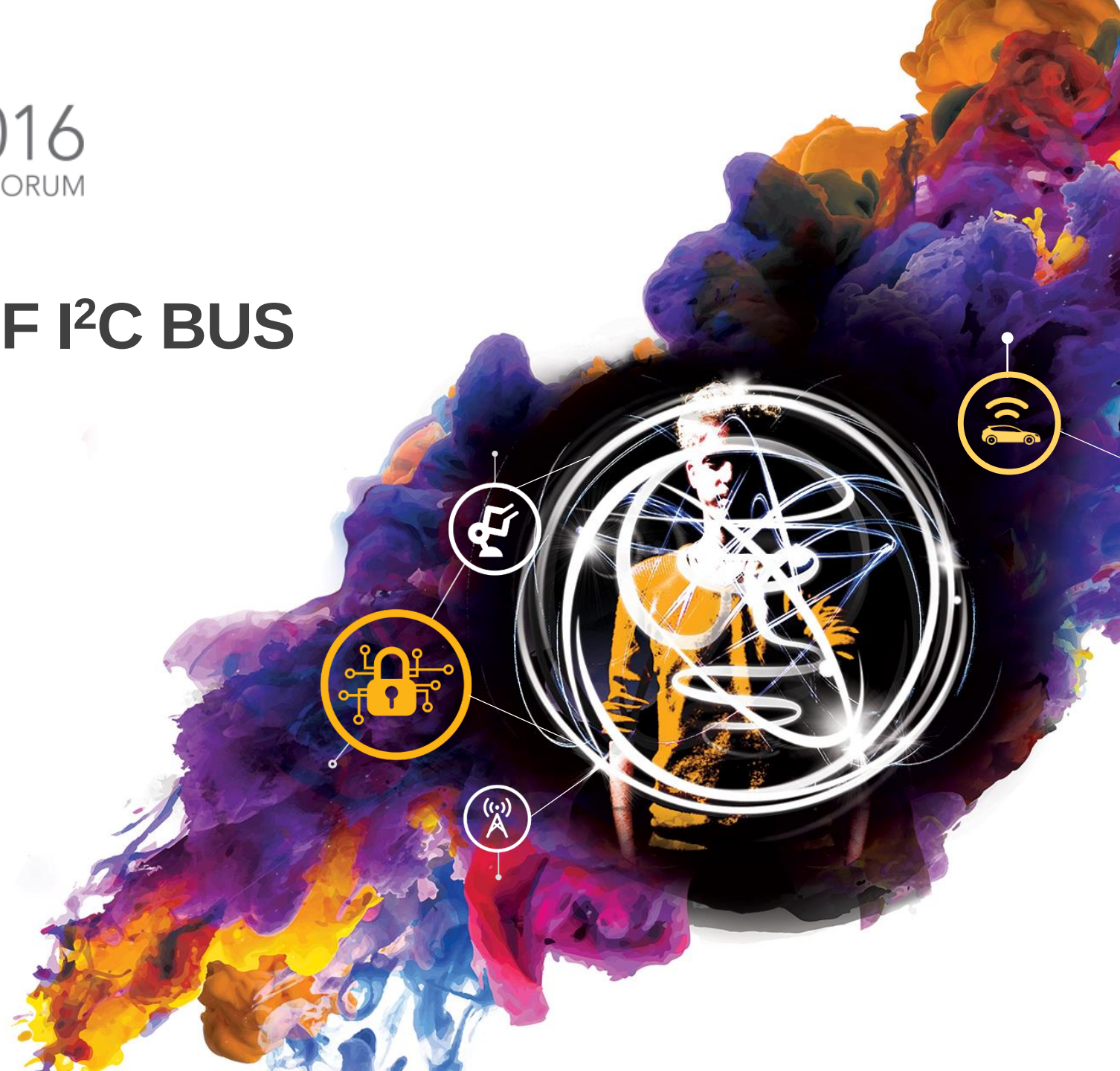
FTF 2016
TECHNOLOGY FORUM

THE INS AND OUTS OF I²C BUS BUFFERS

TF-DES-N1916

EMMANUEL NANA
FTF-DES-N1916
JUNE 7, 2016



AGENGA

- History of I²C BUS
- General I²C BUS Protocol and transaction
- What is a I²C BUS Buffer
- Why I²C BUS Buffer is needed
- How to use a I²C BUS Buffer

History of I²C BUS

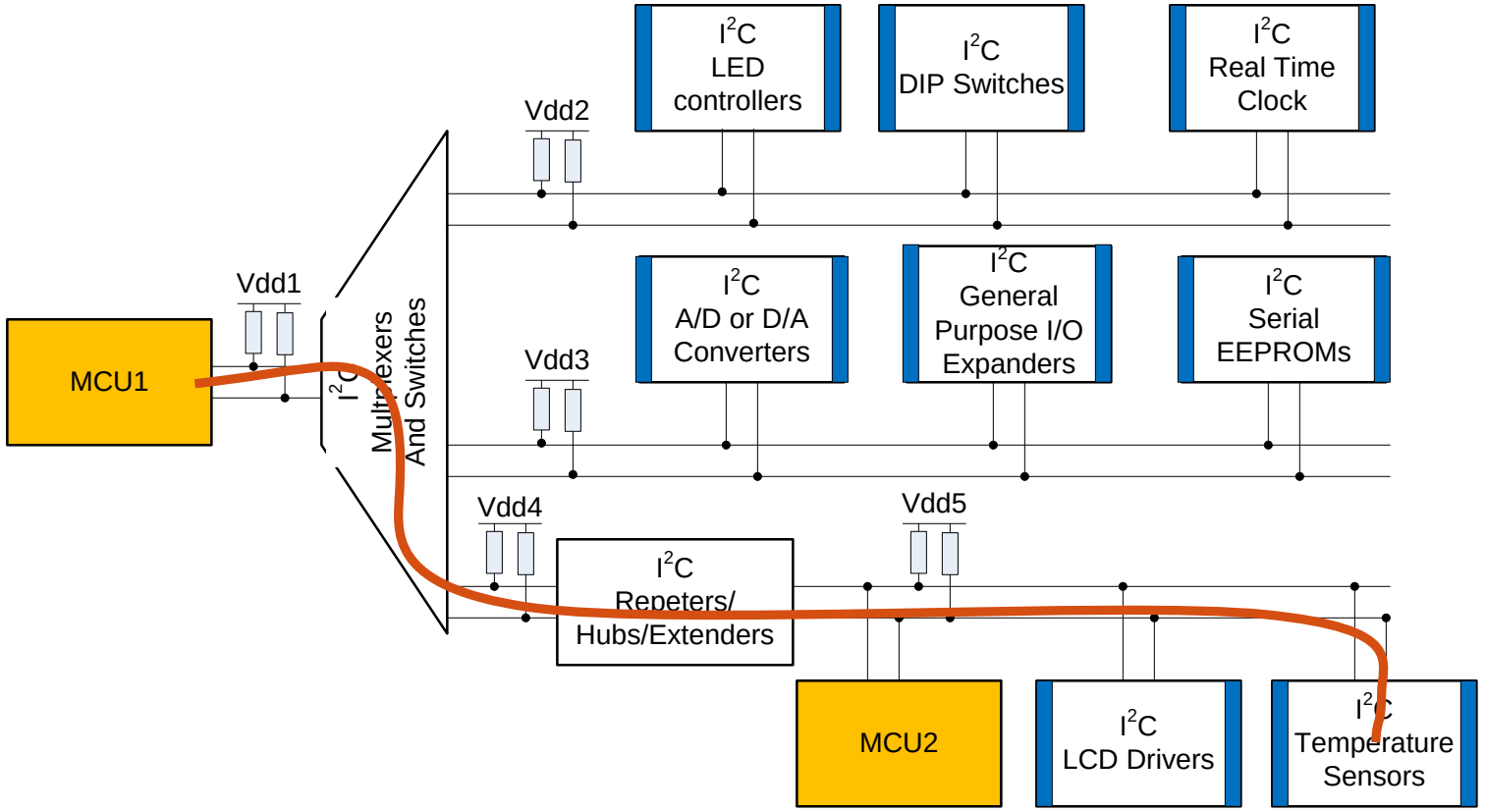
- The I²C BUS was developed in the early 1982 by Philips Semiconductors (now NXP Semiconductors). Its original purpose was to provide an easy way to connect a CPU to peripheral chips in a TV-set
- 1992 Philips Semiconductor released the I²C BUS Specification Version 1.0
  - Fast-mode (Fm) with I²C transfers rate up to 400 Kbit/s
  - 7-bit addressing and 10-bit addressing to allow additional 1024 slave addresses
- 1998 Philips Semiconductor released the I²C BUS Specification Version 2.0
  - High-speed mode (Hs) with I²C transfers rate up to 3.4 Mbit/s
- 2006, Philips Semiconductors CEO Frans van Houten revealed that the company will move forward as NXP Semiconductors
  - I²C specification were published under NXP Semiconductors

