Hybrid European IBIS Summit at SPI 2026

Turin, Italy

June 17, 2026

IBIS for I3C serial transmission

Presenter: Fabio Brina

Author: Fabio Brina

STMicroelectronics

Agenda

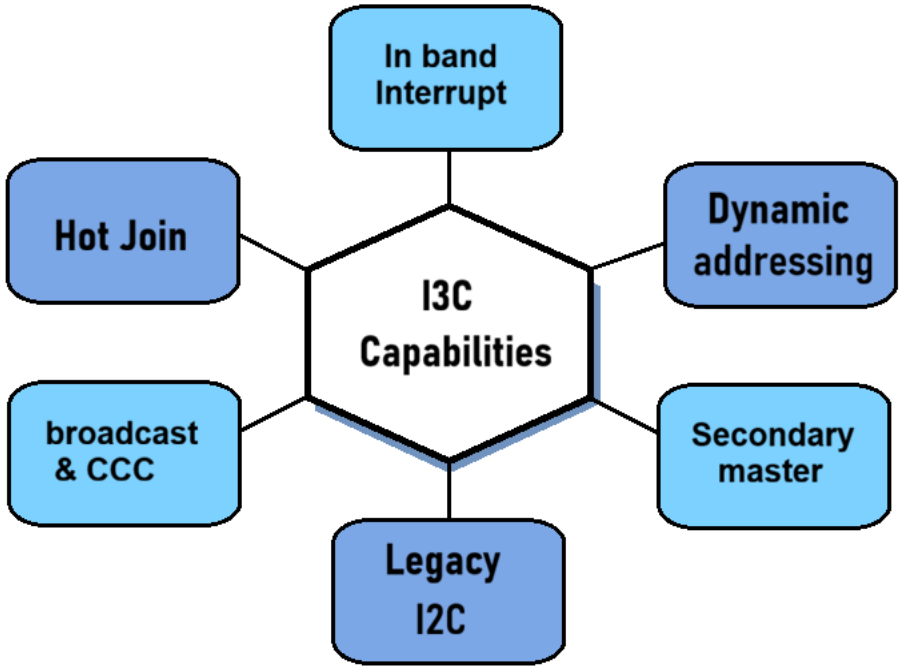
1. Introduction to **I3C** protocol and comparison with SPI and I2C
2. Characterization of I3C driver through IBIS **I/O** and IBIS **I/O_open_drain**
3. Simulation of I3C IBIS models into a specific **Test-bench** with both IBIS types
4. Simulation of the wasted **Power** of I3C IBIS Test-bench
5. Considering the opportunity to update the **I/O_open_drain** Model Type to include an Internal *pull-up resistor* in the open drain structure

I3C Protocol

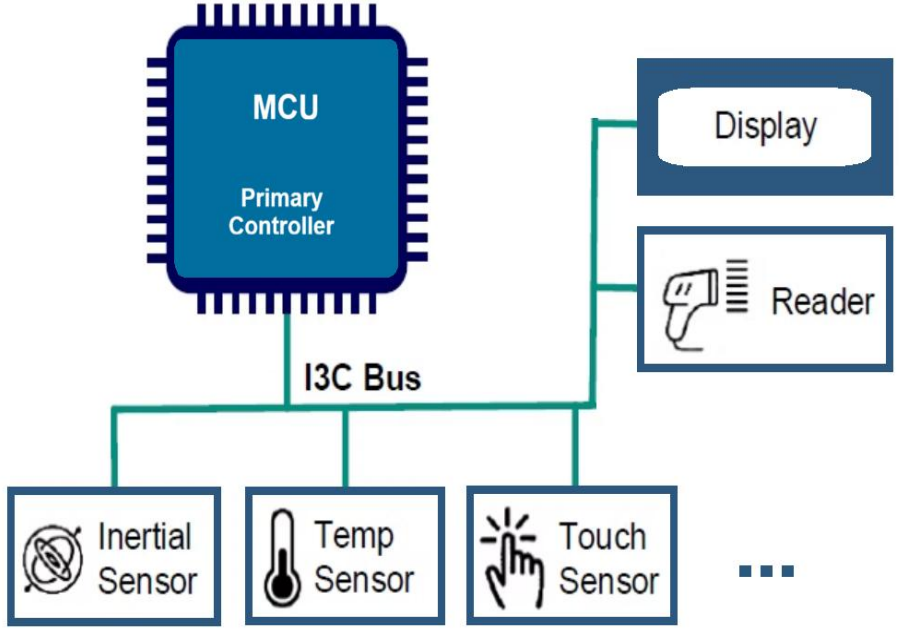
- **I3C (I**mproved **I**nter **I**ntegrated **C**ircuit**)** is a serial half-duplex communication protocol developed initially by the MIPI Alliance.
- **I3C** protocol consider **Start** and **Stop** conditions in any transmission
- **I3C** protocol after each **Start** consider **7 bits of address** assigned by **controller;**
- **I3C** protocol consider **9 bits frame** for data (the 9th bit is the T-bit) *write* or *read* operations
- **I3C** devices use **dynamic** address which are assigned by the controller on the bus (this is called DAA: Dynamic Address Assignment)

