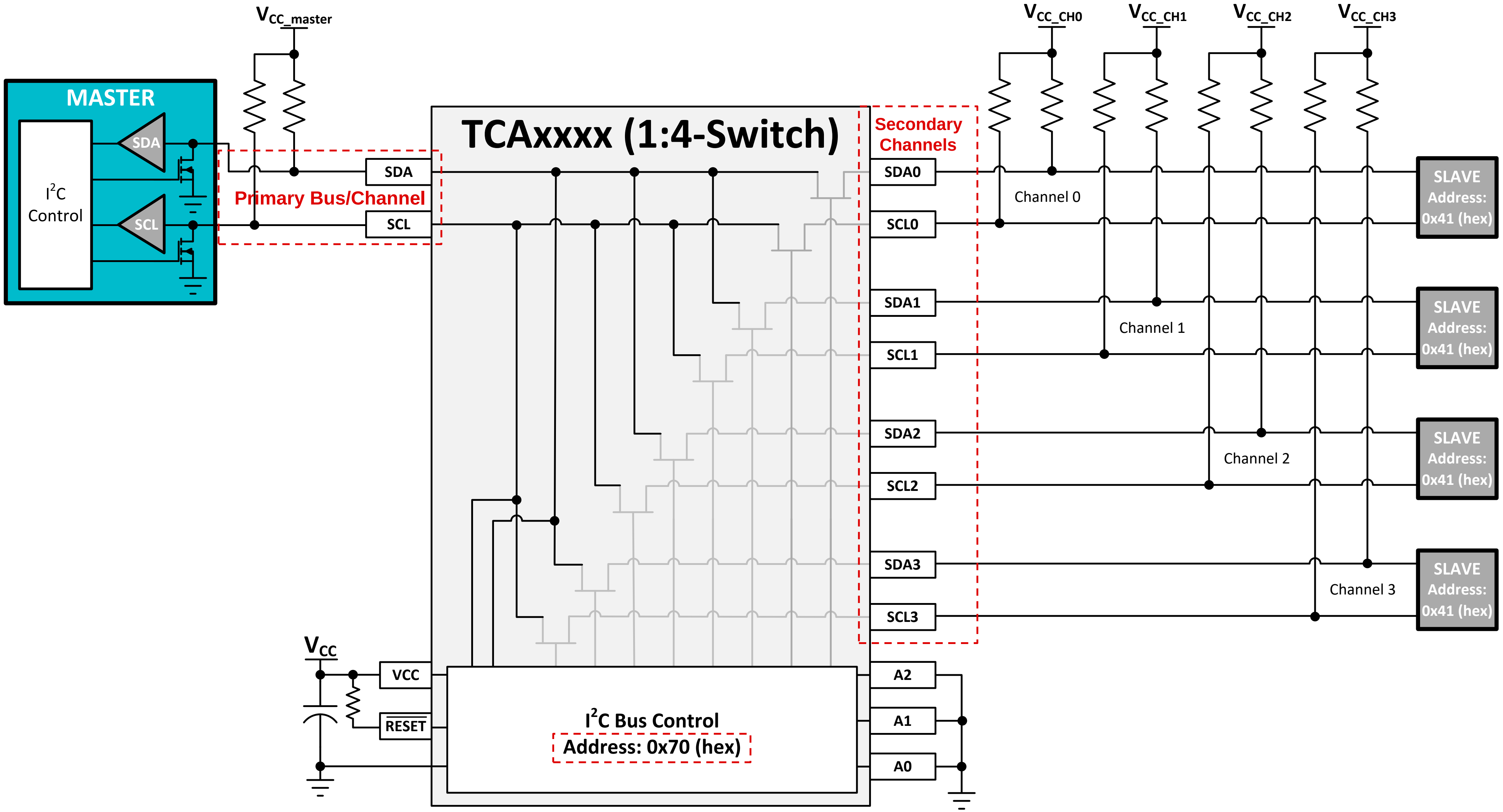
What makes an I2C switch special?

- Ideal speed (supports 400kHz for I2C fast-mode clock speed)
- I2C-controlled (no need to waste GPIOs to control channel selection)

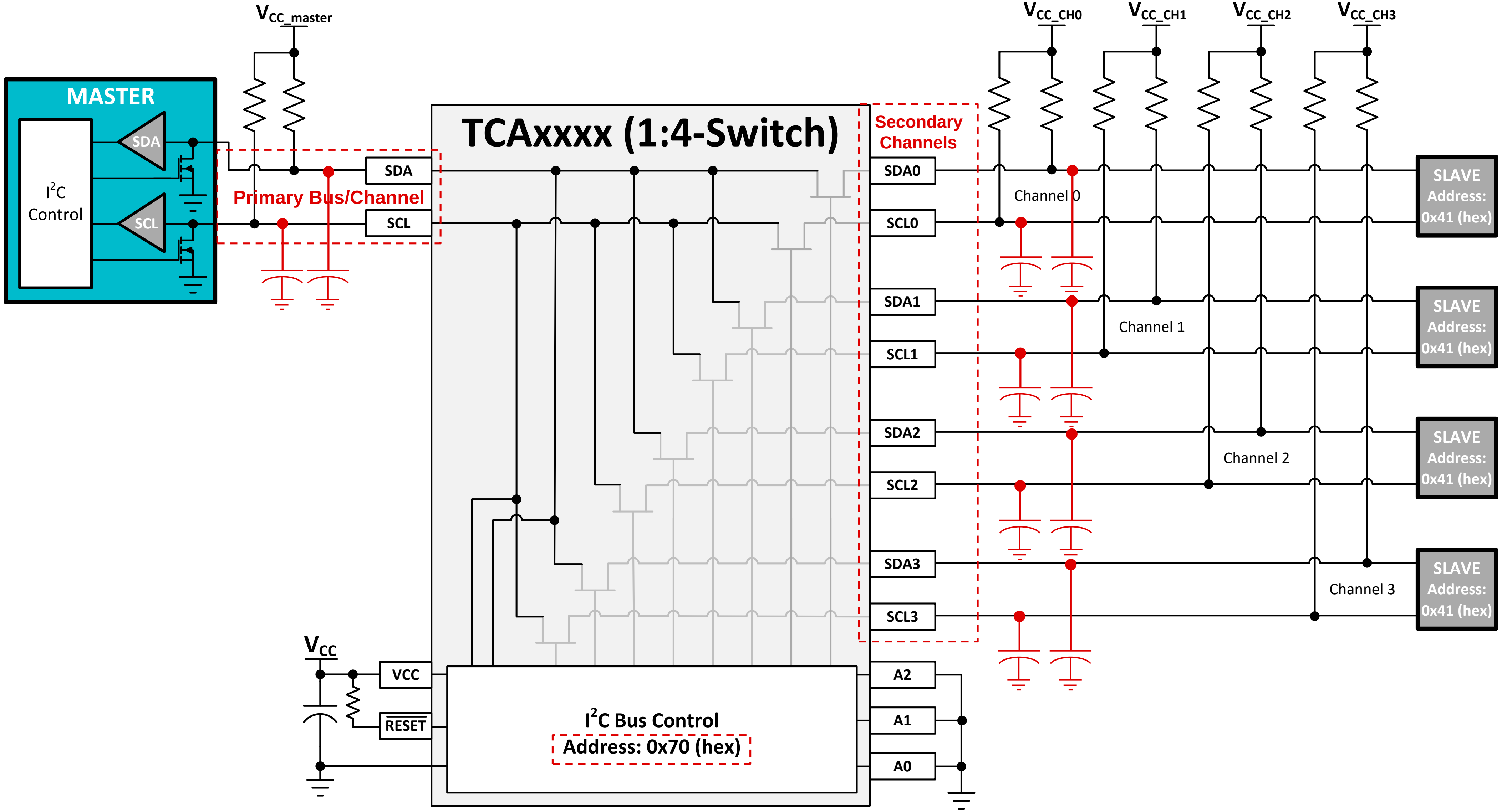
I^2C Devices: Switches: Introduction



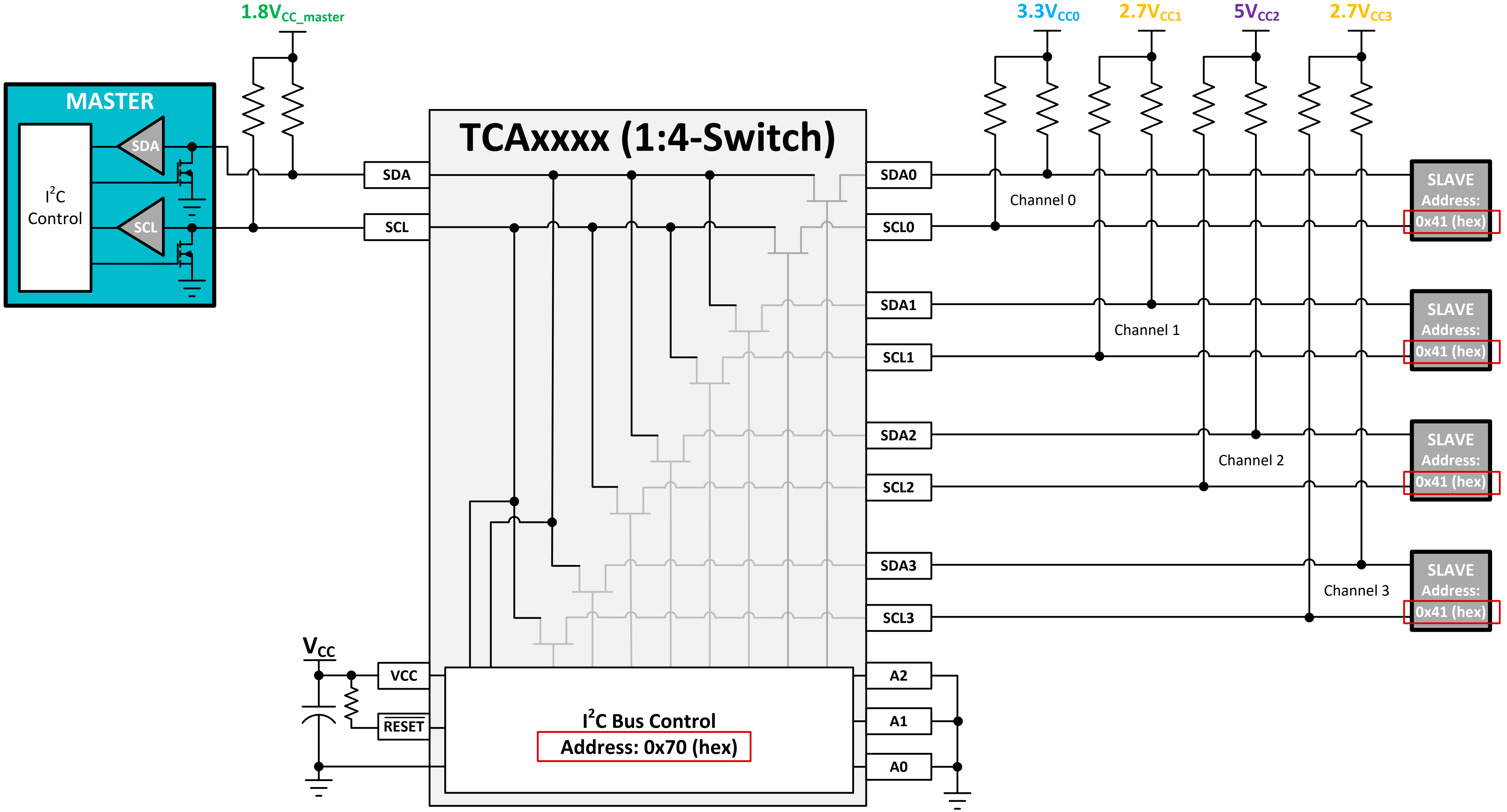
I²C Devices: Muxes – Switches: Introduction



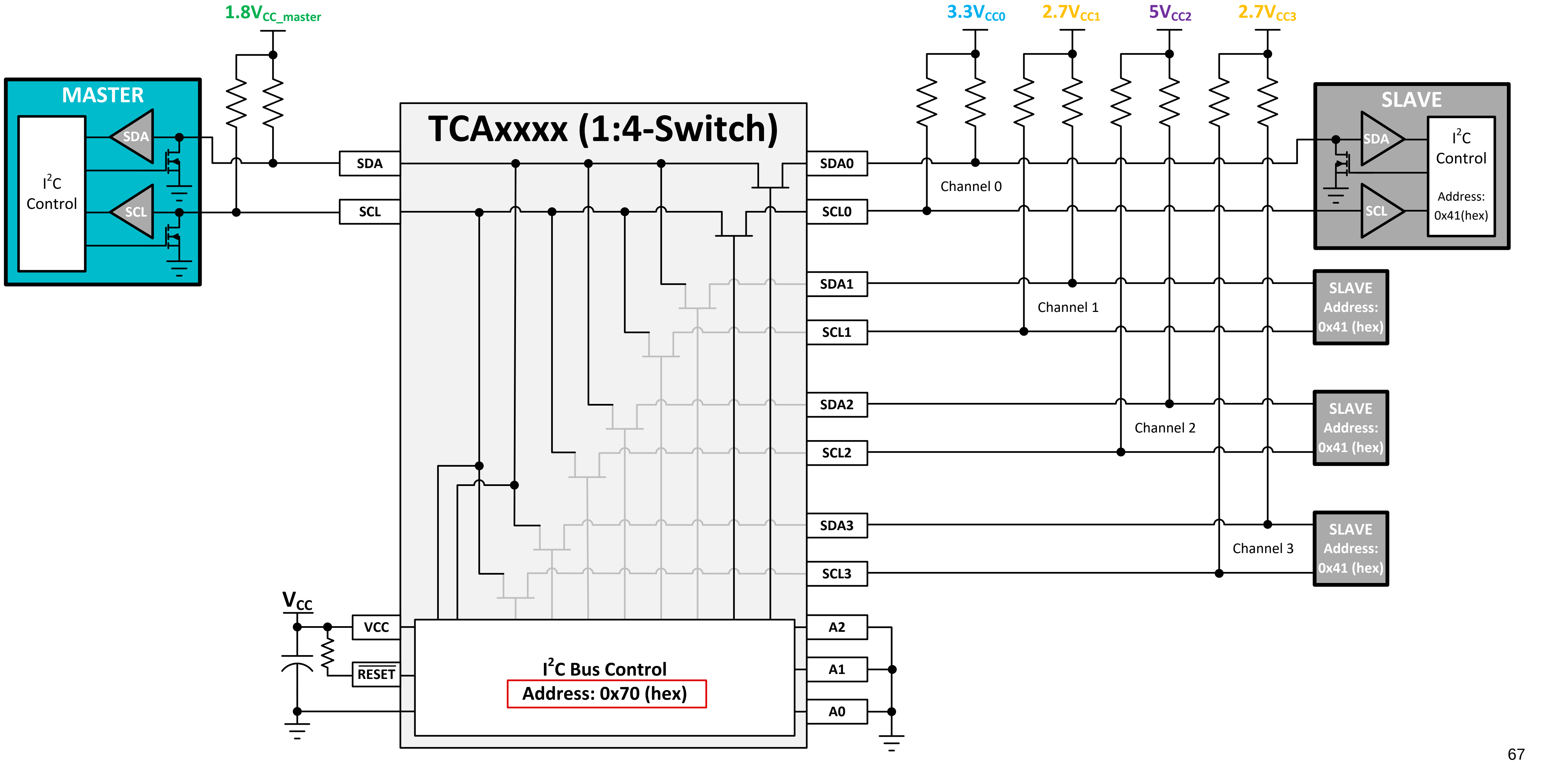
I²C Devices: Muxes – Switches: Introduction