History of I²C BUS

- 2007 NXP Semiconductors released the Version 3 of I²C specification
  - Fast-mode Plus (Fm+) with I²C transfers rate up to 1 Mbit/s.
- 2013, NXP Semiconductors released the I²C BUS Specification Version 4
  - Ultra Fast-mode (UFm) speed up to 5 Mbit/s
- Today, the I²C BUS is used in many other application fields than just audio and video equipment. The bus is generally accepted in the industry as a de-facto standard
- Latest version of the I²C BUS Specification is 6. Download at NXP website: http://www.nxp.com/documents/user_manual/um10204.pdf

General I²C BUS Protocol and Transaction

- I²C BUS can have:
  - Multiple masters
  - Multiple slaves
- Only one master talks to one slave at a time
- All the slaves on the same bus must have different address
- Slow speed device cannot understand higher speed transfer



General I²C BUS Protocol and Transaction

I²C BUS constructs off 9 bit block

START condition:
When SCL is HIGH then SDA goes
from HIGH to LOW

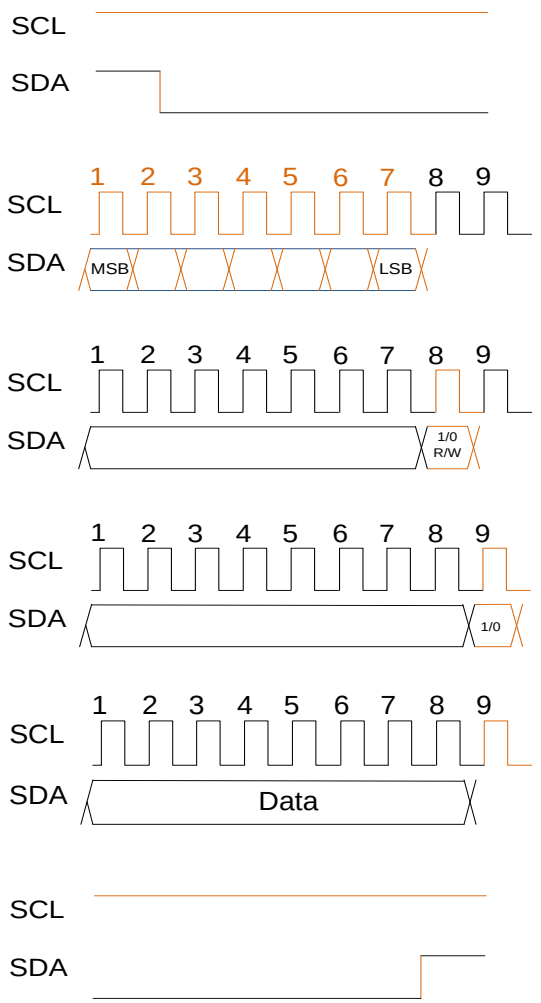
Address bit
7 bit after START condition

Read or Write bit
After 7 bit address, the 8th bit is Read or Write bit.
1 = Read cycle or 0 = Write cycle

ACK
Synchronization bit between master and slave
0 = ACK and 1 = NACK

Data Byte
8-bit after address byte is data byte from master or slave

STOP condition:
When clock line (SCL) is HIGH
then the data line (SDA) goes LOW to HIGH



All slaves on this bus
pay attention !!!

Master wants
to talk slave with this address

Master wants to read or write
0: Write cycle
1: Read cycle

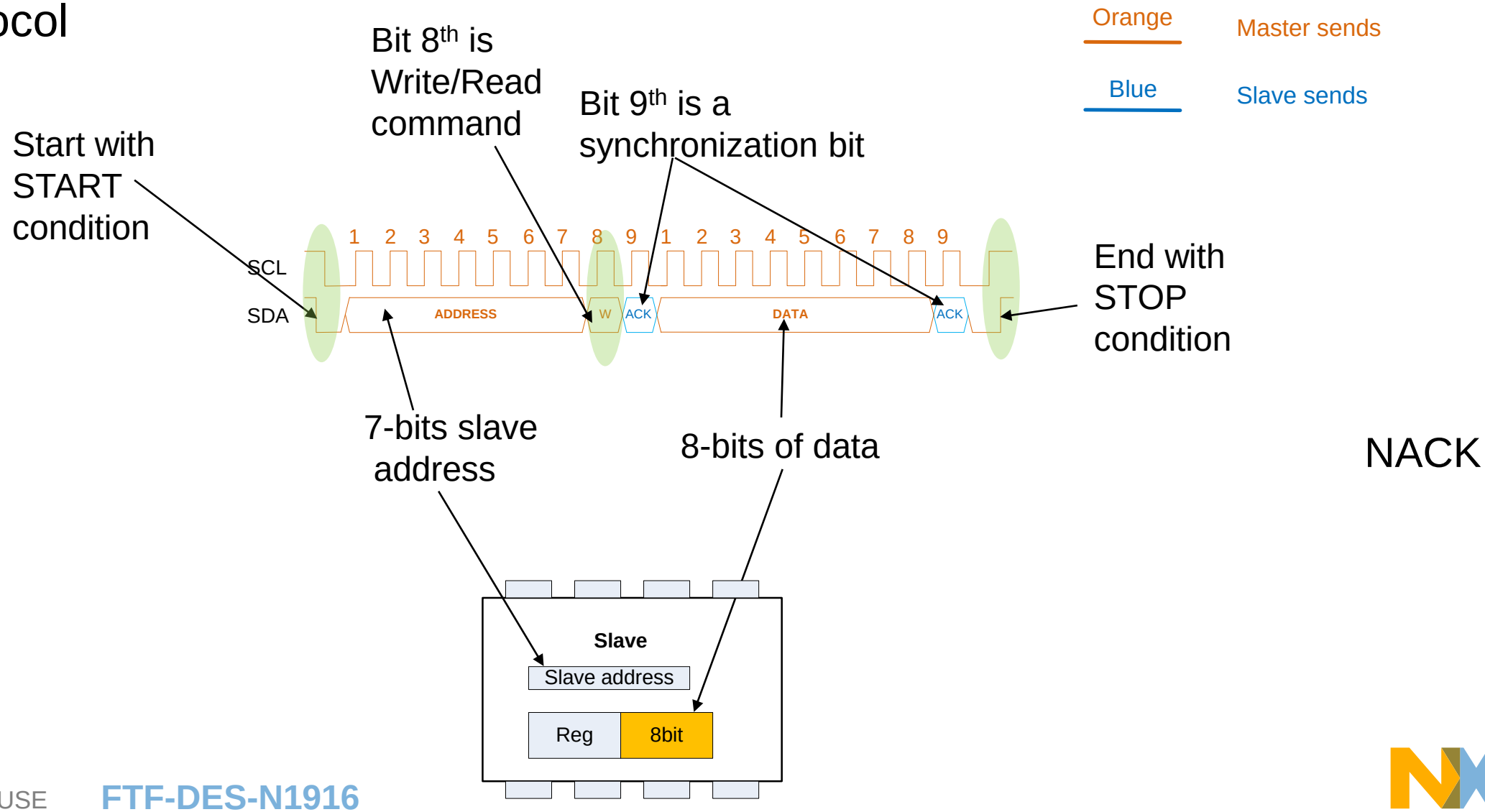
Slave or master:
0: I am here or data received
1: not me or data not received