I3C Bus Applications

I3C Capabilities



I3C application is particularly suitable for communications in multi-sensors with Microcontrollers :



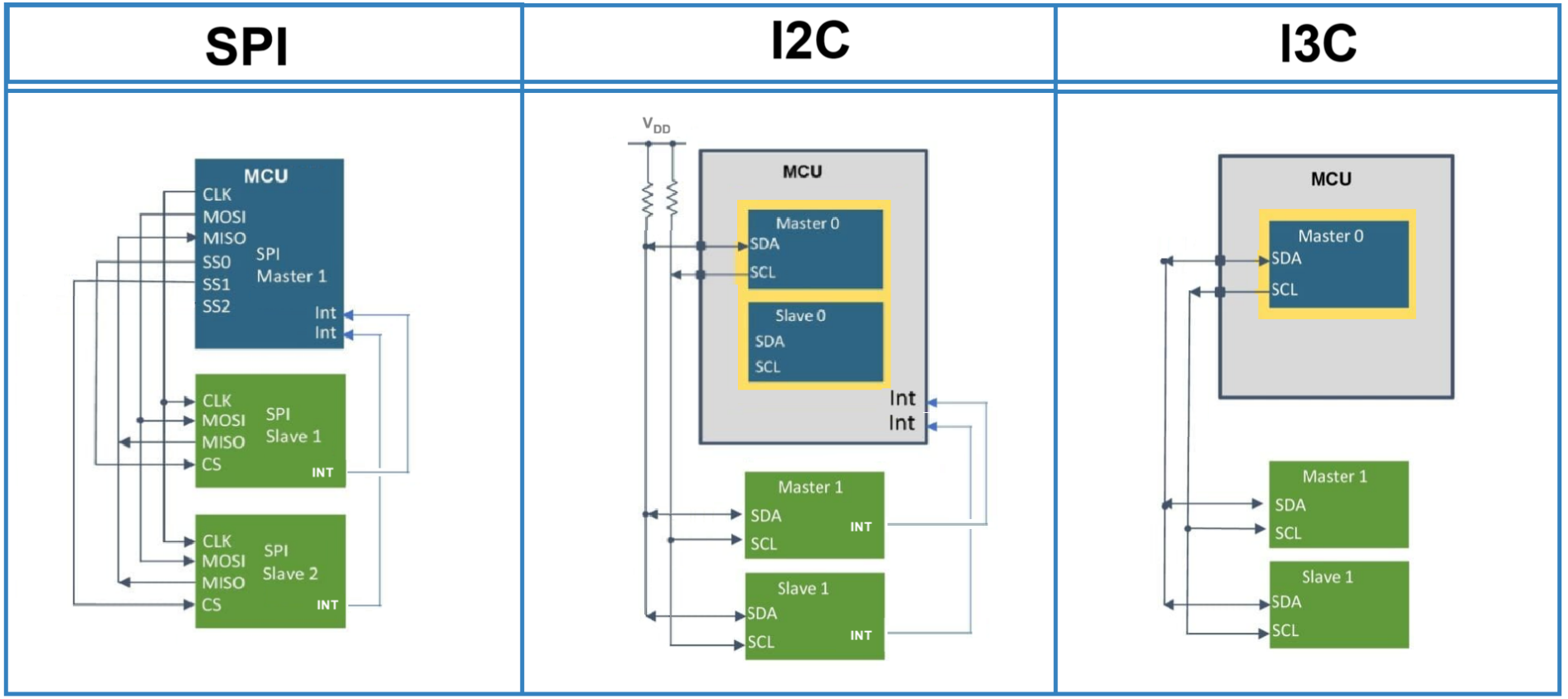
SPI – I2C – I3C

Comparisons:

SPI	I2C	I3C
Full duplex communications	Each device in the bus is independently addressable	Multi-drop capability and dynamic addressing
4 wires, plus separate wires for Interrupts	2 wires, plus separate wires for Interrupts	only two wires
speed (typ): 10 - 20 Mbps	speed (typ): 100 - 400 Kbps	speed (typ): 12.5 Mbps
Push-Pull drivers	Open Drain drivers	Open Drain & Push-Pull drivers

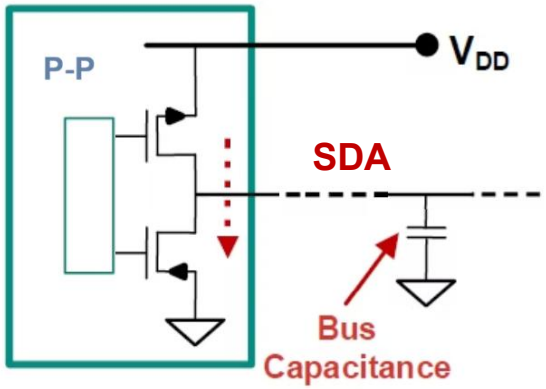
SPI – I2C – I3C

Wire Connections:



SPI – I2C – I3C

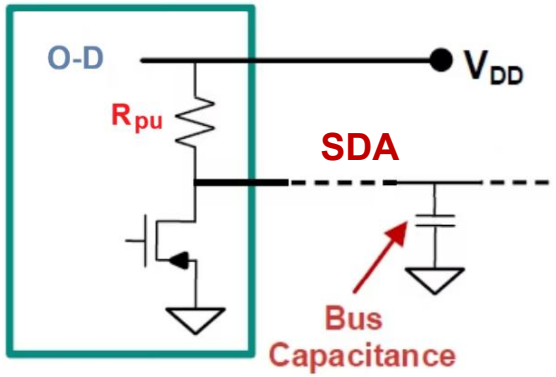
SPI



Push-Pull

Model_type:
I/O

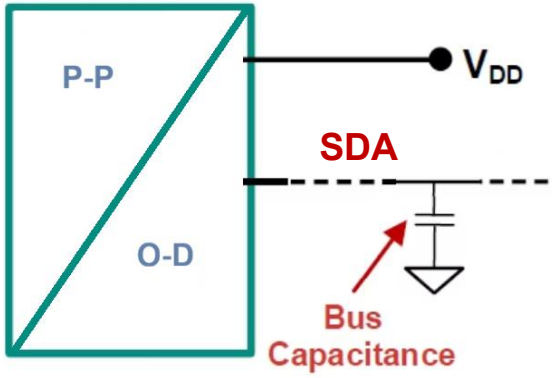
I2C



Open-Drain

Model_type:
I/O_open_drain

I3C



P-P / O-D

Model_type:
?