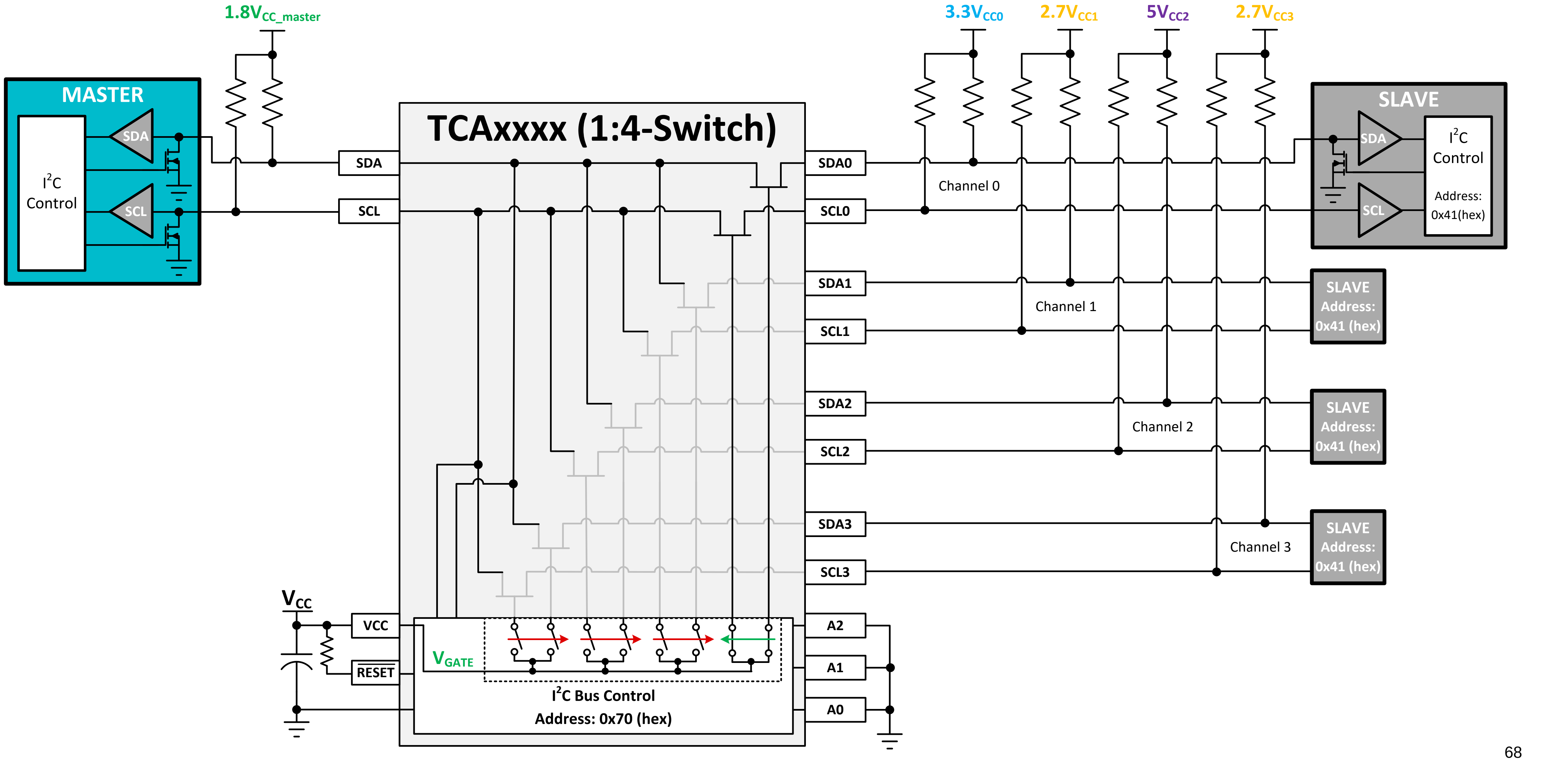
I^2C Devices: Muxes – Switches: Introduction



I^{2}C Devices: Muxes - Switches: Theory of Operation



I^{2}C Devices: Muxes - Switches: Theory of Operation



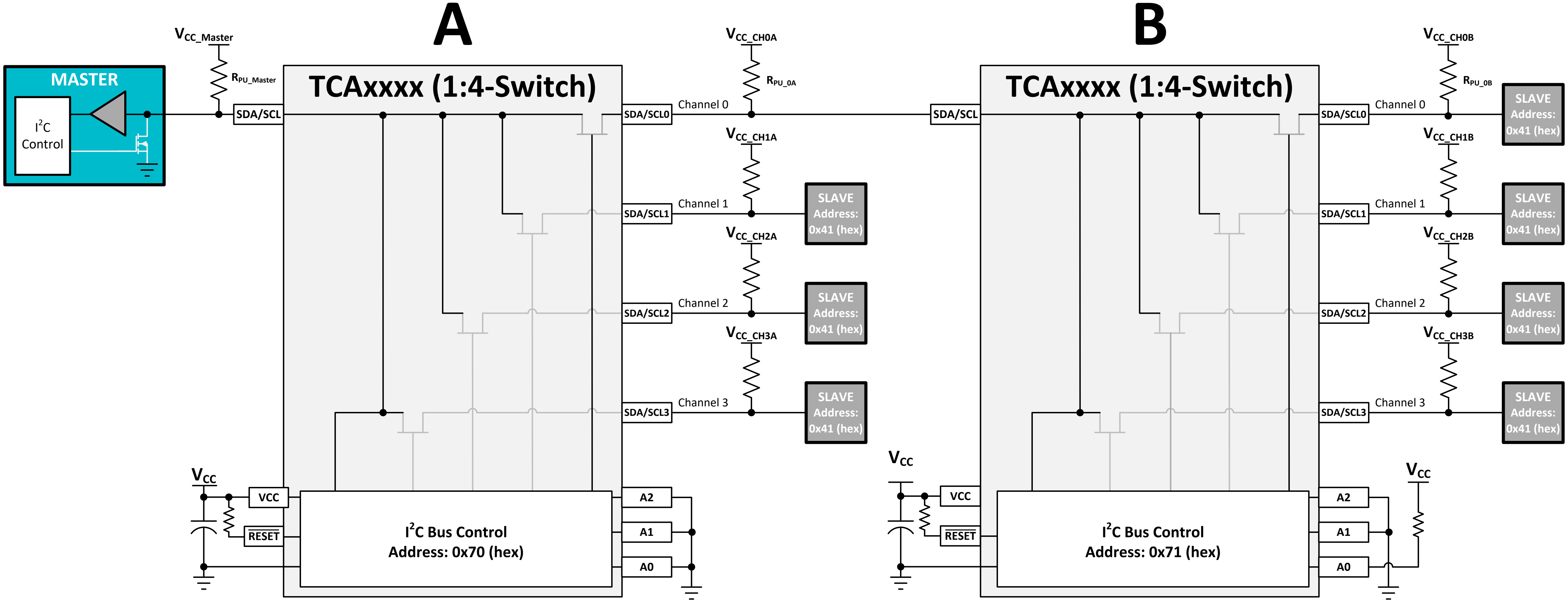
Switches

Design Considerations

Switches in Series

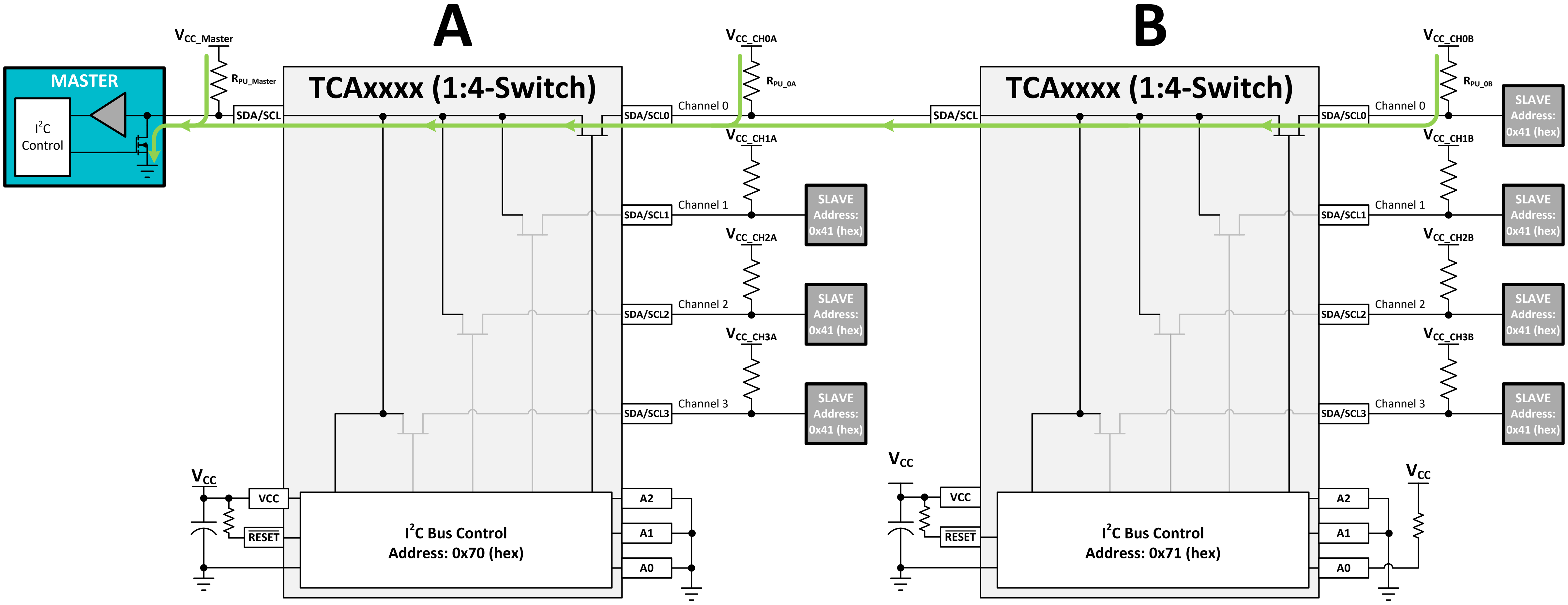
I^2C Devices: Muxes - Switches

Design Considerations- Switches in series



I²C Devices: Muxes - Switches

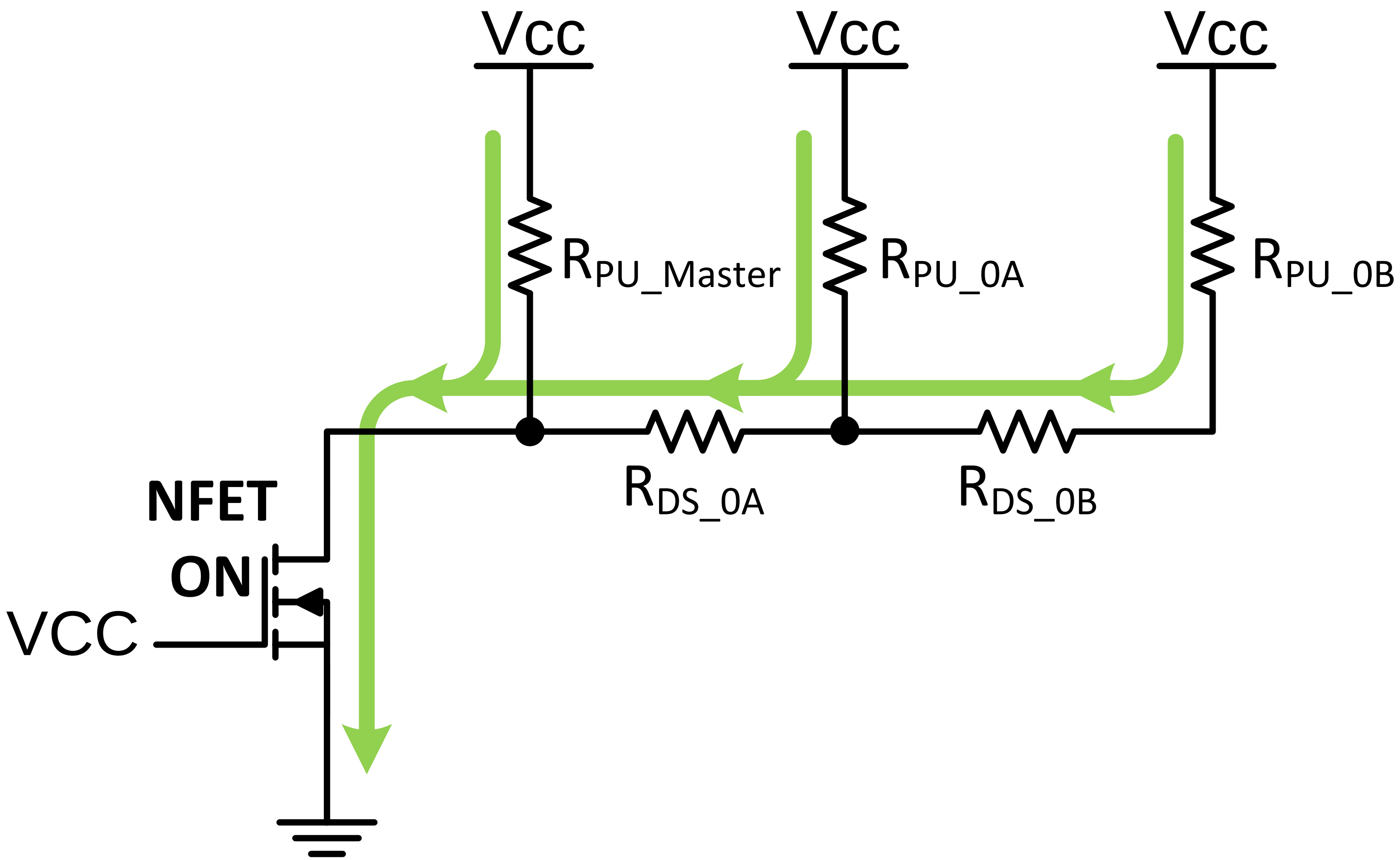
Design Considerations- Switches in series



I²C Devices: Muxes - Switches

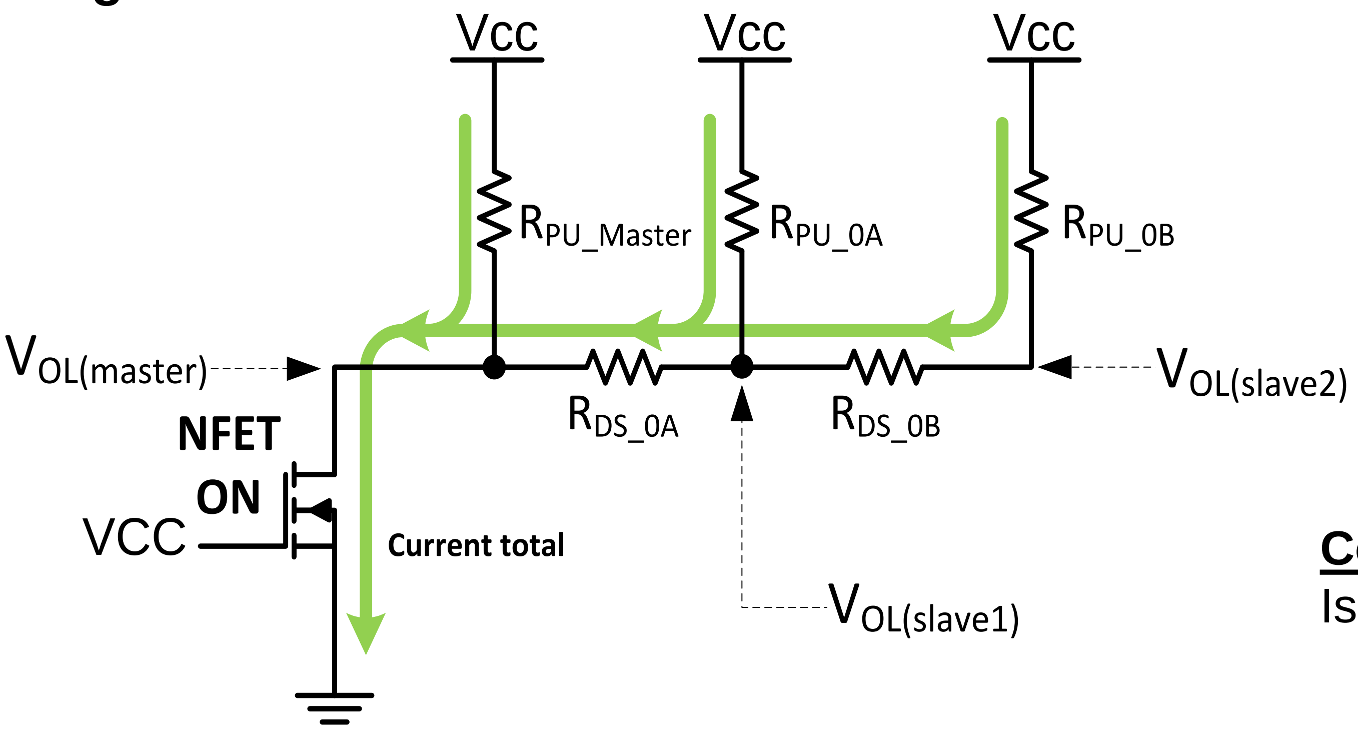
Design Considerations- Switches in series

Simplified circuit



I²C Devices: Muxes - Switches

Design Considerations- Switches in series



Concerns:

Is V(slave)>ViL(slave)?