Data byte
Master sends data when write cycle
Slave sends data when read cycle

Master notifies the slave this is the
end of transaction

General I²C BUS Protocol and Transaction

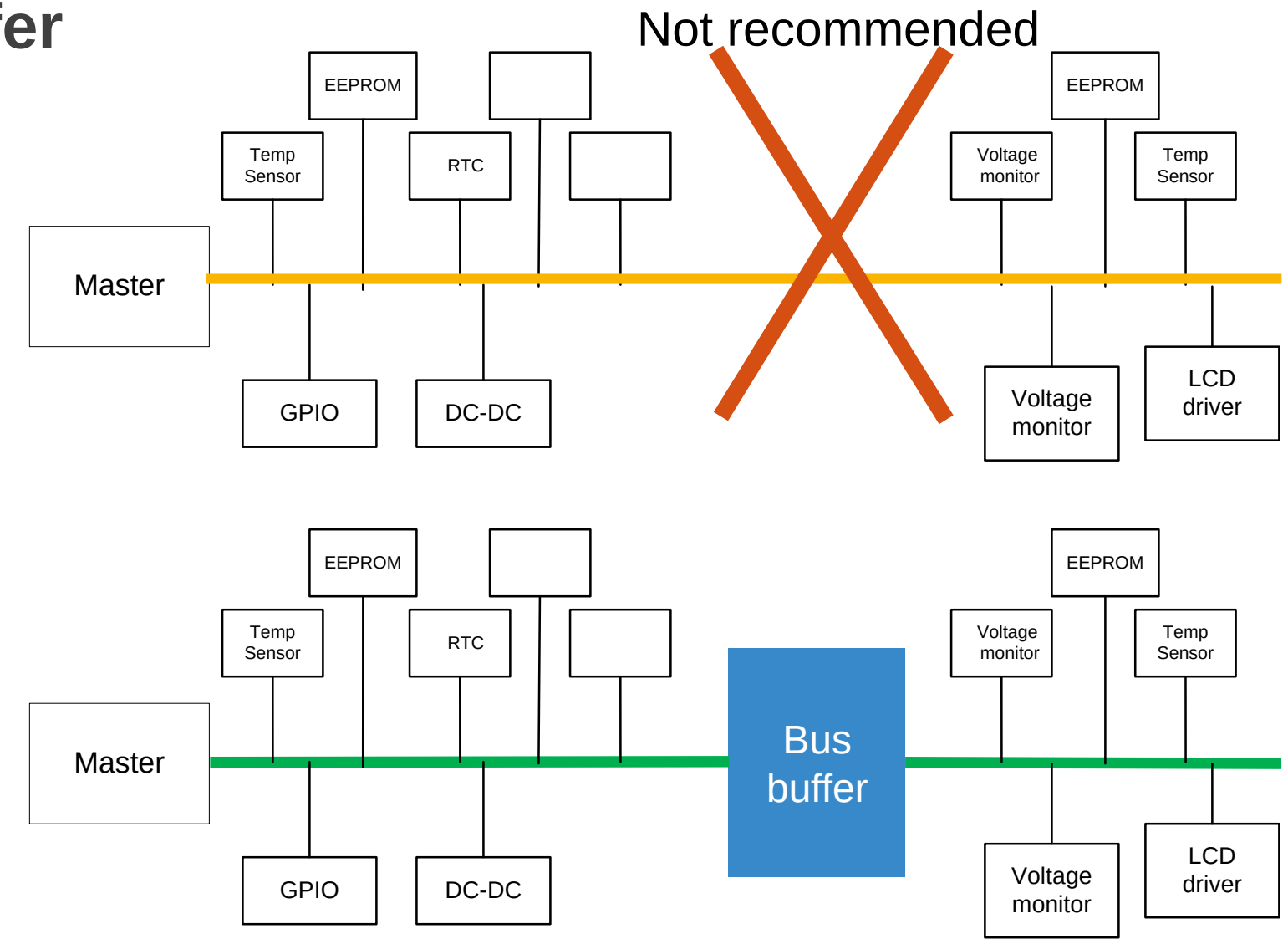
I²C protocol



What Is an I²C BUS Buffer

Bus buffer is for:

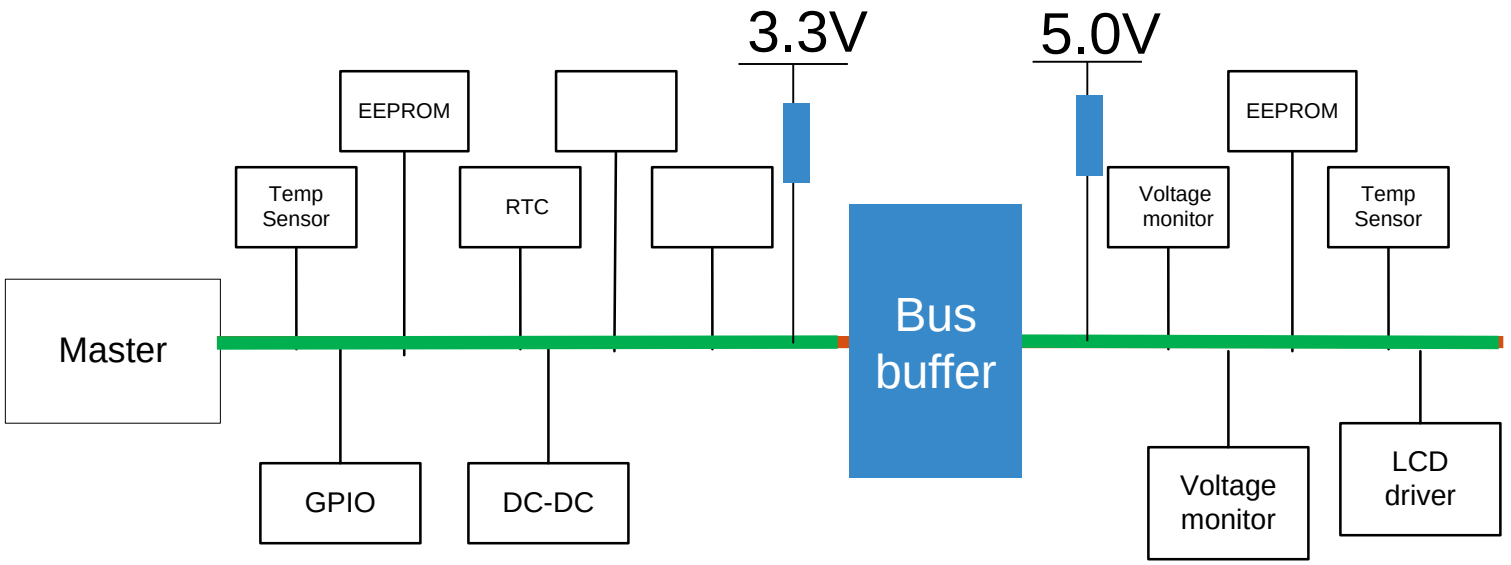
- Divide Capacitive load of I²C BUS in half (Bus isolation)



What Is an I²C BUS Buffer

Bus buffer is for:

- Divide Capacitive load of I²C BUS in half (Bus isolation)
- Connect two different bus voltage together (Level translation)
- Voltage range from 0.8 V to 5.5V



What Is an I²C BUS Buffer

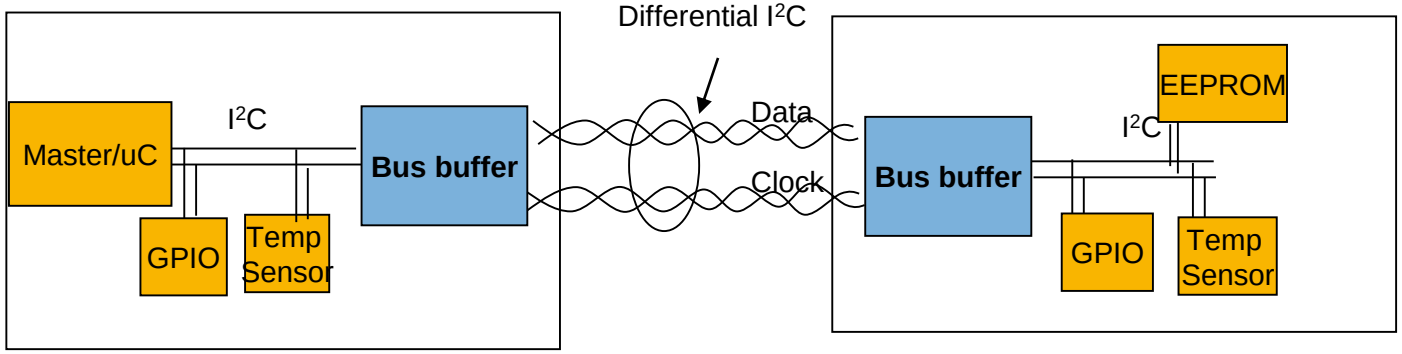
Bus buffer is for:

- Divide Capacitive load of I²C BUS in half (Bus isolation)
- Connect two different bus voltage together (Level translation)
- Voltage range from 0.8 V to 5.5V
- Increate long range connection (Extend longer bus)

What Is an I²C BUS Buffer

Bus buffer is for:

- Divide Capacitive load of I²C BUS in half (Bus isolation)
- Connect two different bus voltage together (Level translation)
- Voltage range from 0.8 V to 5.5V
- Increate long range connection (Extend longer bus)
- Avoid system noise (differential I²C BUS)