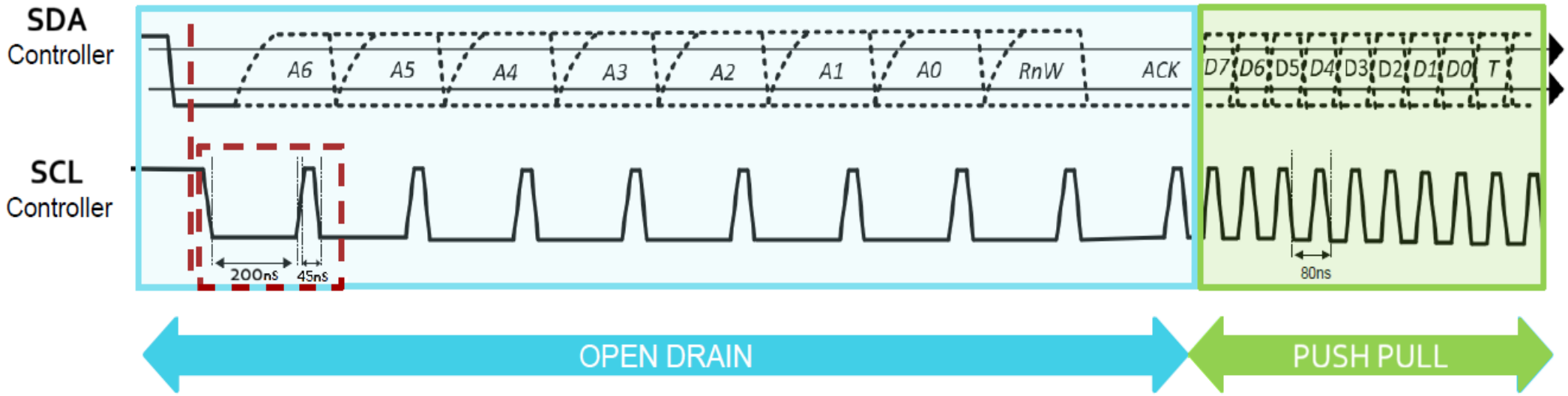
IBIS Modeling

I3C – SDR Mode (Single Data Rate)

SCL operates as Clock signal , SDA is in **Open Drain** mode (Addressing);

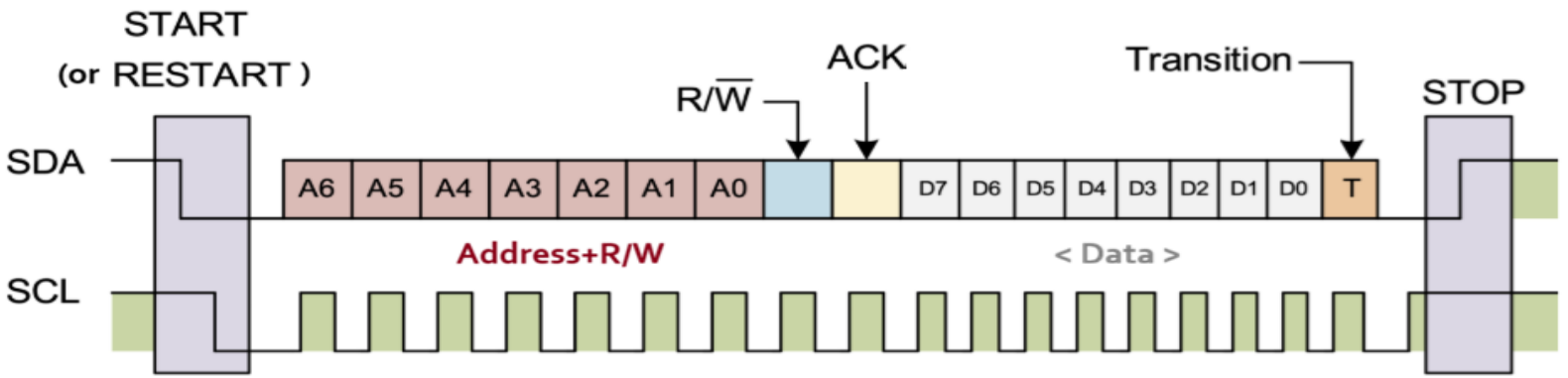
SCL high period is 45ns, the frequency is around 4MHz.