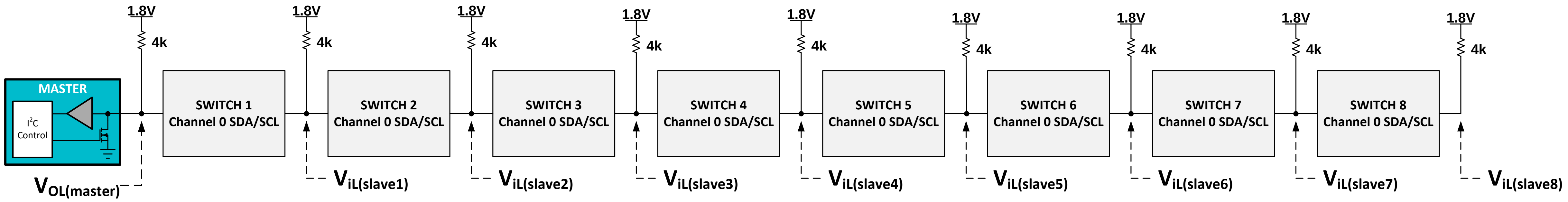
- $V_{OL_{(master)}} = Ron * Current_{total}$
- $V_{(slave1)} = V_{OL_{(master)}} + Rds_{0A} * (Current_{0A} + Current_{0B})$
- $V_{(slave2)} = V_{(slave1)} + Rds_{0B} * Current_{0B}$

I^{2}C Devices: Muxes - Switches

Design Considerations- Switches in series

Lets Consider a more extreme example with 8 switches in series

Assumptions for this example:	
Pull up resistors	4k ohms
Master’s Ron	110 ohms
Pull up voltages	1.8V
Rds of all channels	20 ohms

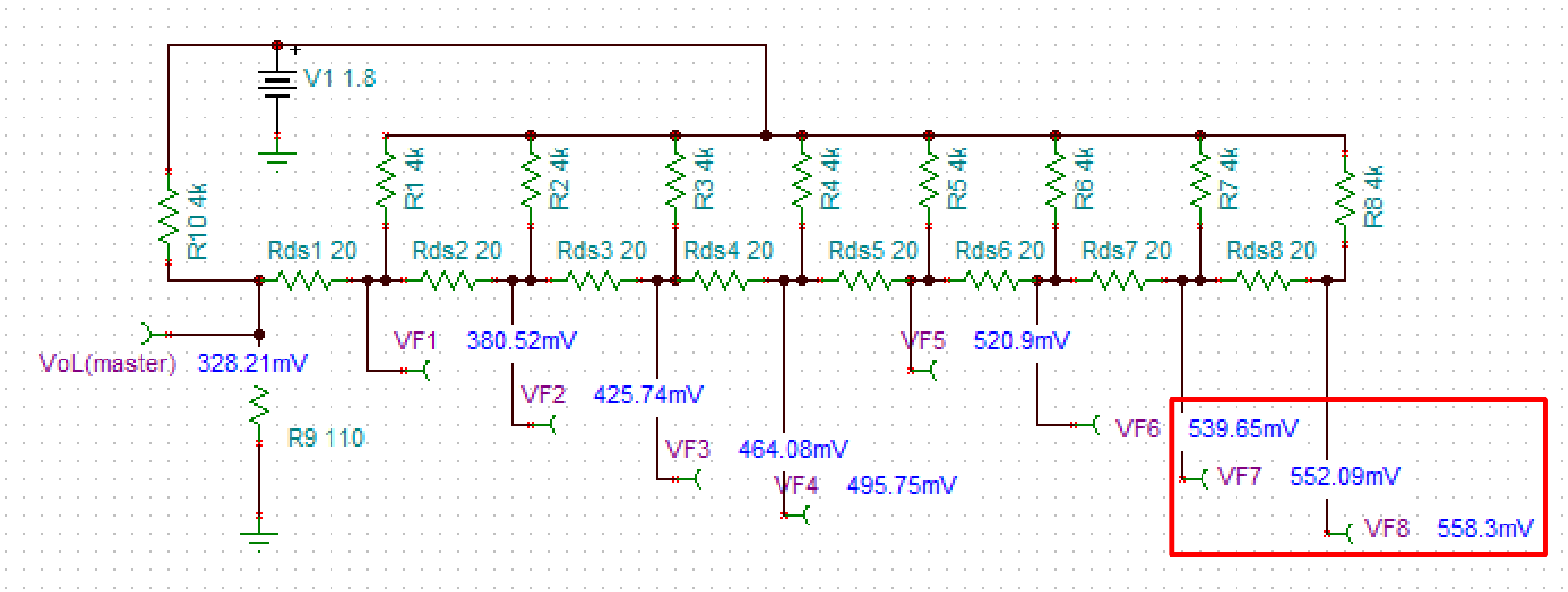


1.8V
4k
1.8V
4k
1.8V
4k
1.8V
4k
1.8V
4k
1.8V
4k
1.8V
4k
1.8V
4k
1.8V
4k
MASTER
I²C Control
SWITCH 1 Channel 0 SDA/SCL
SWITCH 2 Channel 0 SDA/SCL
SWITCH 3 Channel 0 SDA/SCL
SWITCH 4 Channel 0 SDA/SCL
SWITCH 5 Channel 0 SDA/SCL
SWITCH 6 Channel 0 SDA/SCL
SWITCH 7 Channel 0 SDA/SCL
SWITCH 8 Channel 0 SDA/SCL
V_OL(master)
V_iL(slave1)
V_iL(slave2)
V_iL(slave3)
V_iL(slave4)
V_iL(slave5)
V_iL(slave6)
V_iL(slave7)
V_iL(slave8)

I^{2}C Devices: Muxes - Switches

Design Considerations- Switches in series

Simulation results:



V seen by slaves must be <ViL to register as a valid low

I^{2}C Devices: Muxes - Switches

Design Considerations- Switches in series

Analyzing results:

Device	Voltage (V)	ViL required (Vcc*30%)	Is V<ViL?
Master	0.328	0.540	Yes
Slave 1	0.380	0.540	Yes
Slave 2	0.425	0.540	Yes
Slave 3	0.464	0.540	Yes
Slave 4	0.495	0.540	Yes
Slave 5	0.520	0.540	Yes
Slave 6	0.540	0.540	Yes/No
Slave 7	0.552	0.540	No
Slave 8	0.558	0.540	No

Switches

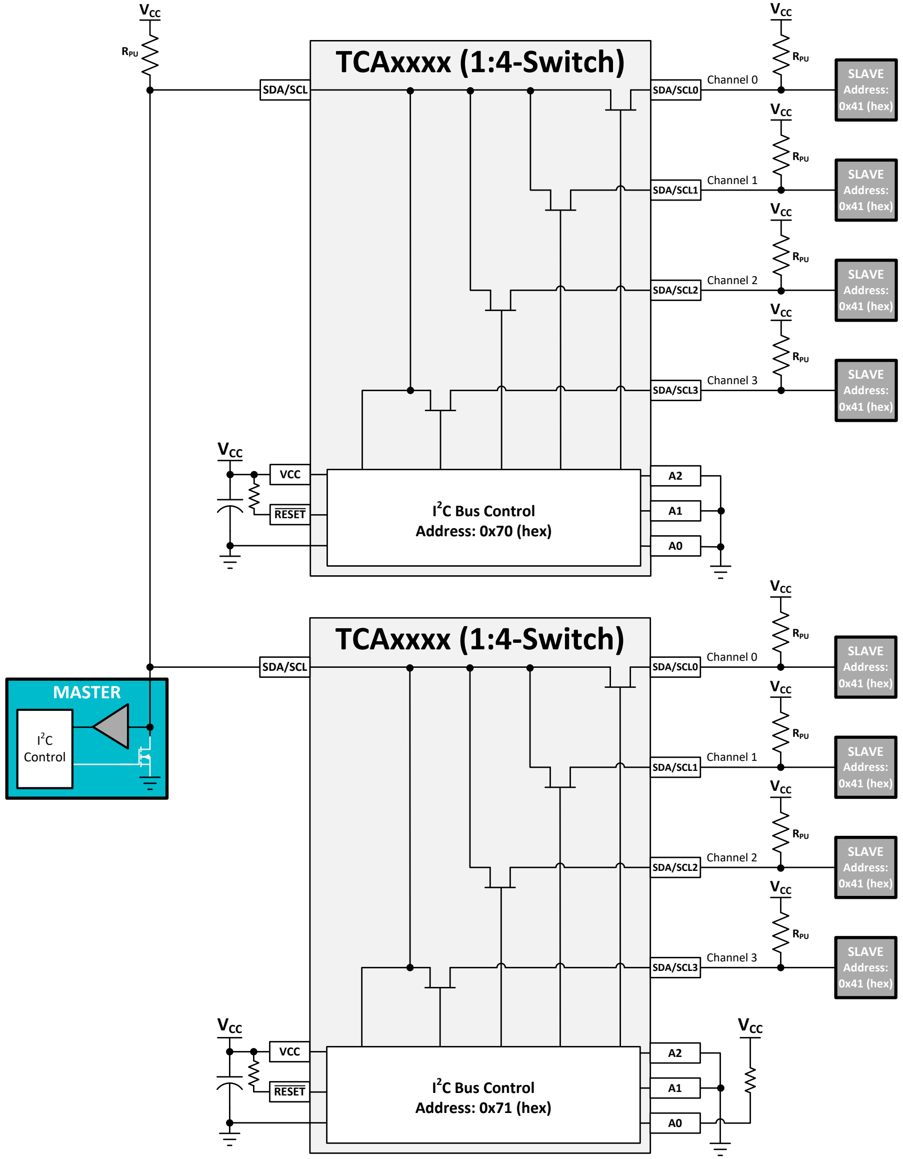
Design Considerations

Switches in Parallel

I^{2}C Devices: Muxes - Switches

Design Considerations- Switches in Parallel

- What happens when all channels enabled?

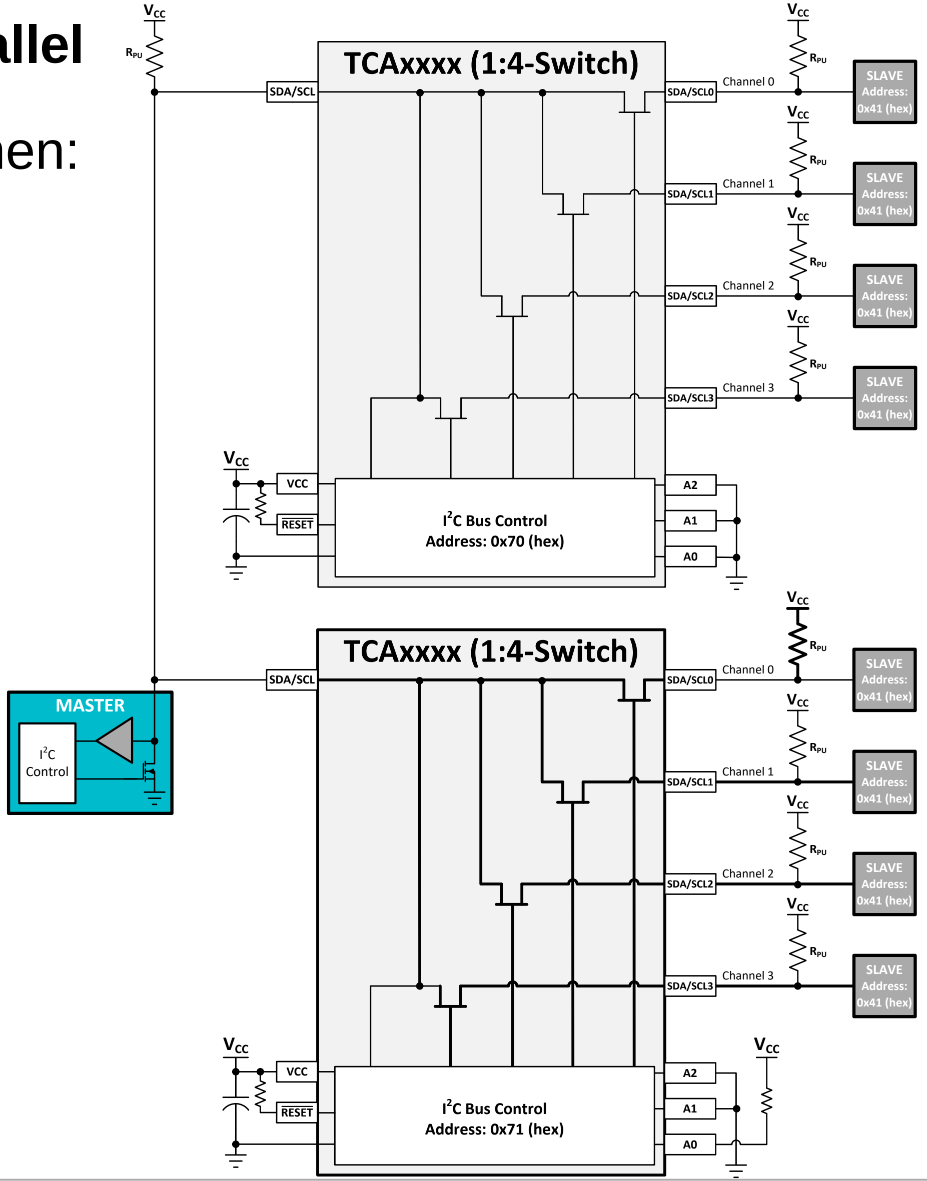


I²C Devices: Muxes - Switches

Design Considerations- Switches in Parallel

What happens when all channels enabled then:

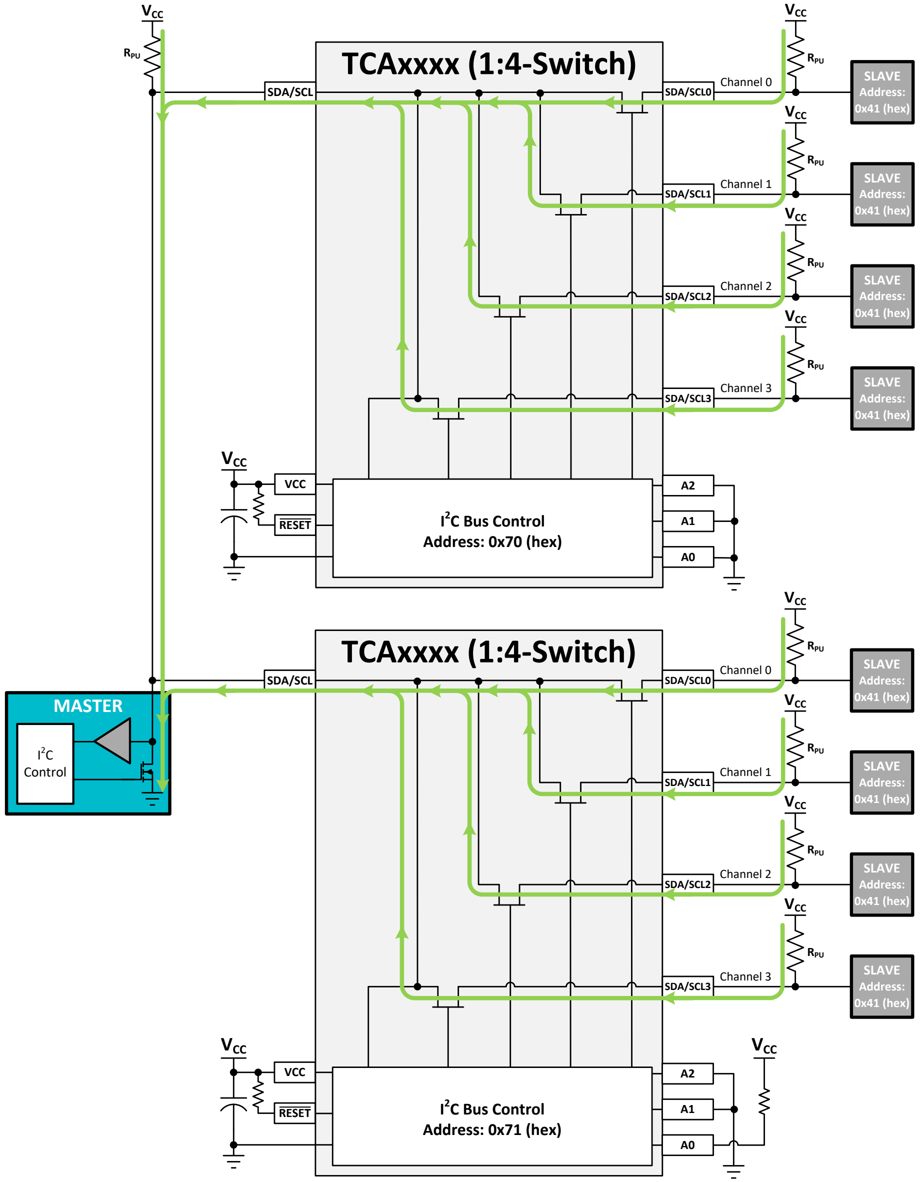
- *master* pulls low?
- *slave* pulls low?



I²C Devices: Muxes - Switches

Design Considerations- Switches in Parallel

- What happens when all channels enabled?