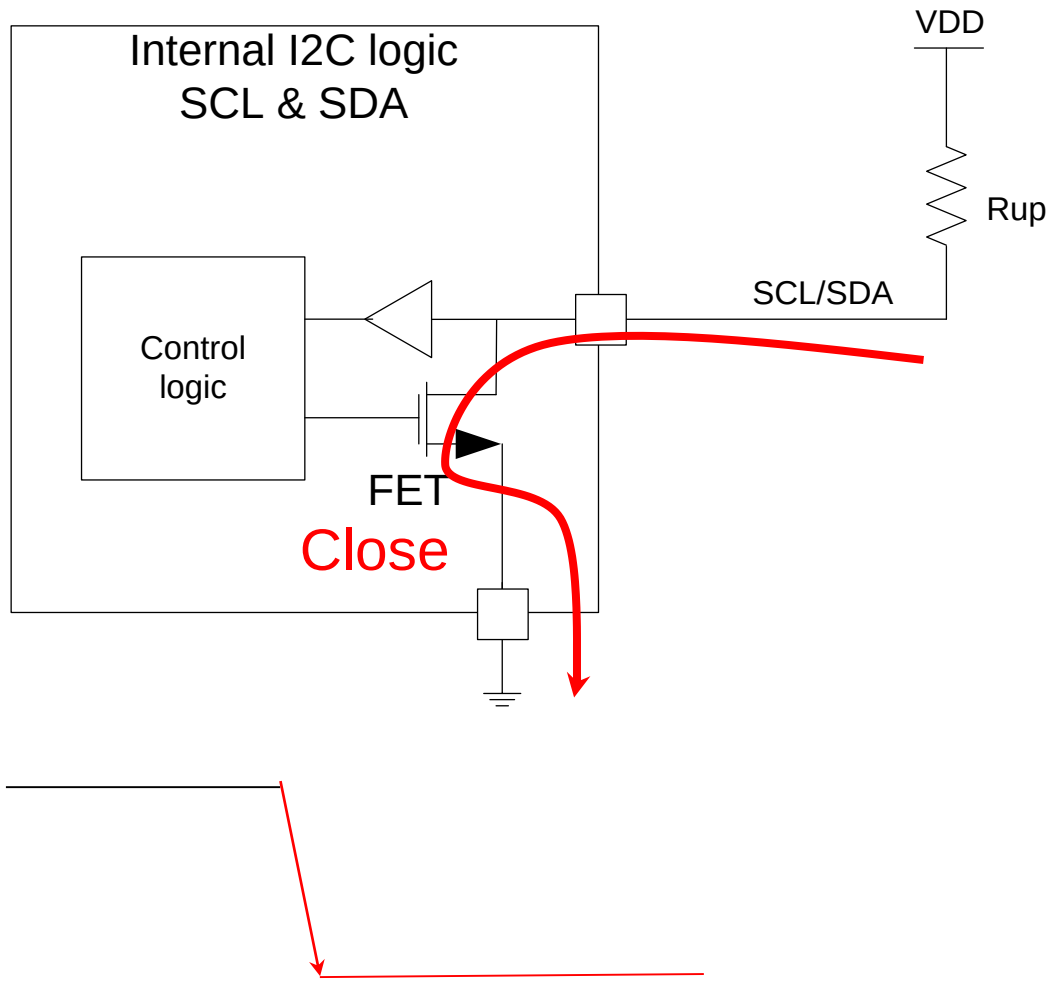
Why I²C BUS Buffer Is Needed?

I²C internal block diagram

How I²C BUS works

When master or slave output LOW:

- FET is active and pull the output to ground
- If Rup value is small the ground offset will be higher



Why I²C BUS Buffer Is Needed?

I²C internal block diagram

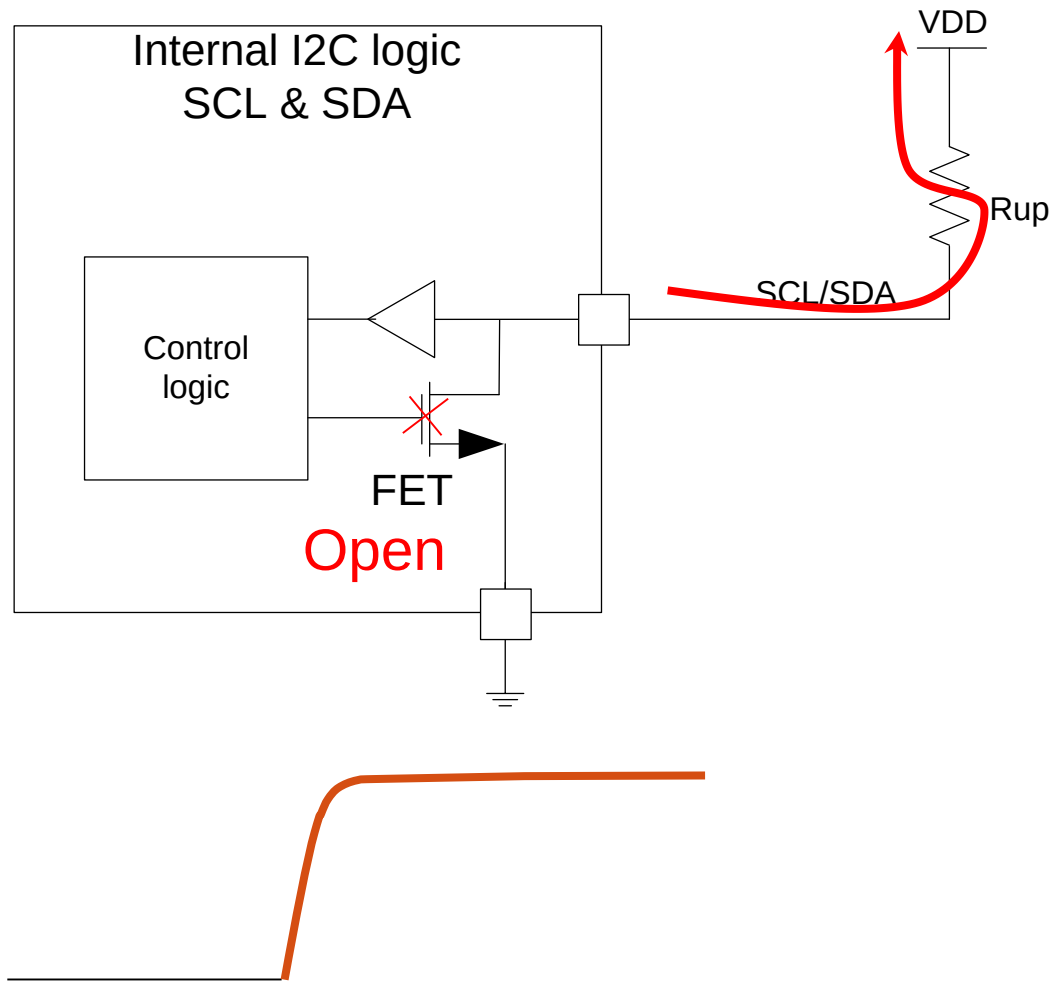
How I²C BUS works

When master/slave output LOW:
- FET is active and pull the output to ground

When master/slave output is HIGH
- FET open and Rup will pull SCL/SDA to VDD

Max drive strength of CPU for I²C is 3mA or 400pF

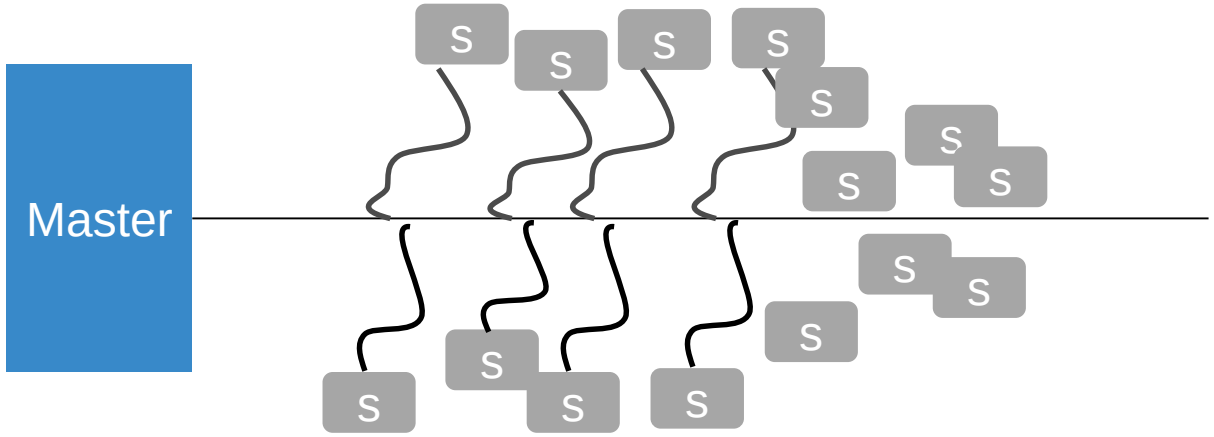
Number of slave devices are depended on:
- Drive strength of FET
- Capacitance of PCB trace length
- Capacitance load of each pin connect to the bus

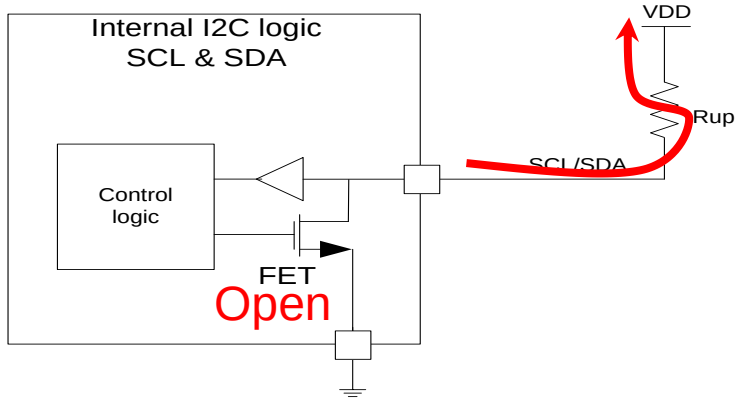


Why I²C BUS Buffer Is Needed?