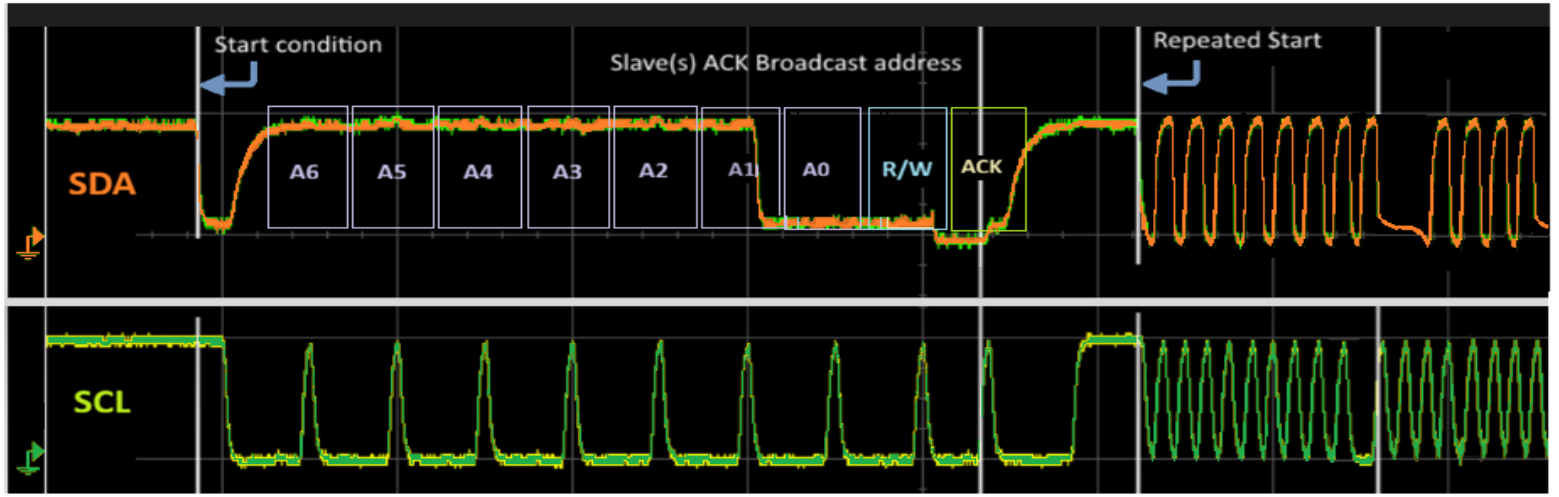
After *ACK* the Controller change SDA in **Push-Pull** mode and SCL increase its Clock to **12.5 MHz** (period = 80ns)



I3C communication

I3C – SDR Mode (Single Data Rate)