I^{2}C Devices: Muxes - Switches

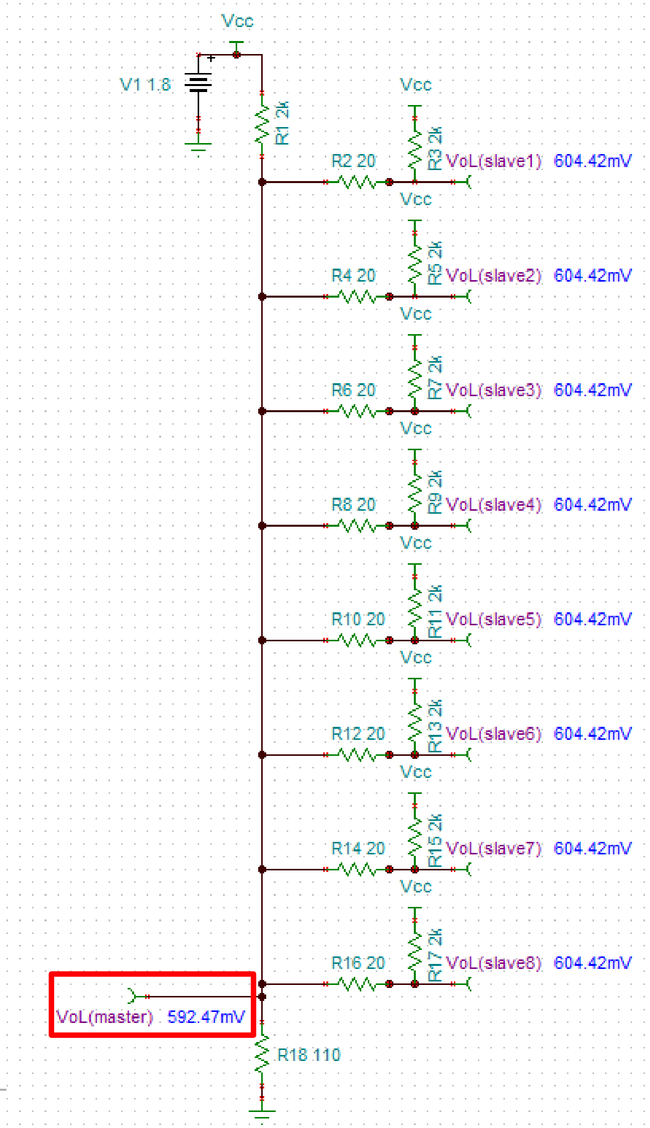
Design Considerations- Switches in Parallel

Simulation

Assumptions for this example:	
Pull up resistors	2k ohms
Master’s Ron	110 ohms
Pull up voltages	1.8V
Rds of all channels	20 ohms

Problem:

VoL(master)>ViL(master)



I^2C Devices: Muxes - Switches

Design Considerations- Switches in Parallel

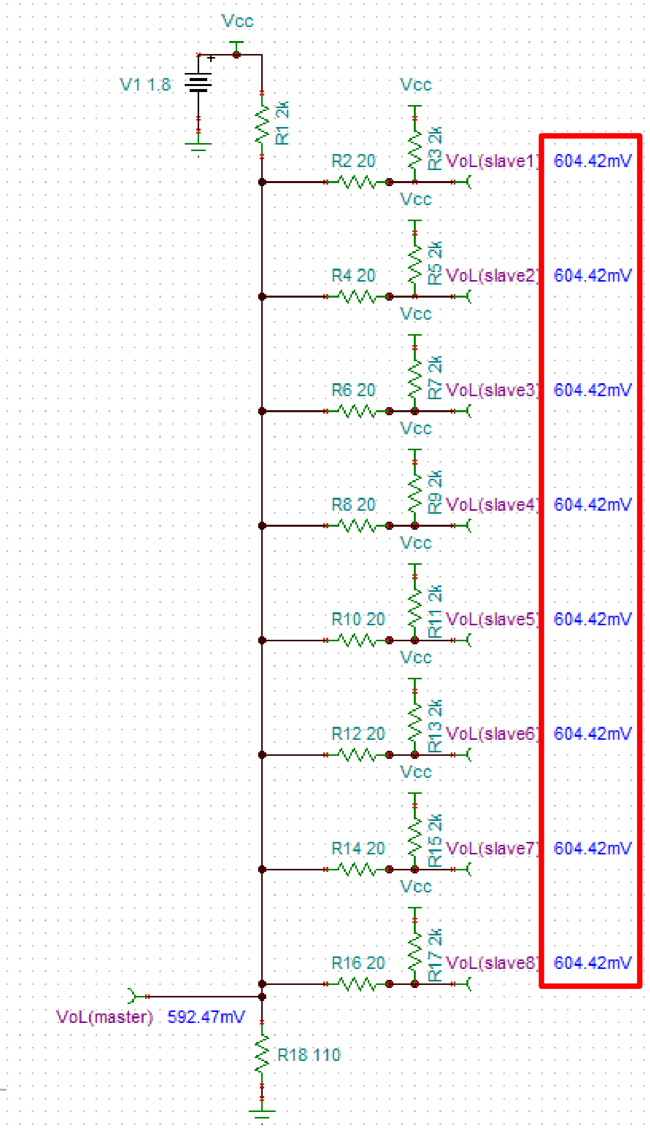
Simulation

Assumptions for this example:	
Pull up resistors	2k ohms
Master's Ron	110 ohms
Pull up voltages	1.8V
Rds of all channels	20 ohms

Problem:

VoL(master)>ViL(master)

V(slave)>ViL(slave)

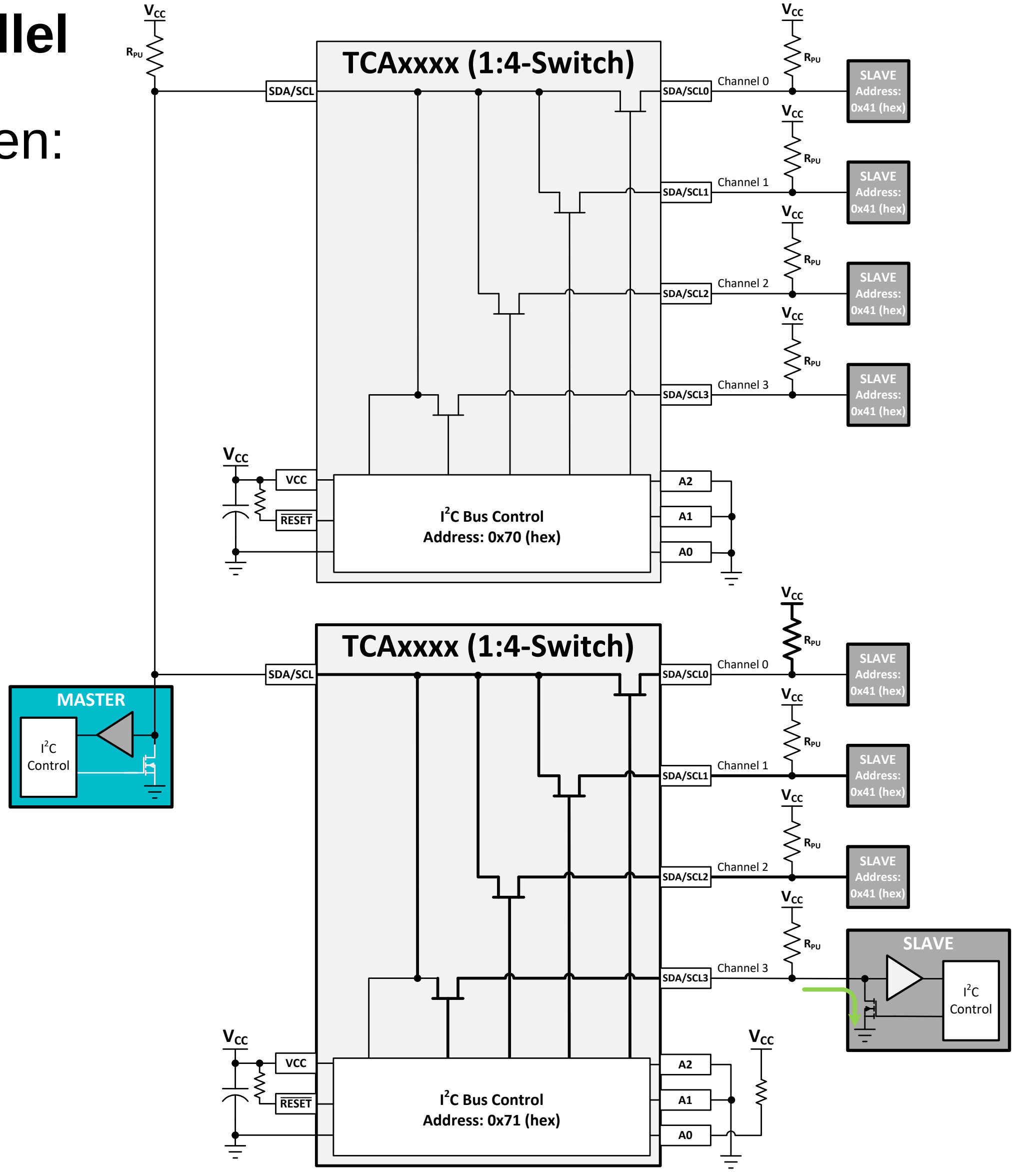


I^{2}C Devices: Muxes - Switches

Design Considerations- Switches in Parallel

What happens when all channels enabled then:

- *master* pulls low?
- *slave* pulls low?



I^2C Devices: Muxes - Switches

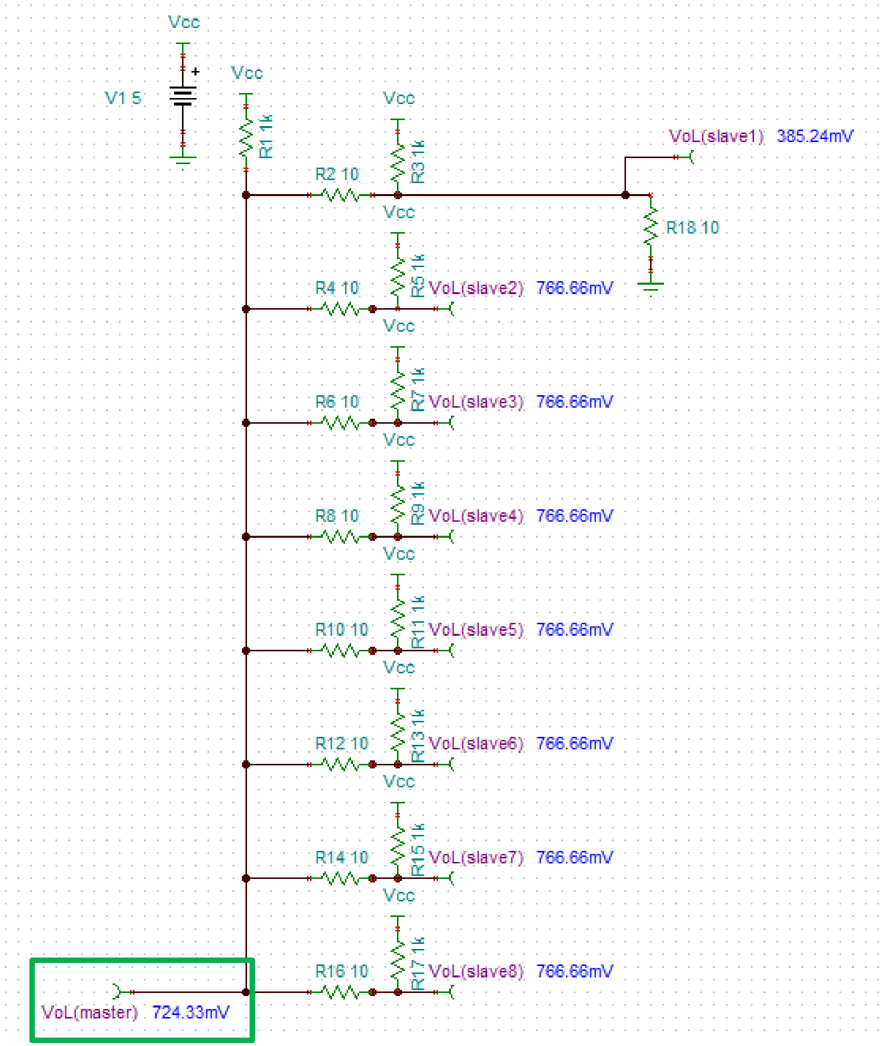
Design Considerations- Switches in Parallel

Simulation

Assumptions for this example:	
Pull up resistors	1k ohms
Slave’s Ron	10 ohms
Pull up voltages	5V
Rds of all channels	10 ohms

ViL<=1.5V to be valid

VoL(master)= 0.724 V



I^{2}C Devices: Muxes - Switches

Design Considerations- Switches in Parallel

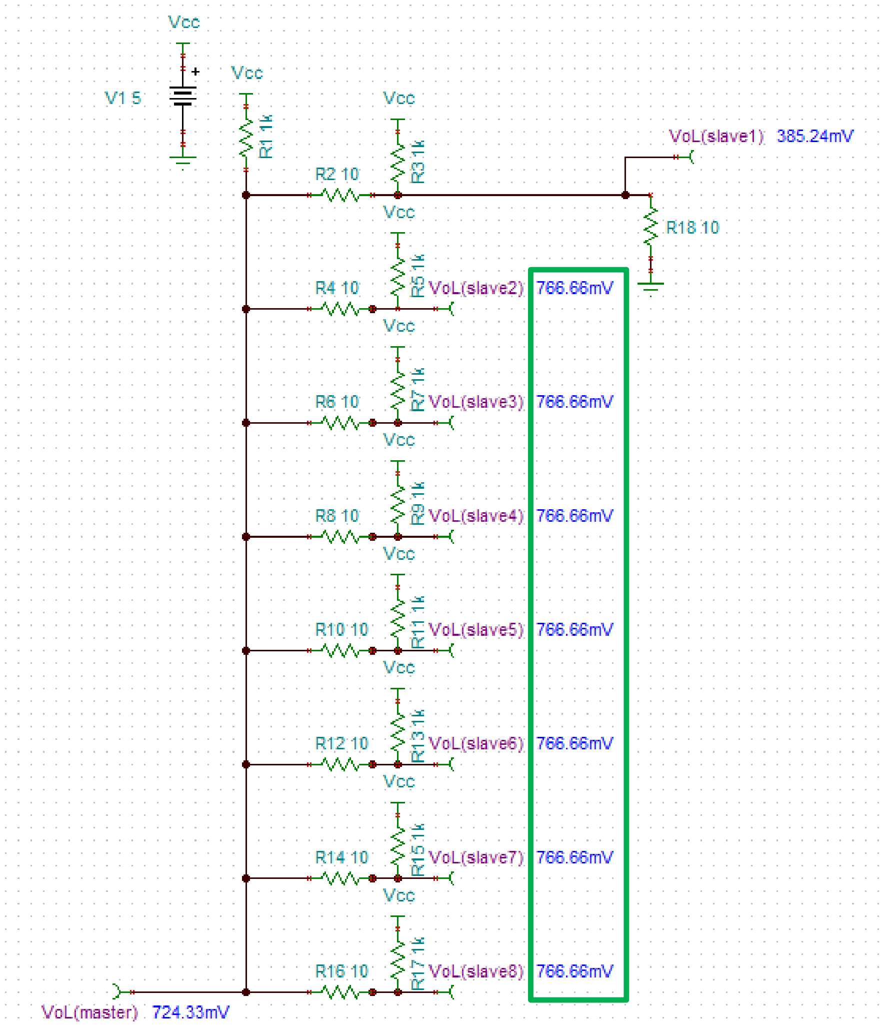
Simulation

Assumptions for this example:	
Pull up resistors	1k ohms
Slave's Ron	10 ohms
Pull up voltages	5V
Rds of all channels	10 ohms

ViL<=1.5V to be valid

VoL(master)= 0.724 V

Vmax(slave) = 0.767 V



I^{2}C Devices: Muxes - Switches

Design Considerations- Switches in Parallel

Simulation

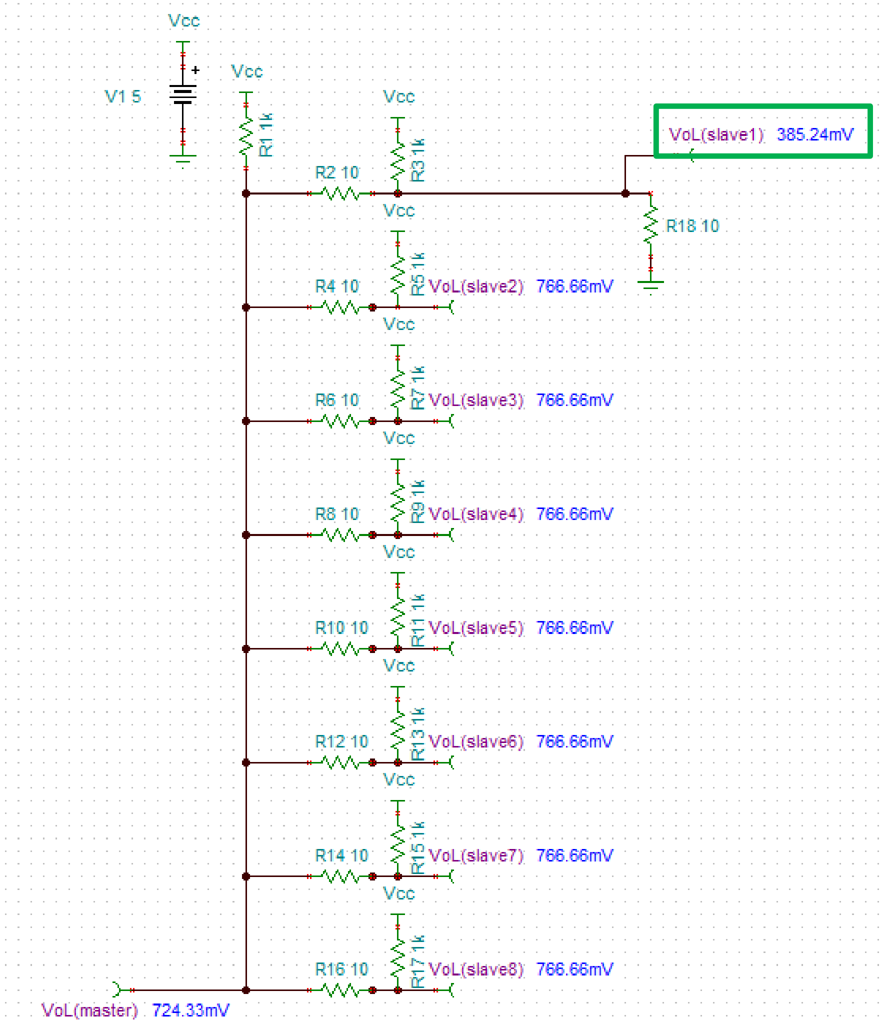
Assumptions for this example:	
Pull up resistors	1k ohms
Slave's Ron	10 ohms
Pull up voltages	5V
Rds of all channels	10 ohms