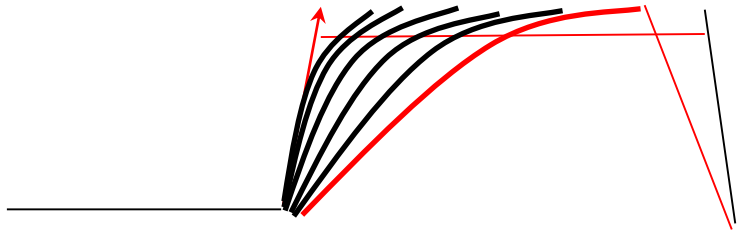
What happens when the bus has too many slave devices

- Longer trace length
- More capacitance load
- Distort rise and fall time



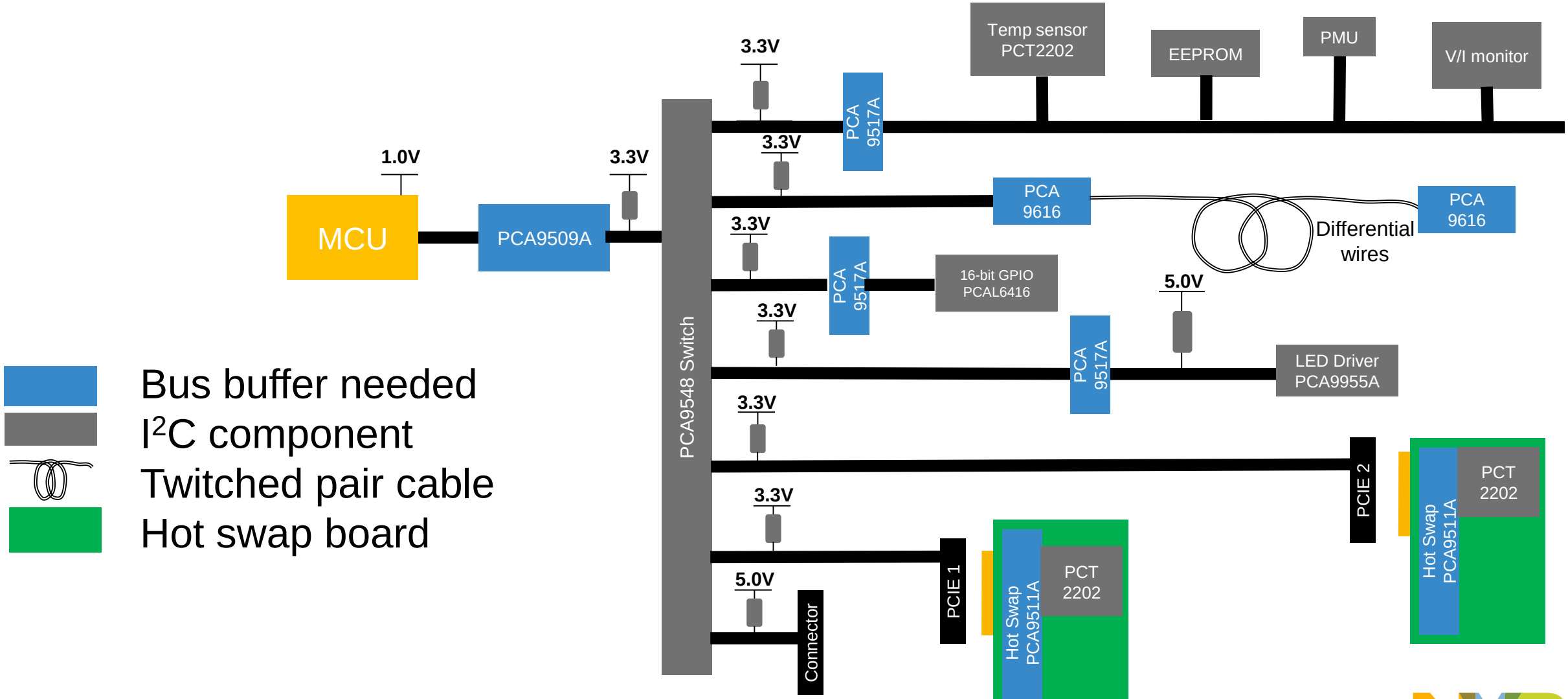


Mode	standard	Fm	Fm+
Rise time max	1000ns	300ns	120ns



System needs bus buffers to improve quality of the signals

Example for Bus Buffers



I²C BUS Buffer Family

Repeaters

- **PCA9509** Processor to SMBus — SO
- **PCA9509A** Processor to SMBus — SO
- **PCA9509P** Processor to SMBus — SO
- **PCA9517A** 0.9 – to 5.5V — SO
- **PCA9617A** 0.8 to 5.5V — SO
- **PCA9507** RTA for HDMI — SO
- **PCA9527** PCA9507 + ½ PCA9517 — SO
- **PCA9515/15A/16A** — SO
- **PCA9518A** 5-Channel Hub Expander — SO
- **PCA9519** 4 x PCA9509 — SO
- **PCA9646** 30-mA Drive; 1:4-Channel — NO

Hot-Swap Buffers

- **PCA9508** Active Level Shifter — SO
- **PCA9510A** No Accelerator — IO
- **PCA9511A** 0.6-V Threshold — IO
- **PCA9512A/12B** Active Level Shifter — IO
- **PCA9513A** 92-µA Current Source — IO
- **PCA9514A** 0.8-V Threshold — IO

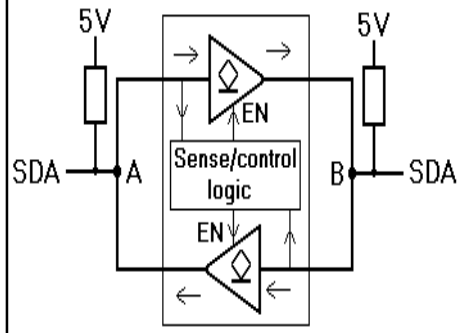
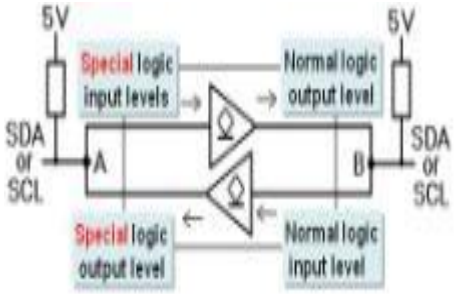
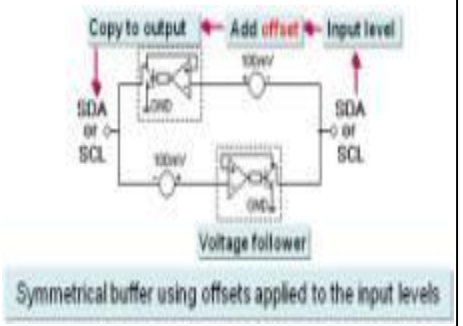
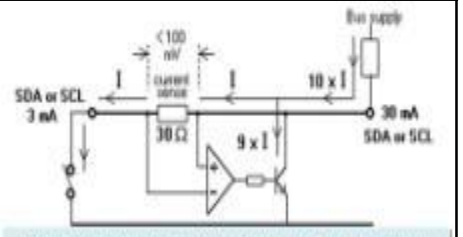
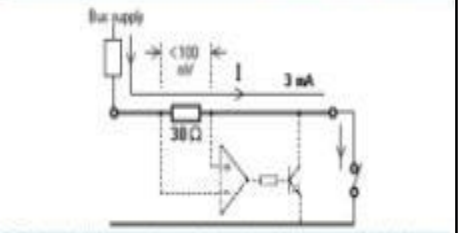
Extenders (Long Cable)
(Long Cable Drivers)

- **P82B96** [1] — SO
- **P82B715** No Static Offset — AM
- **PCA9600/9601** 1-MHz Speed — SO
- **PCA9614/15** 2-CH Differential Driver — SO
- **PCA9616** 3-CH Differential Driver — SO