ViL<=1.5V to be valid

VoL(master) = 0.724 V

Vmax(slave) = 0.767 V

V(slave1) = 0.385 V



I^{2}C Devices: Muxes - Switches

Design Considerations- Switches in Parallel

Simulation

Assumptions for this example:	
Pull up resistors	1k ohms
Slave’s Ron	10 ohms
Pull up voltages	5V
Rds of all channels	10 ohms

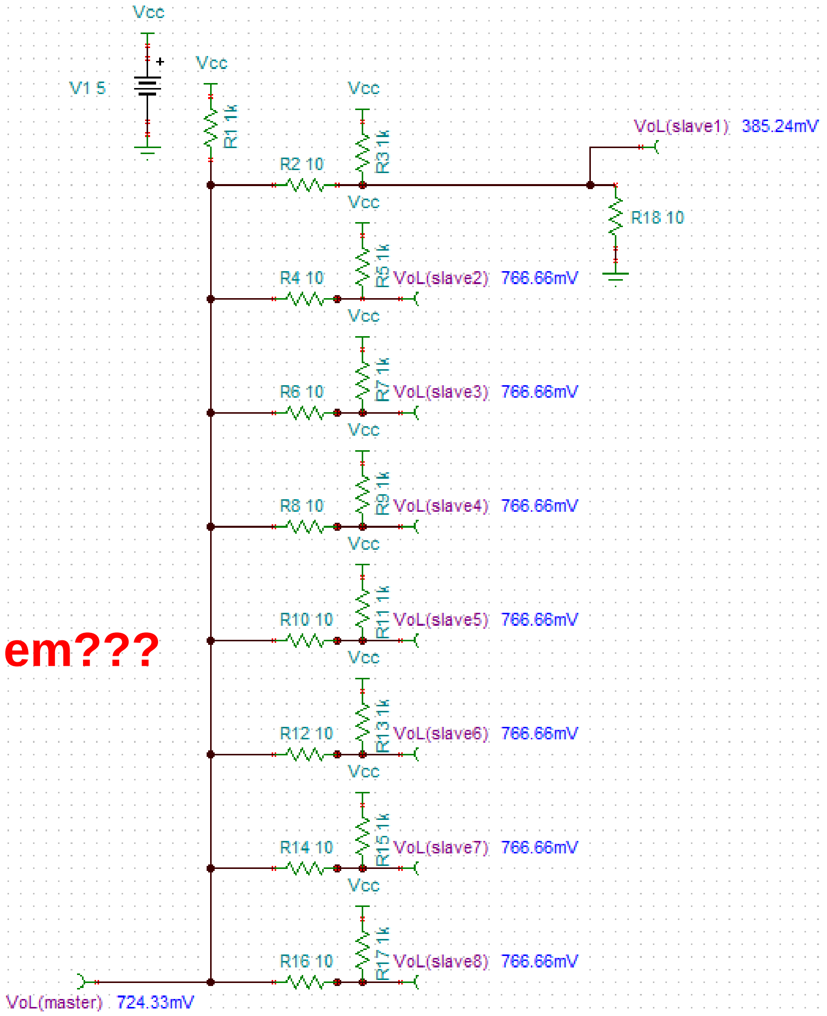
ViL<=1.5V to be valid

VoL(master) = 0.724 V

Vmax(slave) = 0.767 V

V(slave1) = 0.385 V

What is the problem???



I^{2}C Devices: Muxes - Switches

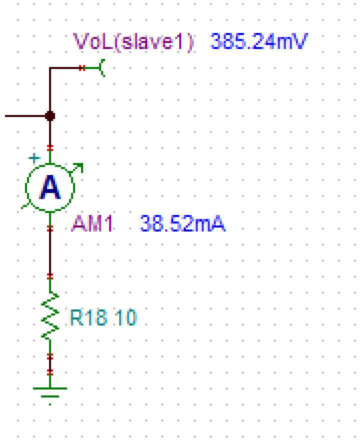
Design Considerations- Switches in Parallel

Recall:

V(slave1) = 0.385V

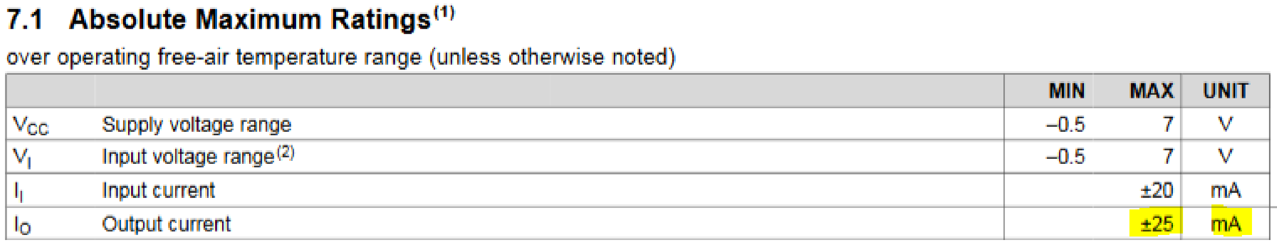
Ron(slave1) = 10 ohms

IoL = **38.5mA**



Concern:

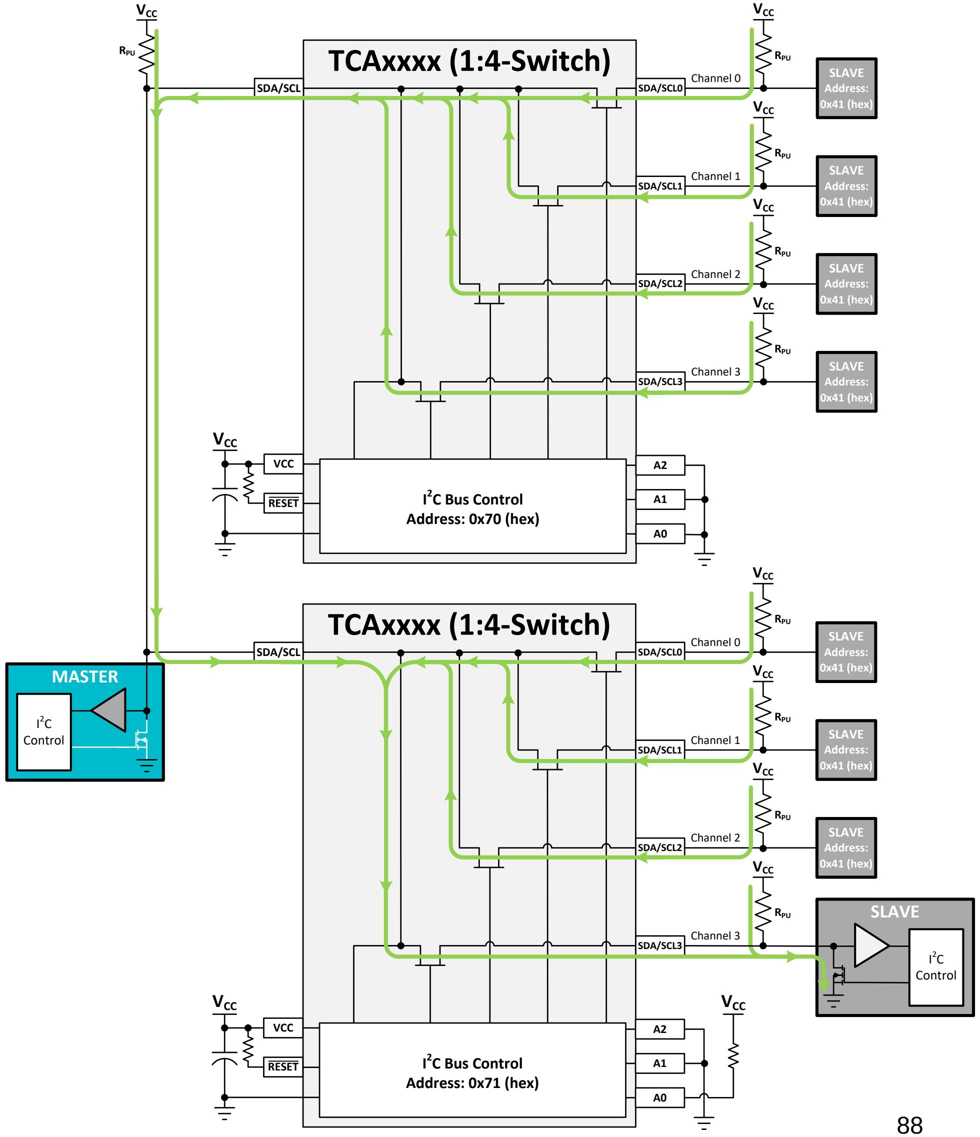
- Can Slave’s pull down FET handle this much current?
  - Let’s assume yes……this is typically a lot and usually cannot
- Can our switch handle this?
  - Remember: all this current flows through ONE pass FET



7.1 Absolute Maximum Ratings[(1)]

over operating free-air temperature range (unless otherwise noted)

		MIN	MAX	UNIT
V_{CC}	Supply voltage range	–0.5	7	V
V_I	Input voltage range[(2)]	–0.5	7	V
I_I	Input current		±20	mA
I_O	Output current		±25	mA



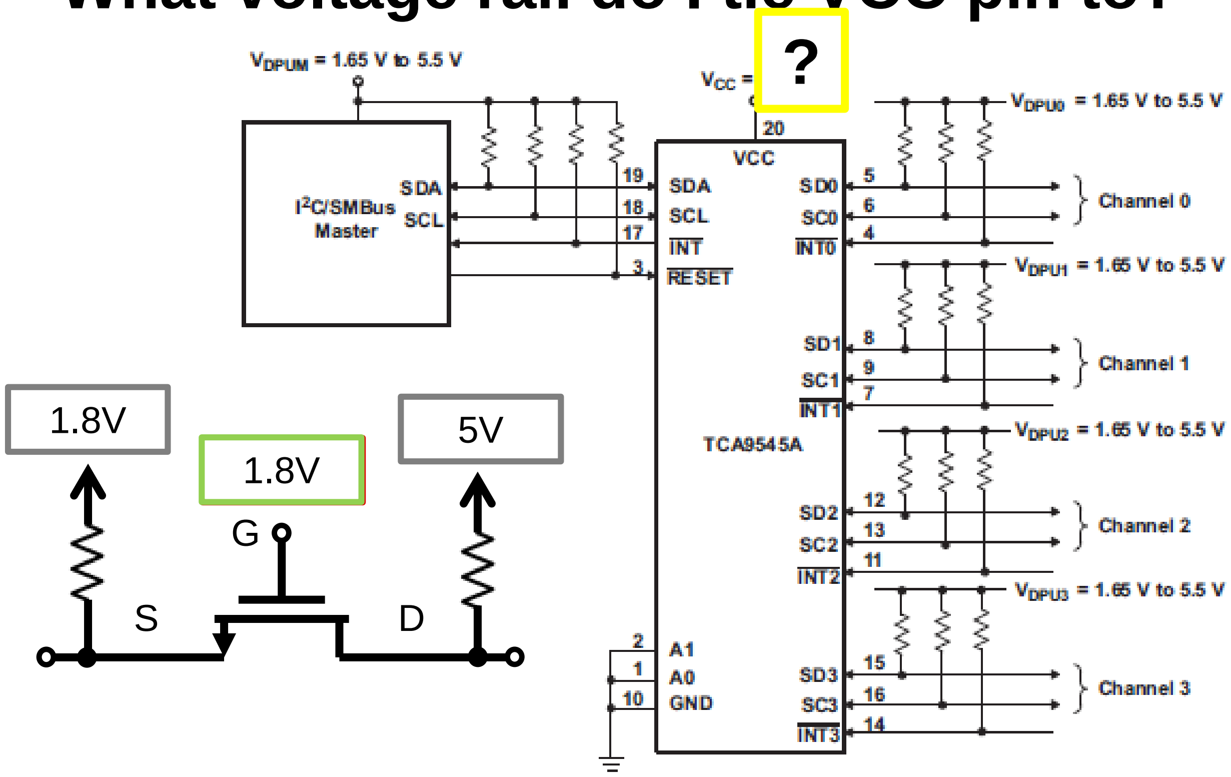
Switches

Common Questions

Switches:

Common Questions

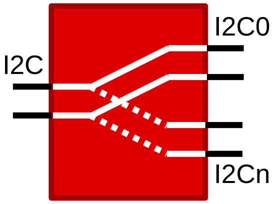
What voltage rail do I tie VCC pin to?



$$V_{CC} = [V_{DPUx}]_{min}$$

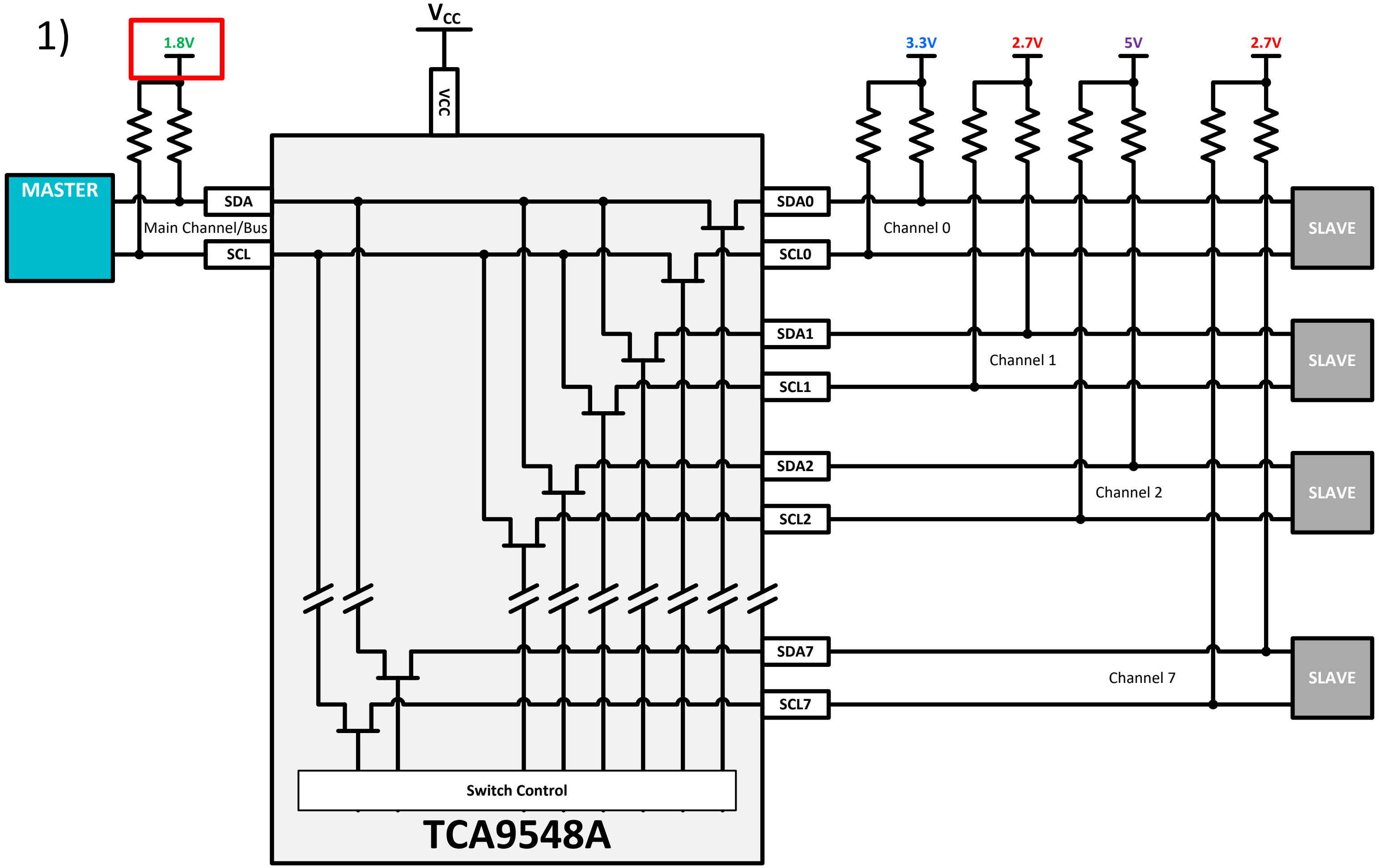
Switches:

Common Question: What voltage rail do I tie VCC pin to?