NO = No Offset
SO = Static Offset
IO = Incremental Offset
AM = Amplifier

[1] P82B96 is widely used for opto-isolation applications
Blue ➔ 1 MHz system

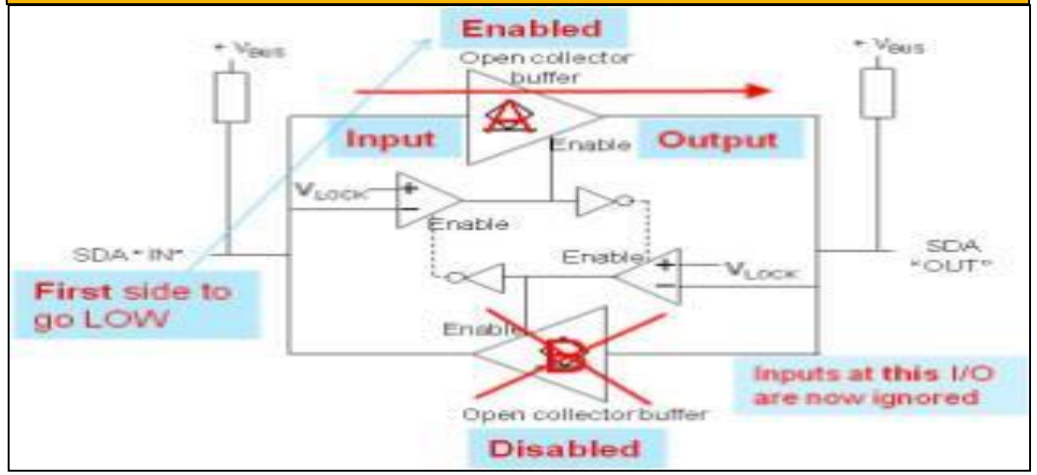
Understanding I²C BUS Buffer Technologies

No Offset	Static Offset	Incremental Offset	Amplifier
		 Symmetrical buffer using offsets applied to the input levels	 Asymmetrical device, amplifies sink current in one direction  Behaves like a passive resistor when higher current side is pulled low
Characteristics: • Buffer automatically changes direction in response to the applied drive signal • For use on SDA line ONLY • Buffer output has "**no offset**" and pulls its output down very close to 0V	Characteristics: • A fixed third logic level is introduced on the special I/O side • The third logic level is higher (**static offset**) than 0.4V, but low enough to still be a low to the other devices on the bus	Characteristics: • A dynamic third logic level is introduced by adding an **incremental offset** voltage to the input • Either bus will be driven (down), by a voltage follower, to a level 100mV higher than the voltage level of the other bus	Characteristics: • The bus current is **amplified**, but in only one signal direction • Current gain enables a 3-mA driver on the input side to drive a bus with 30-mA pull-up on the other side

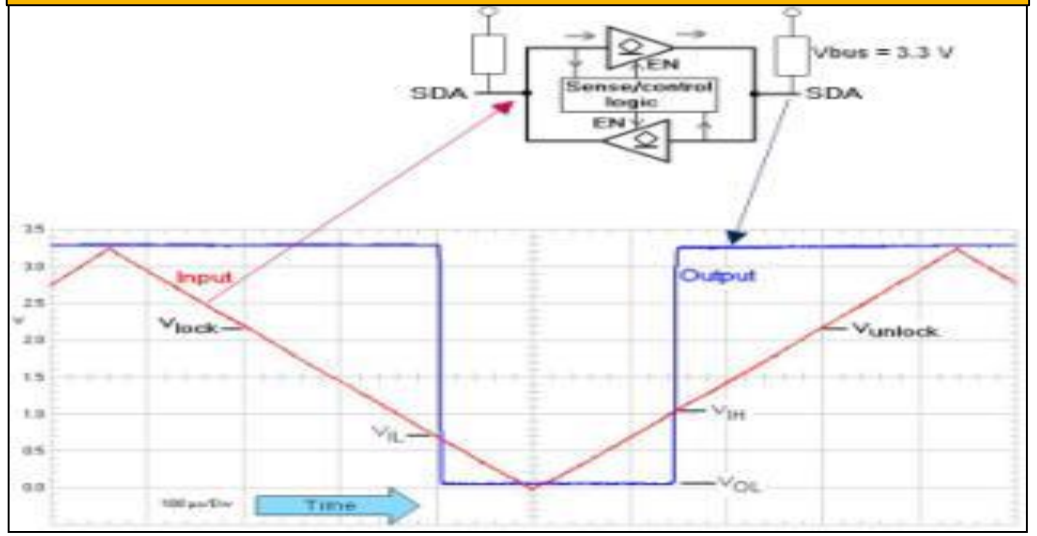
No Offset Buffer Characteristics

- Compatible (but not compliant) with I²C BUS or similar buses
- Uni-directional clock depend on direction setting
- The first one to score (first input to go low) takes control of the bus and the other is disabled
- Compatible (but not compliant) with I²C BUS or similar buses
- When an input falls below Vlock, the buffer becomes enabled and the other buffer is disabled. When the controlling input signal rises above Vunlock, the buffer is disabled and the device is ready to process any side that goes low next or may be already low, but was unable to take control
- To prevent lock-up, Buffers A and B must never be enabled at the same time
- clock stretching is not allowed and a true multi-master system cannot be implemented
- Both buffers are able to drive their outputs fully low, near 0V
- Devices with “No Offset” are PCA9525, PCA9605 and PCA9646