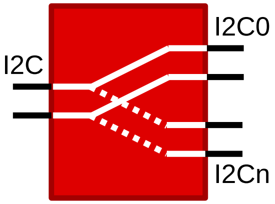
Step 1: Look to see what the lowest pull up voltage is

1)



Switches:

Common Question: What voltage rail do I tie VCC pin to?



Step 2: Vpass = lowest pull up voltage so use this to find what the MAX Vcc value can be

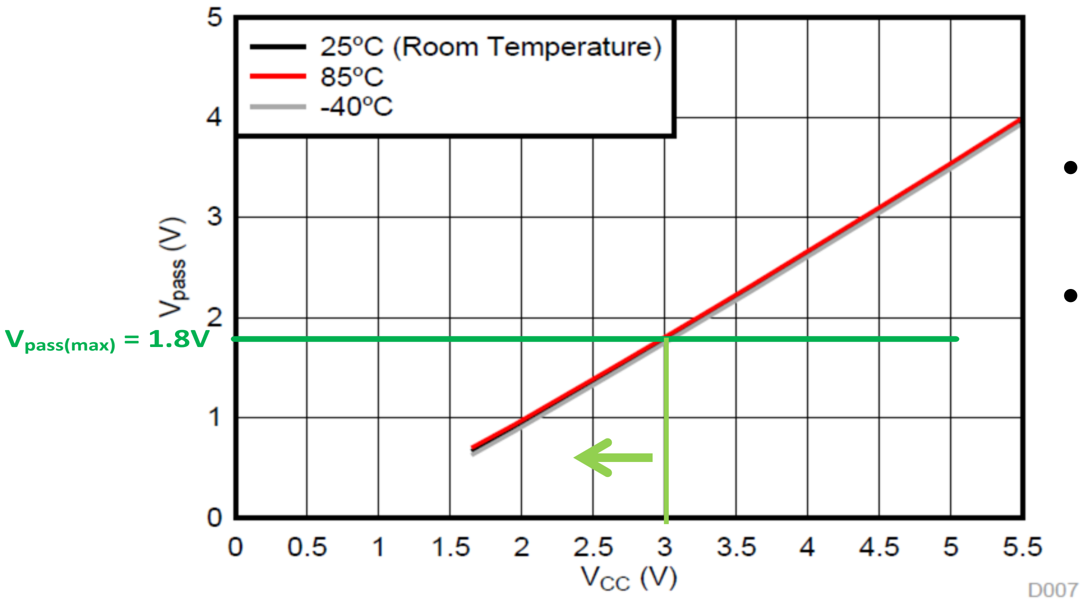
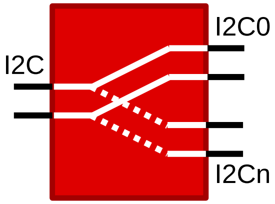


Figure 14. Pass-Gate Voltage (V_{pass}) vs Supply Voltage (V_{CC}) at Three Temperature Points

- In this example it appears Vcc can be 3V or lower
- But is that the case?

Switches:

Common Question: What voltage rail do I tie VCC pin to?



Step 3: If the pulling voltage on the primary lines is the lowest..........

Check if Vpullup_mainbus>=Vcc(max)*0.7

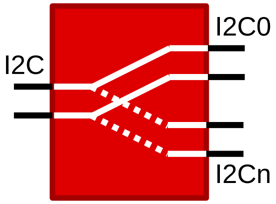
In this case Vcc(max) can only be about 2.57V

- If we chose 3V like in step 2 then our device would never see ViH and would likely treat the I2C bus as low

- Step 3 is necessary to ensure our device can communicate to the I2C bus

Switches:

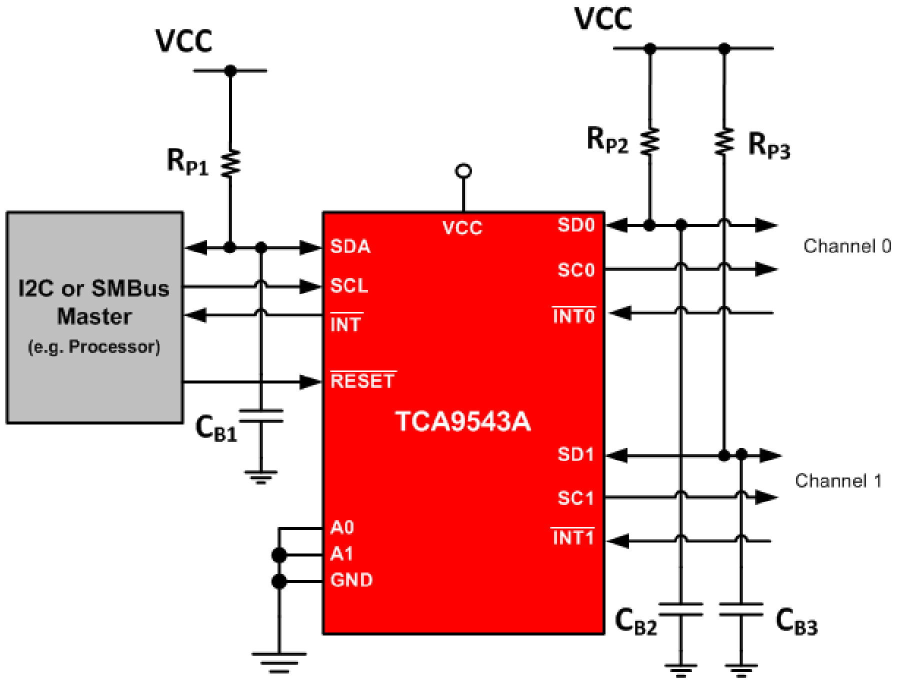
Common Questions



- Can I turn on all channels simultaneously?
  - Need to double check if capacitance on bus does not violate I2C standards
  - Ensure address conflicts do not occur if channels are all enabled
  - Be sure the VoL of master and slaves are at acceptable ViL
  - **Calculate the current through the master/pass FET channel and ALL slave pull-up resistors when the master/slave drives low. It may be very high.**

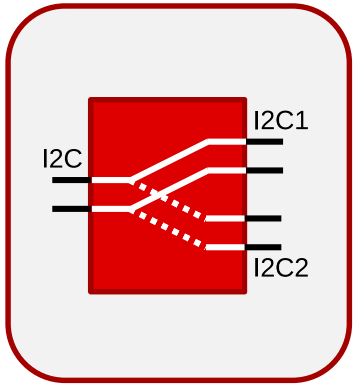
Pull-up resistor calculation for switches

Channel selected	Equivalent pull-up resistance
None of channels are ON	$R_{P1}(\max) = \dfrac{t_r}{0.8473 \times C_{B1}}$
Channel 1 ON	$(R_{P1} \parallel R_{P2})\max = \dfrac{t_r}{0.8473 \times (C_{B1} \parallel C_{B2})}$
Both Channel 1 & 2 ON	$(R_{P1} \parallel R_{P2} \parallel R_{P3})\max = \dfrac{t_r}{0.8473 \times (C_{B1} \parallel C_{B2} \parallel C_{B3})}$



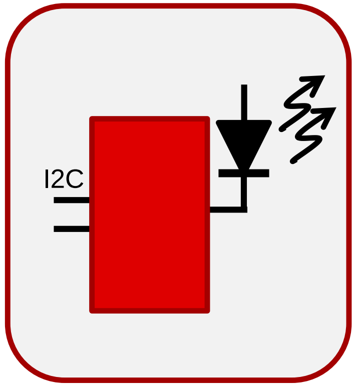
Production
Development

Switches

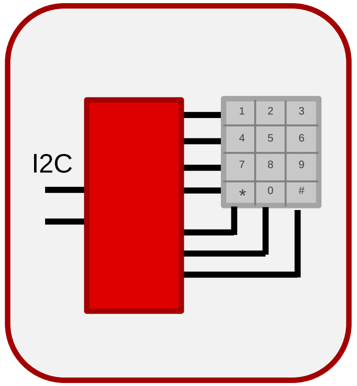


Parameter / Feature	Unit	TCA9543A	TCA9546A	TCA9544A	TCA9545A	TCA9548A
Function		Switch	Switch	Switch	Switch	Switch
Number of channels		2	4	4	4	8
Minimum supply voltage	(V)	1.65	1.65	1.65	1.65	1.65
Maximum supply voltage	(V)	5.5	5.5	5.5	5.5	5.5
Level Translation		●	●	●	●	●
/RESET Pin		●	●		●	●
Interrupt Output		●		●	●	
I2C-bus	(kHz)	400	400	400	400	400

Special function



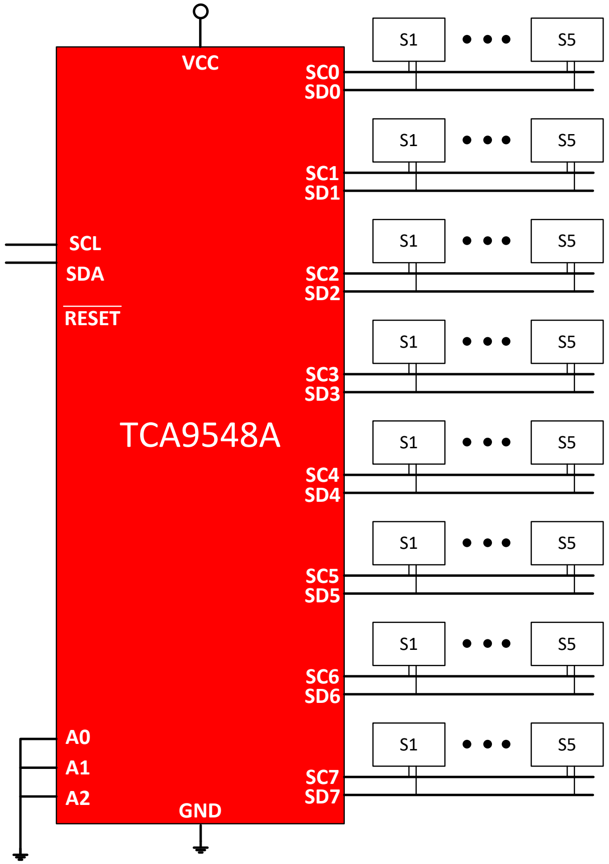
Parameter / Feature	Unit	TCA6507	TCA8418E	TCA8424
Function		LED driver	Key board Controller	Key board Controller
Number of channels		7	18	24
Minimum supply voltage	(V)	1.65	1.65	1.65
Maximum supply voltage	(V)	3.6	3.6	3.6
/RESET Pin			●	
Interrupt Output			●	●
Internal pull-up resistor	(kΩ)			100
I2C-bus	(kHz)	400	400	1000

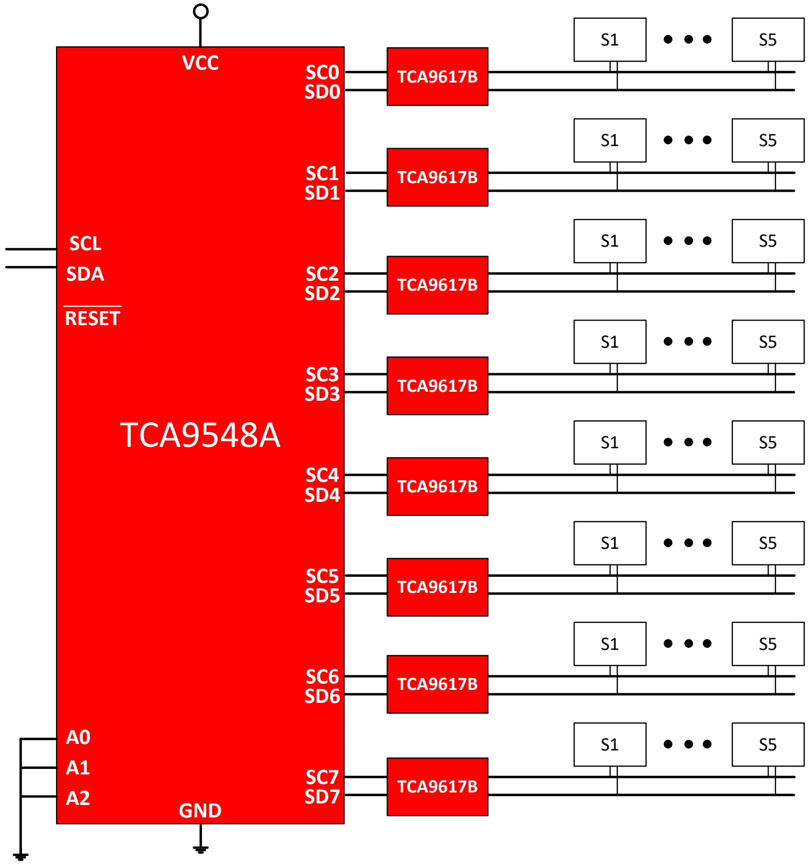


Switch with Too Many Slaves

Repeaters may need to be added to the output channels of the Switch to improve signal integrity of the I^2C signal

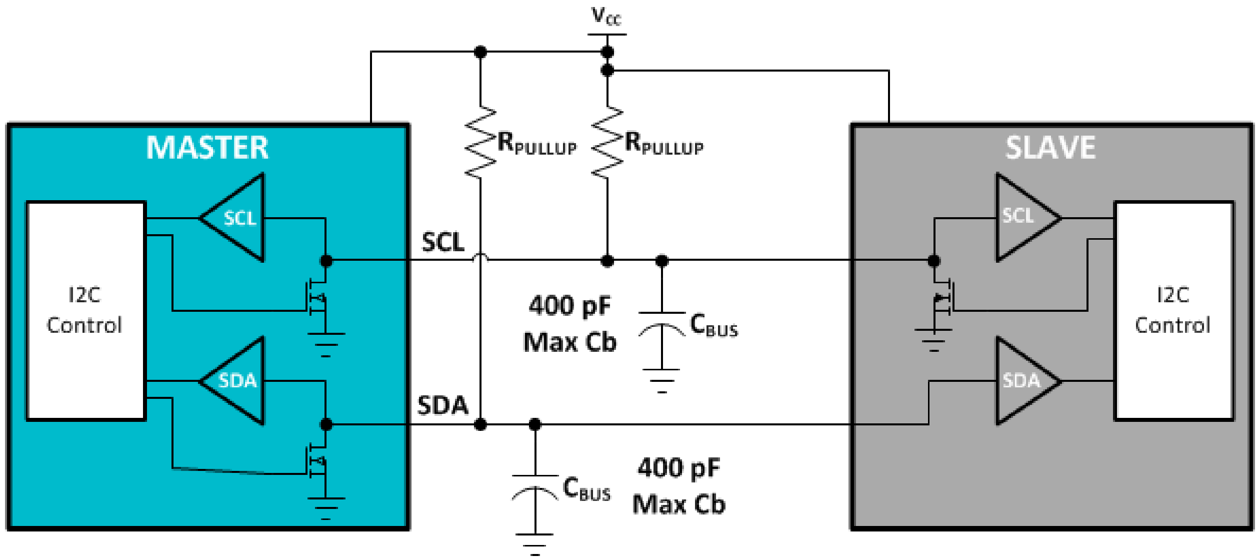
If all channels are turned on at the same time then adding a buffer may be useful





Buffers

I^{2}C Devices: Buffers: Introduction



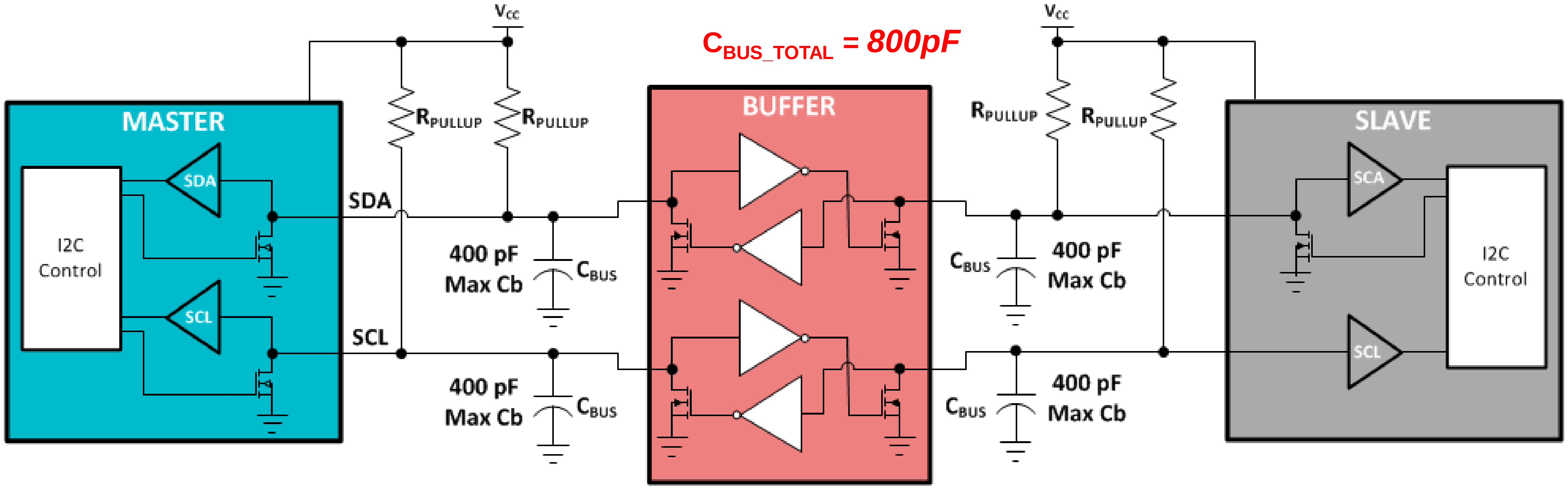
Why do we need buffers?