Scoreboard Direction Control

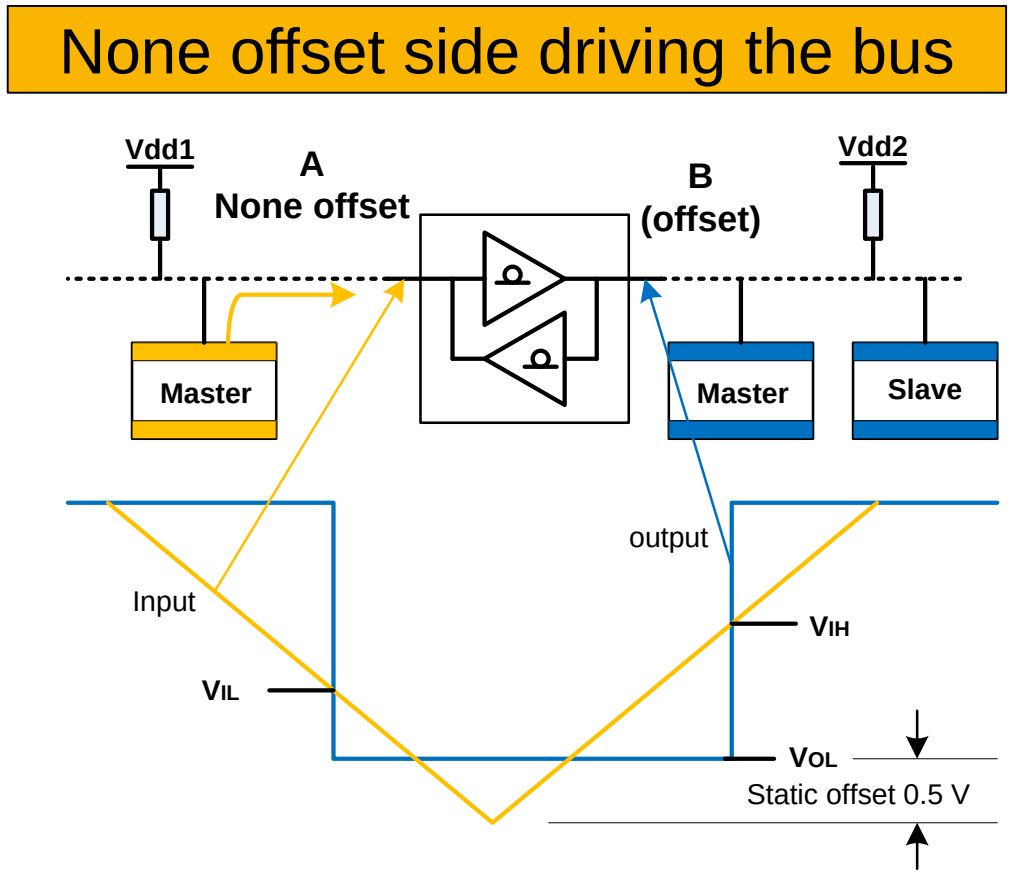


Input/Output Waveform



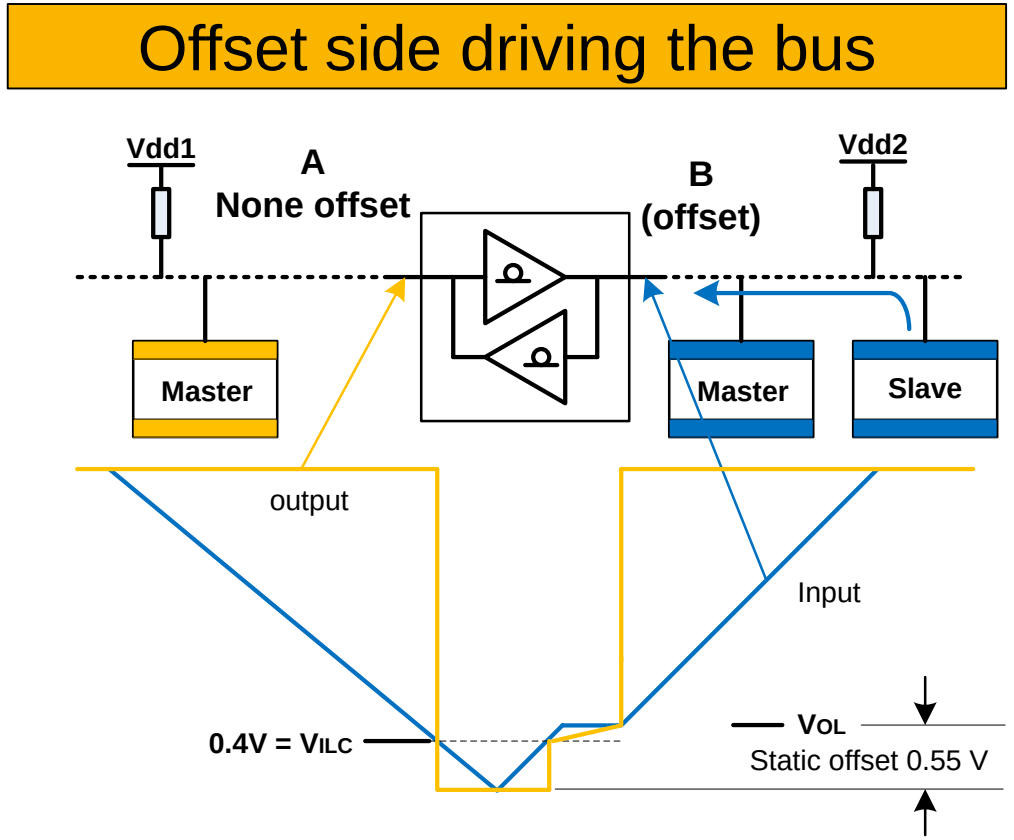
Static Offset Buffer Characteristics

- Compatible (but not compliant) with I^2C BUS or similar buses
- Fully bi-directional
- Support multi-master operation, including clock stretching
- Isolates the bus
  - The resistive and capacitive load will be independent on each side of the bus
- Non offset side drives (A side)
  - **When A input goes below VIL, the B output will be pulled LOW to the static offset**
  - **When A input goes above VIH, the B output will be released to Vdd2**
- Devices with “Static Offset are PCA9507, PCA9508, PCA9509, PCA9515A, PCA9516A, PCA9517A, PCA9518A, PCA9519, PCA9614, PCA9615, PCA9616, PCA9527, PCA9600, PCA9601, P82B96



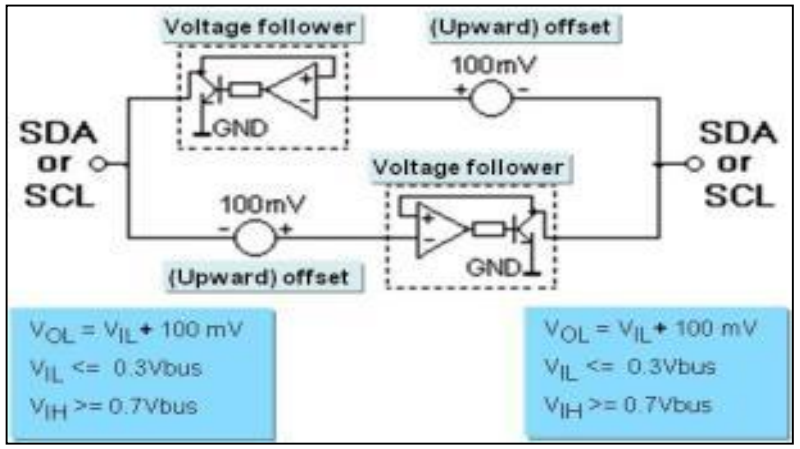
Static Offset Buffer Characteristics

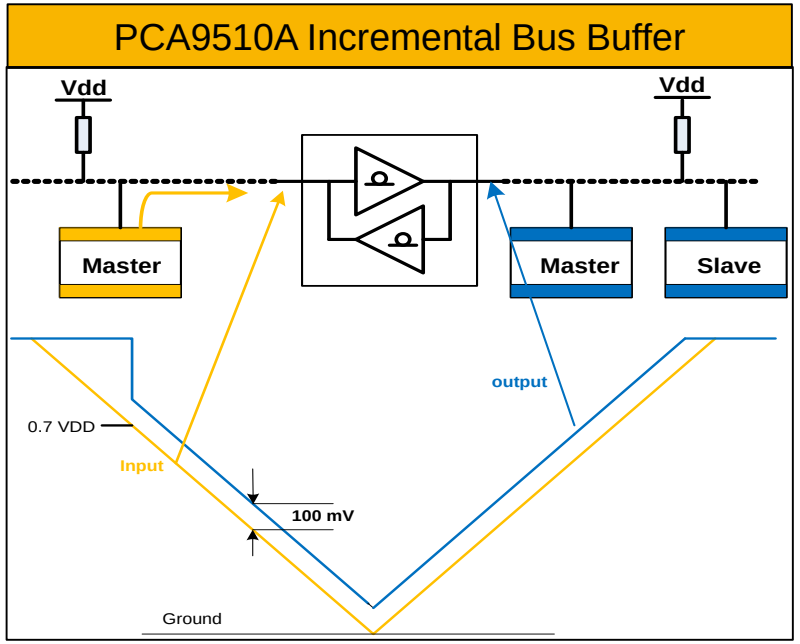
- Compatible (but not compliant) with I^2C BUS or similar buses
- Fully bi-directional
- Support multi-master operation, including clock stretching
- Isolates the bus
  - The resistive and capacitive load will be independent on each side of the bus
- Offset side drives (B side)
  - **When B input goes below VILC, the A output will be pulled LOW to ground**
  - **When B input goes above VILC, the A output will be released**
  - **And the B side will wait at 0.55V until A side above VOL (offset 0.55V)**
- Devices with “Static Offset are PCA9507, PCA9508, PCA9509, PCA9515A, PCA9516A, PCA9517A, PCA9518A, PCA9519, PCA9614, PCA9615, PCA9616, PCA9527, PCA9600, PCA9601, P82B96



Incremental Offset Buffer Characteristics

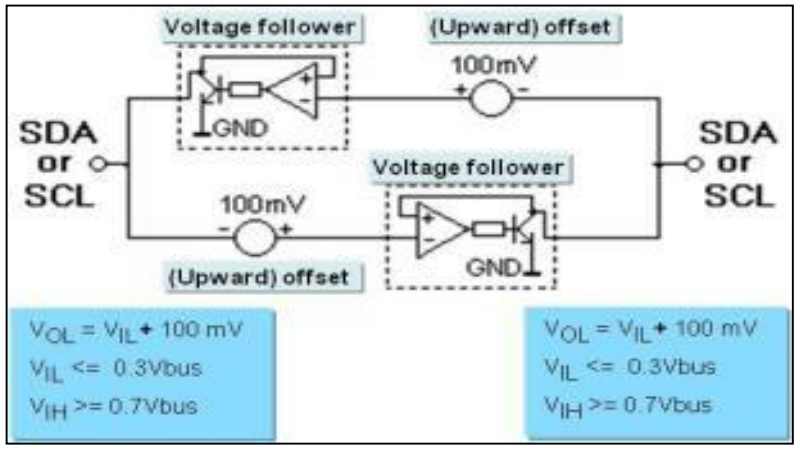
- Compatible with I²C BUS or similar buses
  - Fully bi-directional
  - The V_{IL} and V_{IH} levels are I²C BUS compliant
  - Support multi-master operation, including clock stretching
  - The output will be 100 mV lower then the input signal
  - Pre-charge I²C BUS before make connection
  - Check bus condition before make connection for hot-swap application



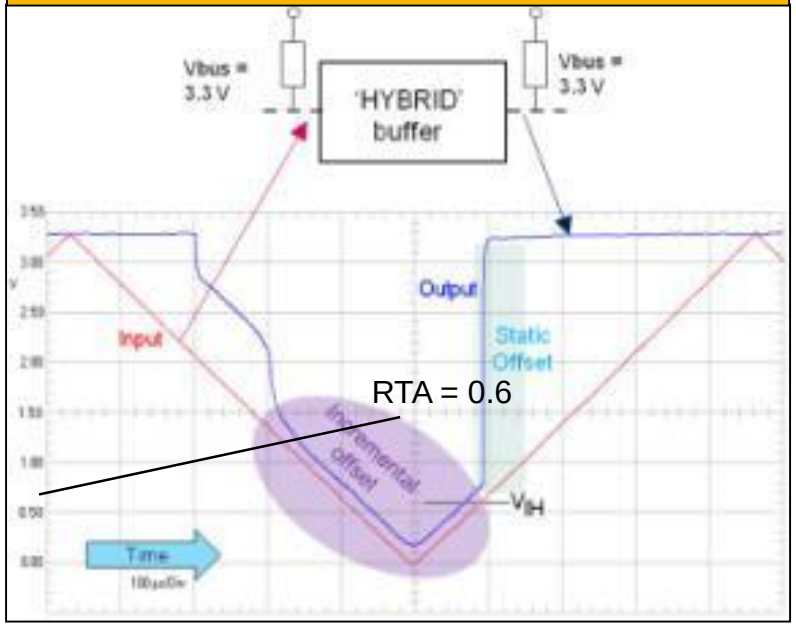


Incremental Offset Buffer Characteristics

- Compatible with I²C BUS or similar buses
  - Fully bi-directional
  - The V_{IL} and V_{IH} levels are I²C BUS compliant
  - Isolates the bus
  - Support multi-master operation, including clock stretching
  - The output will be 100 mV lower then the input signal
  - Pre-charge I²C BUS before make connection
  - Check bus condition before make connection for hot-swap application
  - Rise Time Acceleration