Frequency	Maximum Bus Capacitance allowed
100kHz	400pF
400kHz	400pF
1MHz	550pF

I²C Devices: Buffers:

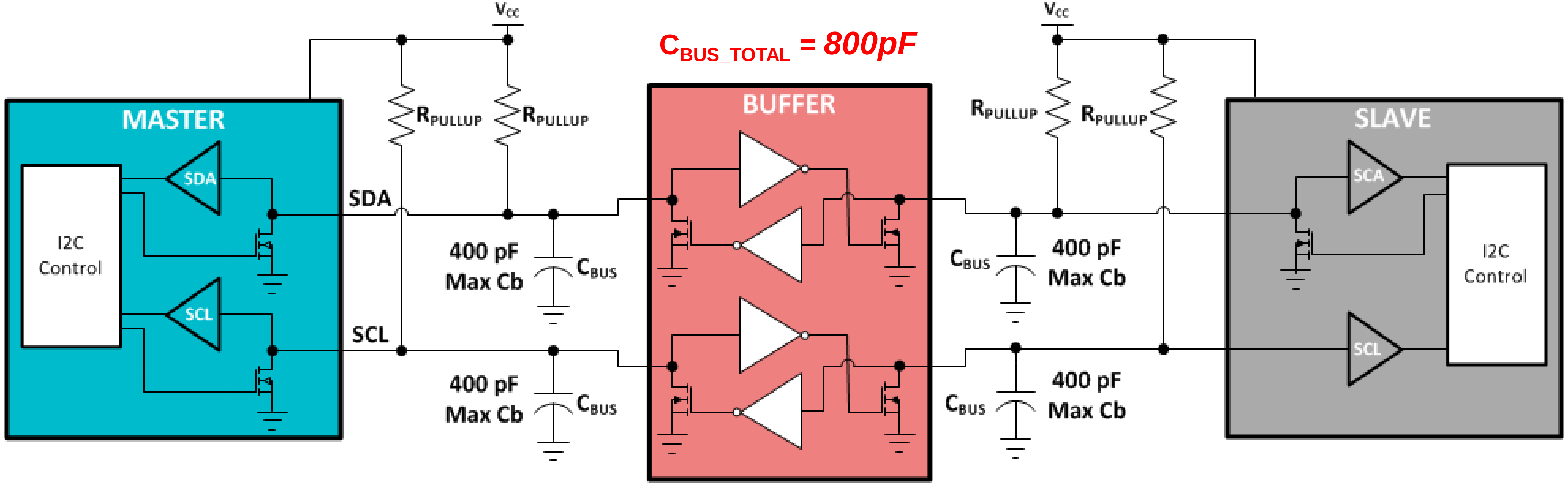
Introduction

Example



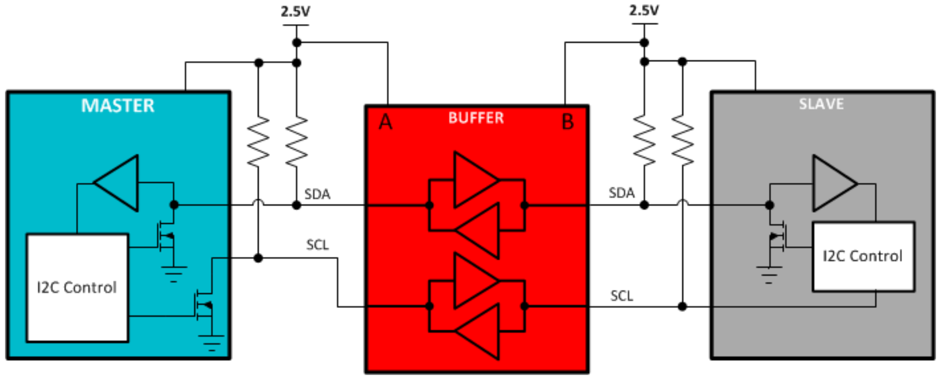
I^{2}C Devices: Buffers: Theory of Operation

There's a problem with this circuit, do you see it?

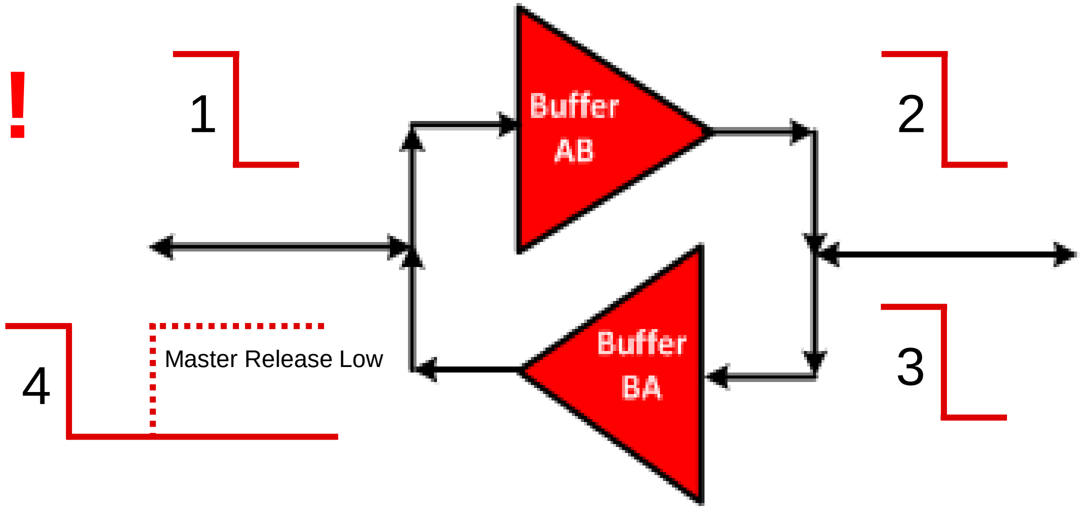


I^{2}C Devices: Buffers: Theory of Operation

Here is simplified version the buffer and we will consider only one bus because the are identical. Lets start with the system in the IDLE state and then have the Master pull low, what happens?



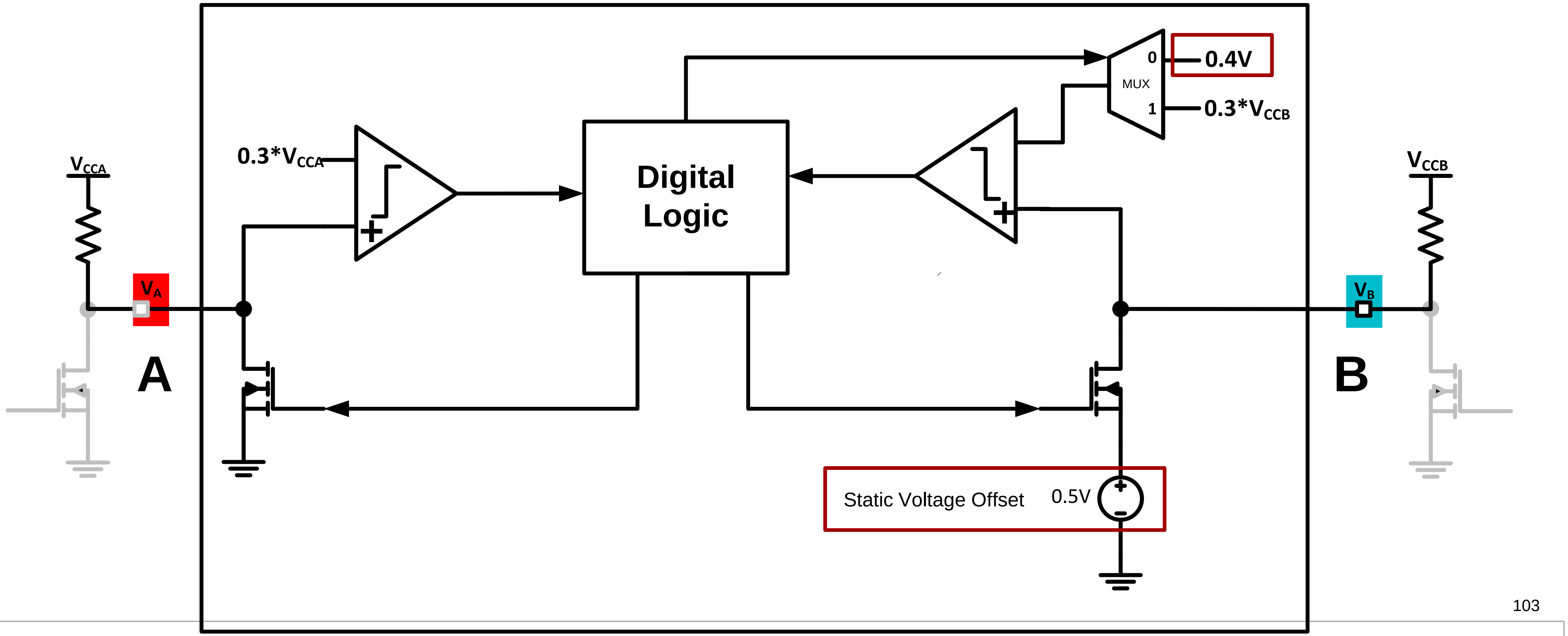
THE BUS LOCKS UP!!!



Who is controlling the LOW?

I^{2}C Devices: Buffers: Theory of Operation

We need a method to determine which side of the buffer is generating the low so that the buffer does not get locked up. Enter, the Static Voltage Offset.



Buffers

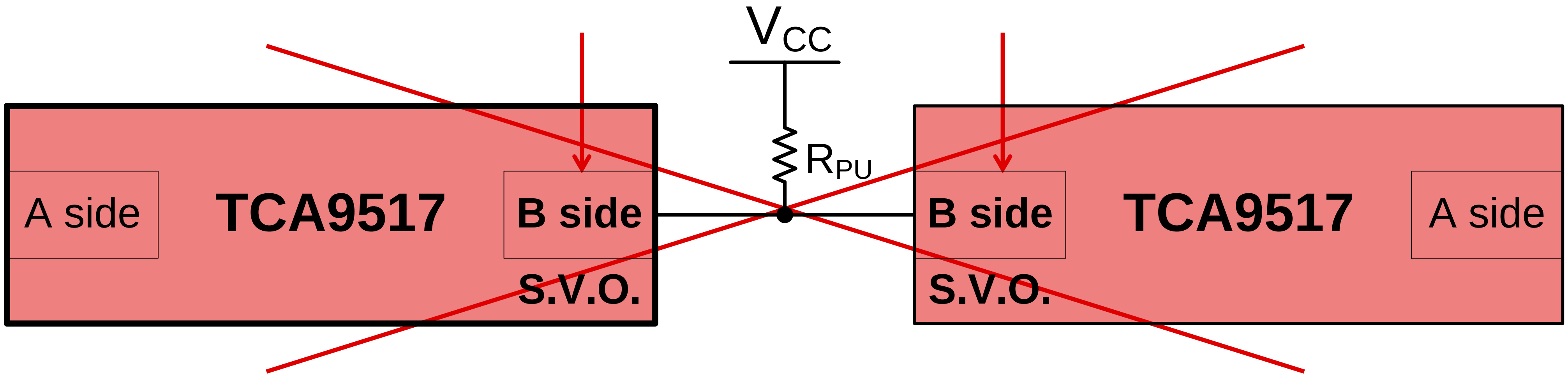
Design Considerations

I^2C Devices: Buffers:

Design considerations

Understanding how TCA9517 works now tells us:

- NOT to connect static voltage offset (SVO) sides to each other (B side to B side)

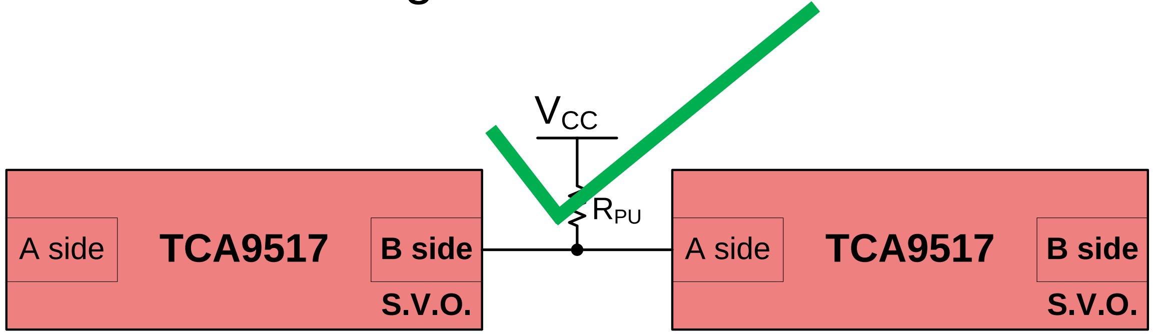


I^{2}C Devices: Buffers:

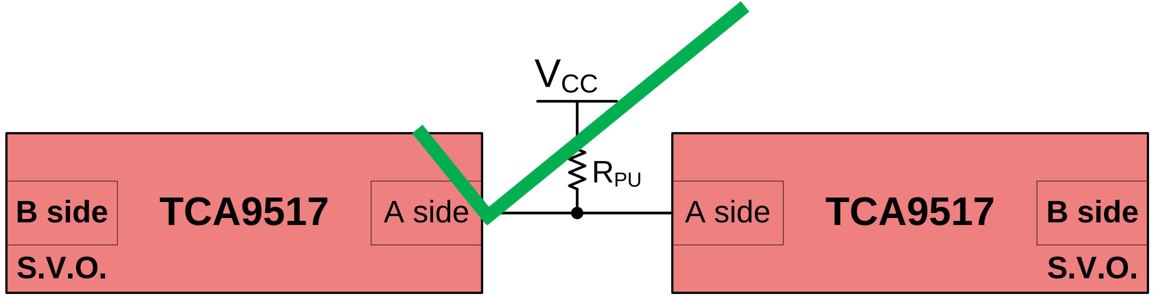
Design considerations

Understanding how TCA9517 works now tells us:

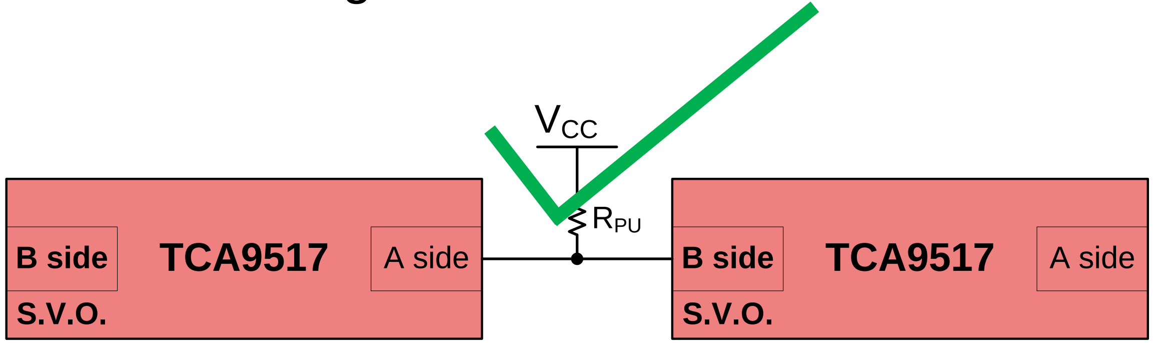
Connecting B side to A side is OKAY



Connecting A side to A side is OKAY



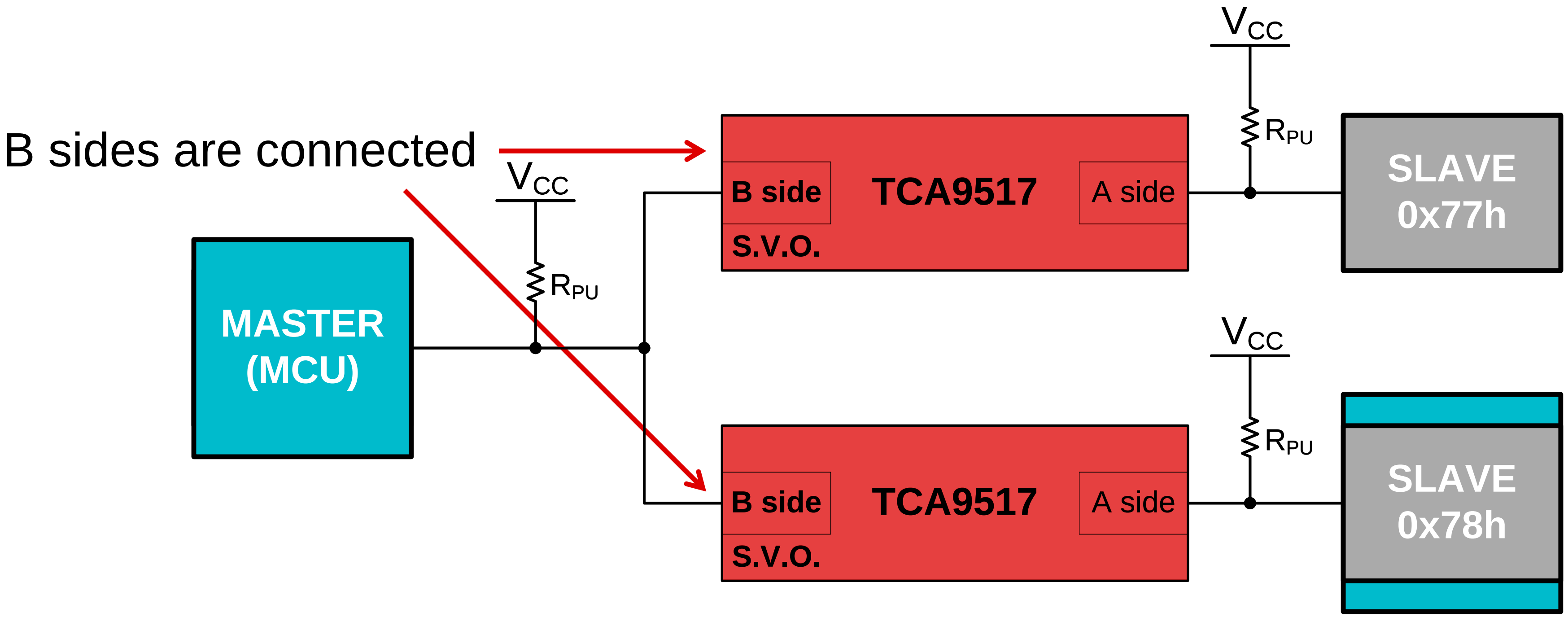
Connecting A side to B side is OKAY



I^{2}C Devices: Buffers:

Design considerations

What's wrong with this picture?



Most cases this is okay, but sometimes an old design is modified where the master and slave switch places

I^{2}C Devices: Buffers:

Design Considerations

Are there other buffer devices on the bus?

- Sides with static voltage offsets should not be connected to each other

Are there devices with low ViL?

- The Vos of buffers can sometimes be larger than ViL

Can you stack buffers in series?

- Yes but there are concerns with this

What are the considerations for using buffers in parallel?

- Offset sides should still not be connected

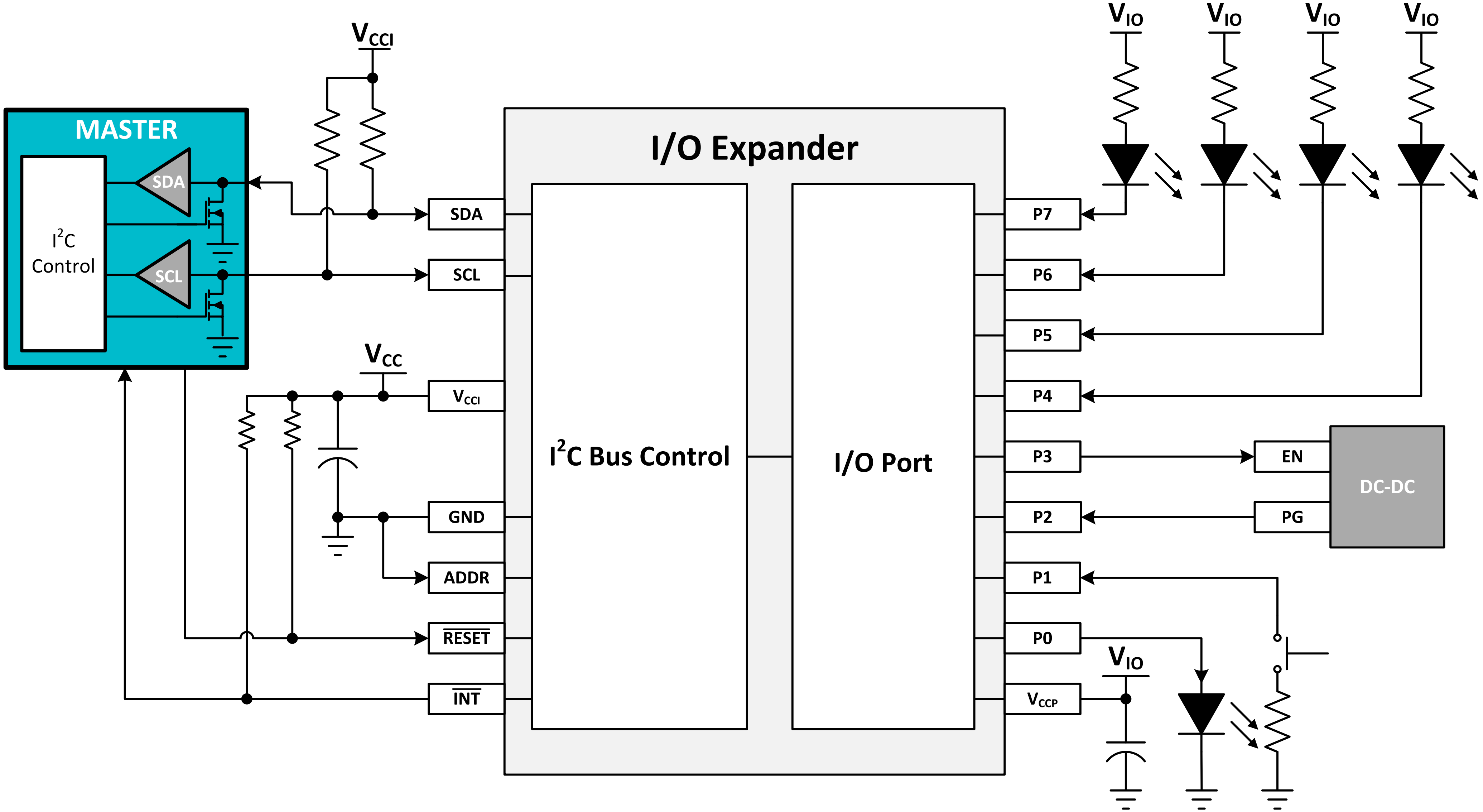
I^{2}C Devices: Buffers: TI Solutions

TCA9509	PCA9515B	TCA9517	TCA9517A	TCA9617B	TCA9617A	TCA4311A

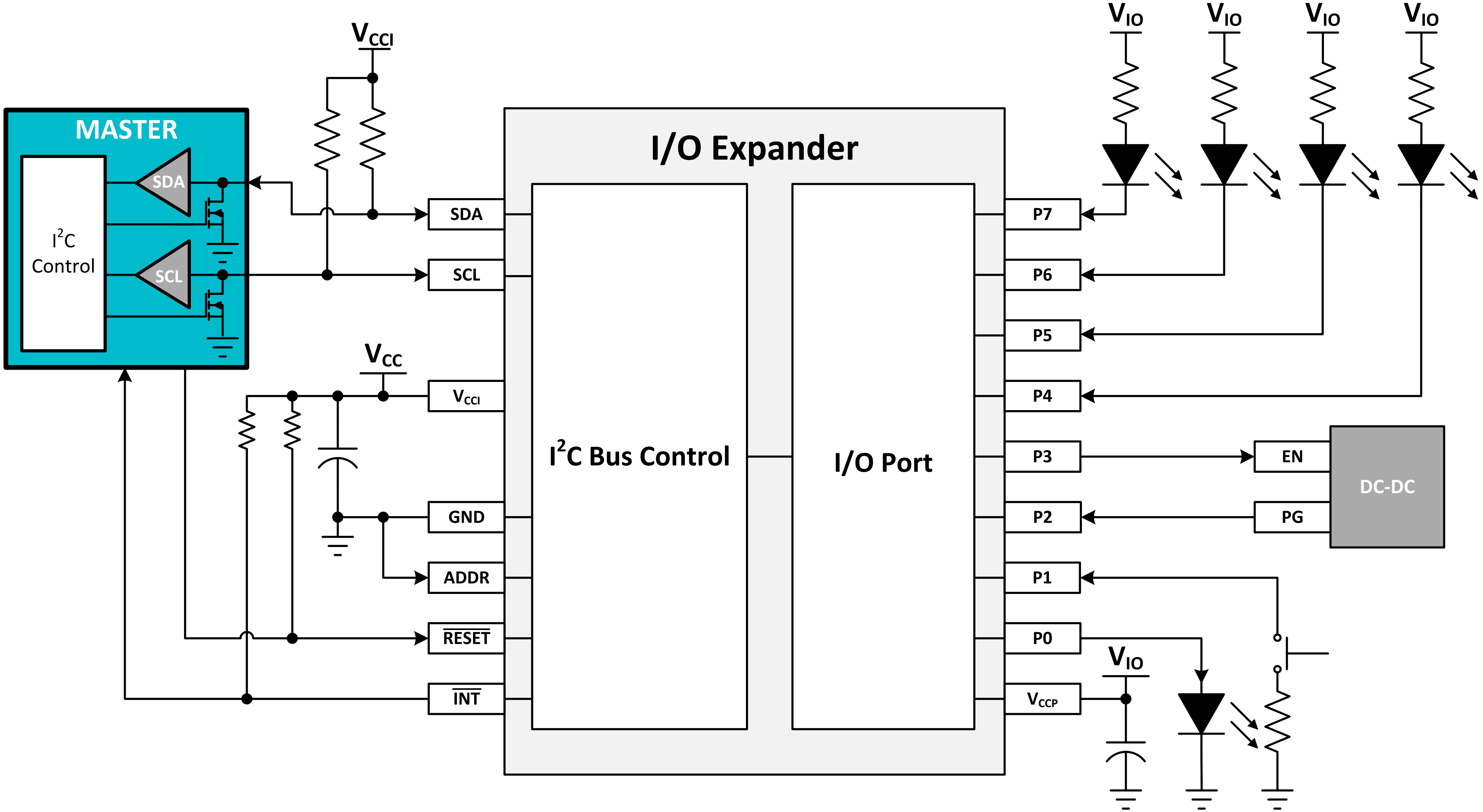
TCA9800	TCA9801	TCA9802	TCA9803	TCA9509

I/O Expanders

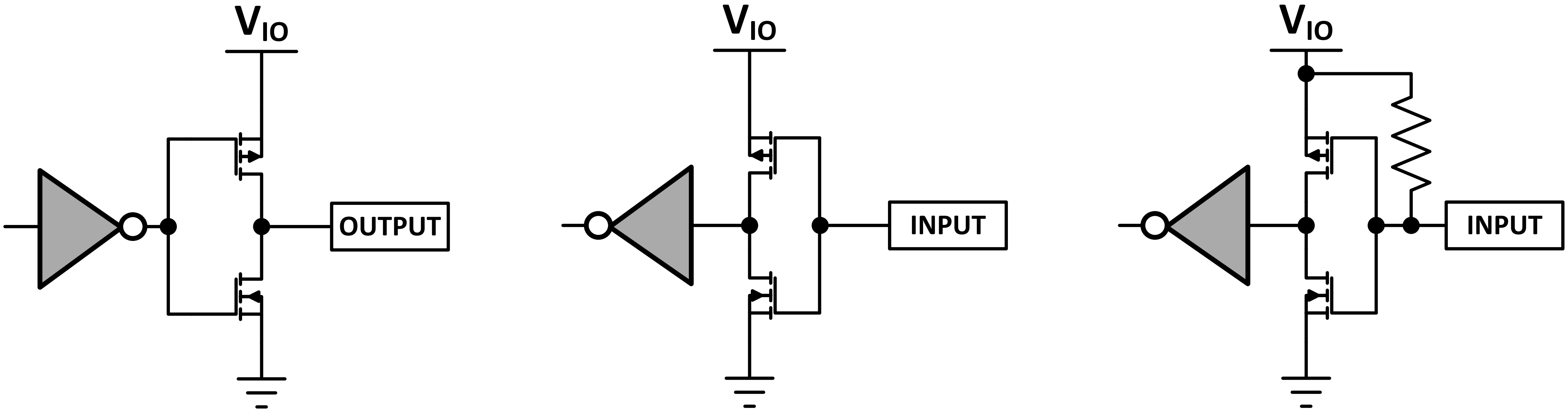
I^2C Devices: I/O Expanders: Introduction



I²C Devices: I/O Expanders: Introduction



I^{2}C Devices: I/O Expanders: Introduction



Other FAQs on IO expanders

Do the address pins have to pulled HIGH or LOW using a resistor?

- It is not a requirement to use a resistor for pulling HIGH or LOW. However, a resistor can be used.

Does TI offer IO expander with stuck bus recovery?

- We do not have an IO expander with stuck bus recovery feature today. The master need to issue as Reset command or power cycle

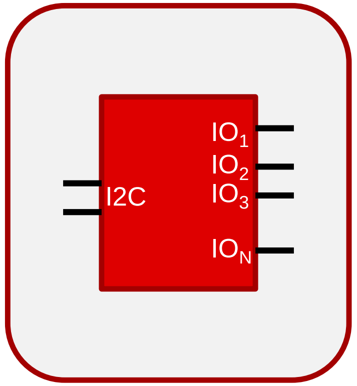
What is the maximum frequency that the P-port toggle at?

- To write to a port you need to send 3 bytes plus the 3 ACK’s and the Start and Stop. So the maximum switching frequency of the P-ports can be ~15 kHz

What is the maximum current you can sink through an 8-bit IO expander

- The recommended current through each port is 10 mA, and the maximum through the IO Expander is 80 mA

IO expanders



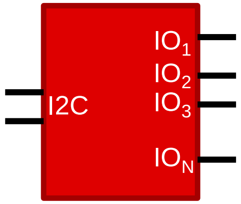
Production
Development

Parameter / Feature	Unit	PCA9536	TCA6408A	TCA7408	TCA9554	TCA9554A	TCA9534	TCA9534A	TCA9538
Number of bits		4	8	8	8	8	8	8	8
Minimum supply voltage	(V)	2.3	1.65	1.65	1.65	1.65	1.65	1.65	1.65
Maximum supply voltage	(V)	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5
Level Translation			●	●					
/RESET Pin			●	●					●
Interrupt Output			●	●	●	●	●	●	●
Internal pull-up resistor	(kΩ)	100		100	100	100			
I2C-bus	(kHz)	400	400	400	400	400	400	400	400
I2C address	Binary	1000 001	0100 00x	1000 01x	0100 xxx	0111 xxx	0100 xxx	0111 xxx	1110 1xx

Parameter / Feature	Unit	PCA6107	PCA9557	TCA6416A	TCA9535	TCA9539	TCA9555	TCA6418E	TCA6424A
Number of bits		8	8	16	16	16	16	18	24
Minimum supply voltage	(V)	2.3	2.3	1.65	1.65	1.65	1.65	1.65	1.65
Maximum supply voltage	(V)	5.5	5.5	5.5	5.5	5.5	5.5	3.6	5.5
Level Translation				●					●
/RESET Pin		●	●	●		●		●	●
Interrupt Output		●		●	●	●	●	●	●
Internal pull-up resistor	(kΩ)						100	100	
I2C-bus	(kHz)	400	400	400	400	400	400	400	400
I2C address	Binary	0011 xxx	0111 xxx	0100 00x	0100 xxx	1110 1xx	0100 xxx	0100 00x	0100 01x

Thank you!

Common questions (#1)



Which IO expander do I need?

Do you need lower power consumption and have mostly output needs?

Do you need a different I2C slave address?

Do you need a hardware /RESET pin

Do you have many inputs distributed across the board?

Do you need a different I2C slave address?

Do you need voltage translation (e.g. to drive LEDs at 5 Volts from a 1.8V I2C bus)?