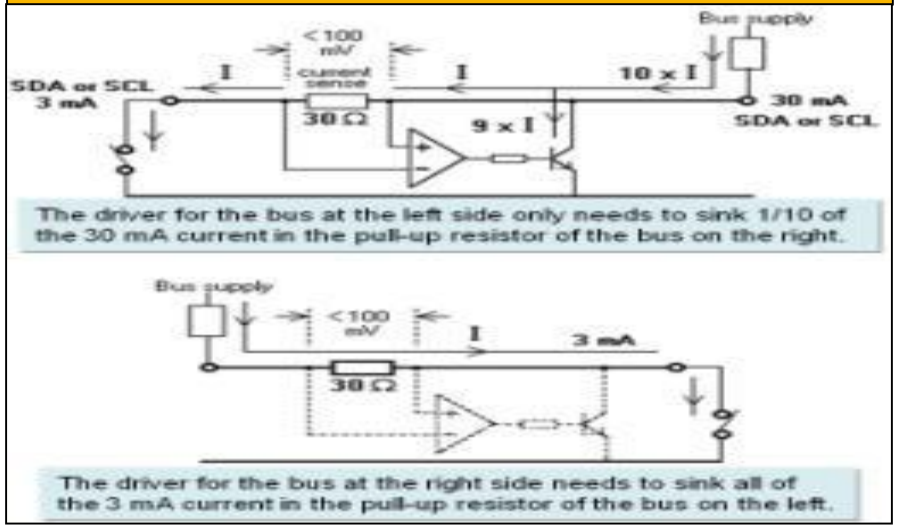
PCA9511A Incremental Bus Buffer



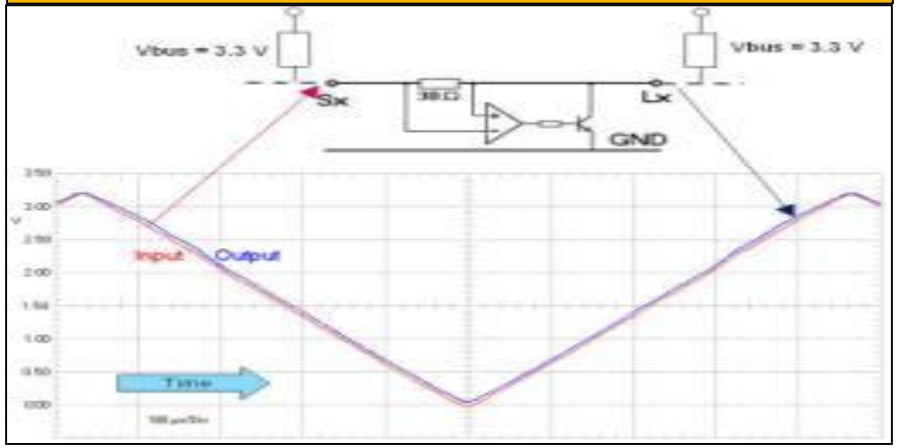
Amplifier Buffer Characteristics

- Compatible (and compliant) with I²C BUS or similar buses
  - Fully bi-directional
  - P82B715 has a current gain of 10x in one direction and allows connection of a bus with its pull-up resistor scaled for 30-mA sink current
  - Because the two I/Os are joined internally by a 30Ω resistor, there can never be a voltage difference greater than 100mV between them. Typically, it's about 70mV. Logic voltage shifting is not possible for this part
  - Capacitive loading is also scaled. 4000pF on the 30-mA bus is equivalent to 400-pF on the 3-mA bus

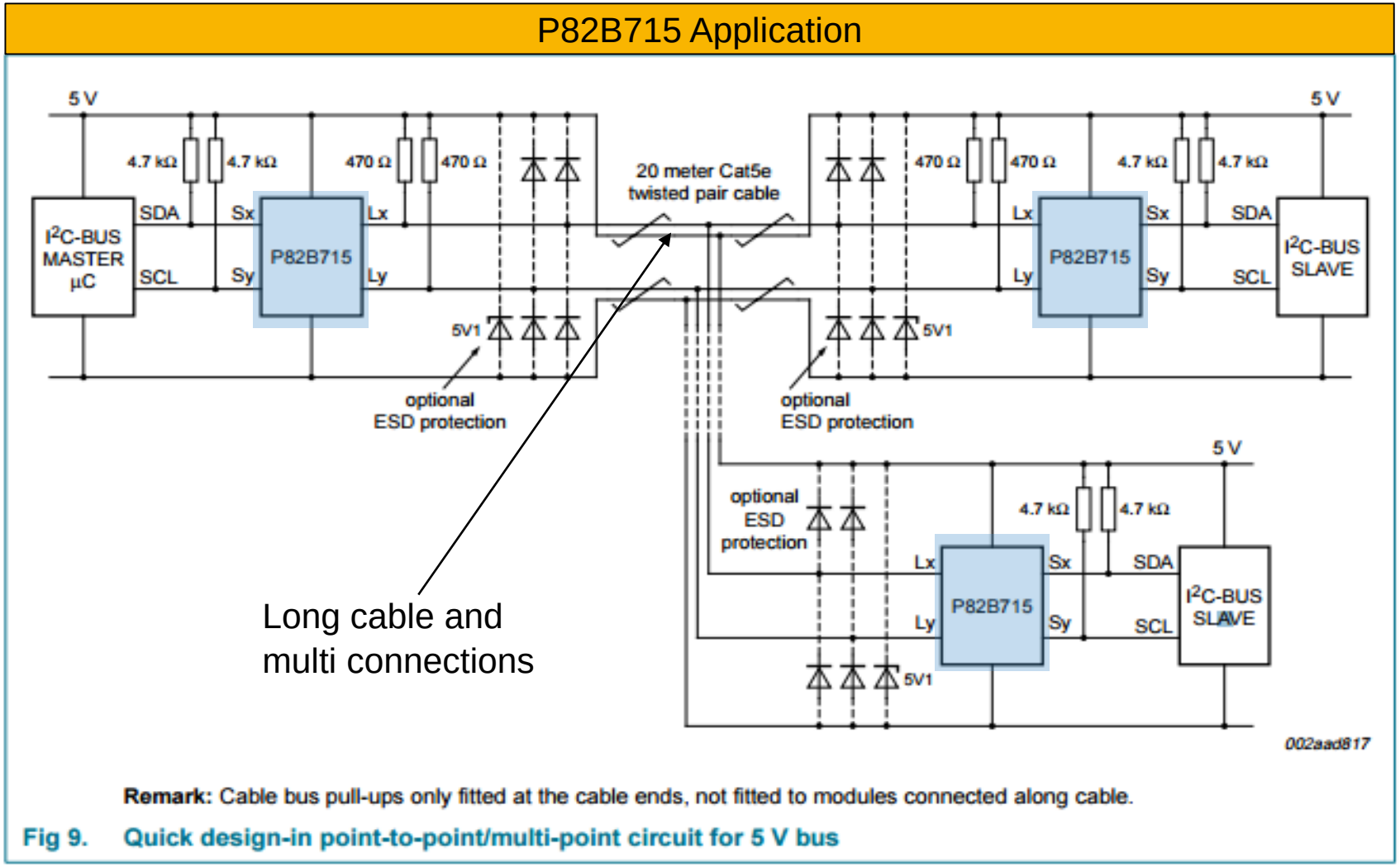
Asymmetrical, No Direction Control



P82B715 Input/Output Waveform



Amplifier Buffer Application



Remark: Cable bus pull-ups only fitted at the cable ends, not fitted to modules connected along cable.

Fig 9. Quick design-in point-to-point/multi-point circuit for 5 V bus

I²C BUS Information

If you have any questions regard to I²C BUS. Please email to i2csupport@nxp.com

For more information about I²C bus buffer products go to www.nxp.com/i2c

or

http://www.nxp.com/products/interface-and-connectivity/interface-and-system-management/i2c/i2c-bus-repeaters-hubs-extenders:mc_41849

Thank you very much

SECURE CONNECTIONS
FOR A SMARTER WORLD

ATTRIBUTION STATEMENT