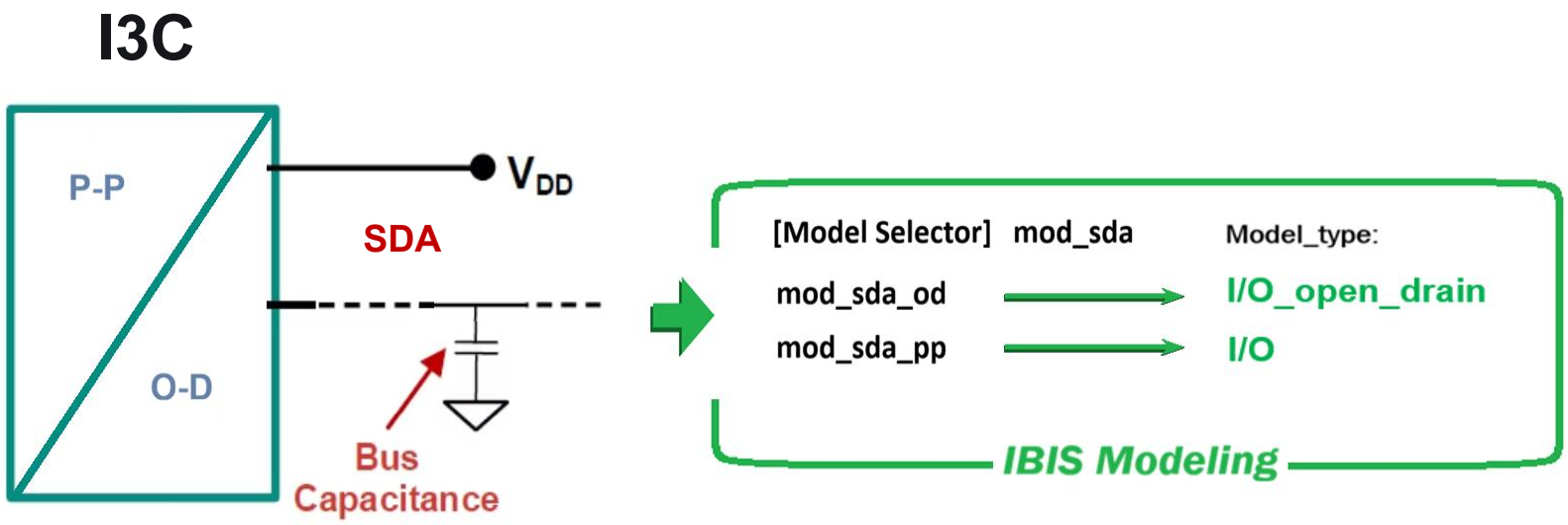


ACK bit confirm the address assignment to the Slave is correct

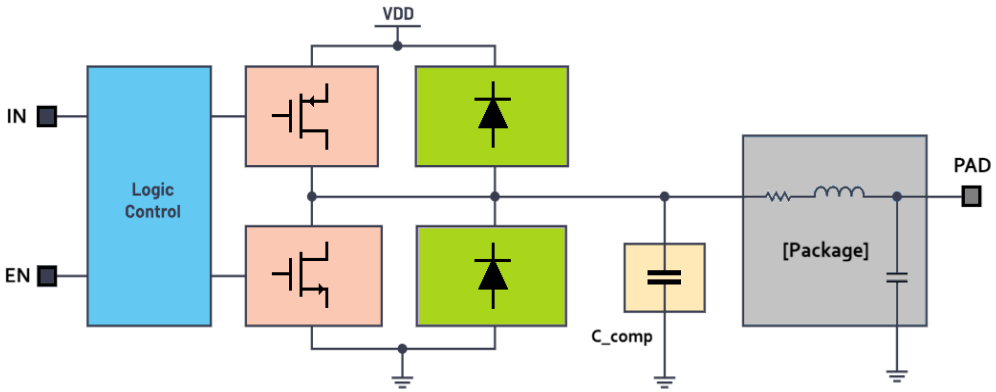
T-bit is an *odd-parity* bit (when Controller is writing)

T-bit is an *end of data* flag (when Controller is reading)

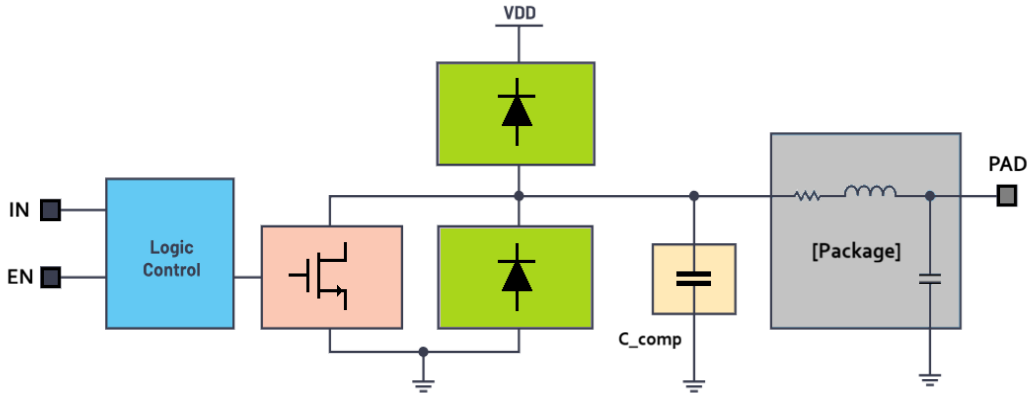
IBIS models for I3C



IBIS I/O Structure



IBIS I/O_open_drain Structure



IBIS models for I3C

IBIS I/O Model

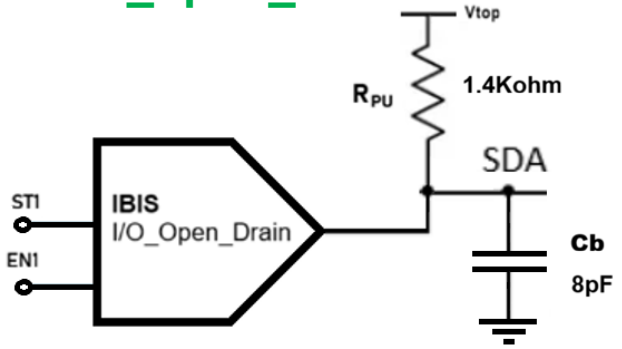
```
|
|************************************************
|            Model mod_sda push_pull
|************************************************
|
[Model]     mod_sda_pp
Model_type  I/O
Polarity    Inverting
Enable      Active-High
Vinl    =   0.540V
Vinh    =   1.260V
Vmeas   =   0.900V
Cref    =   50pF
|
|
C_comp      0.843pF   0.769pF   0.925pF
|
|
[Model Spec]
Vinl    0.540V  0.486V  0.594V

  ...
```

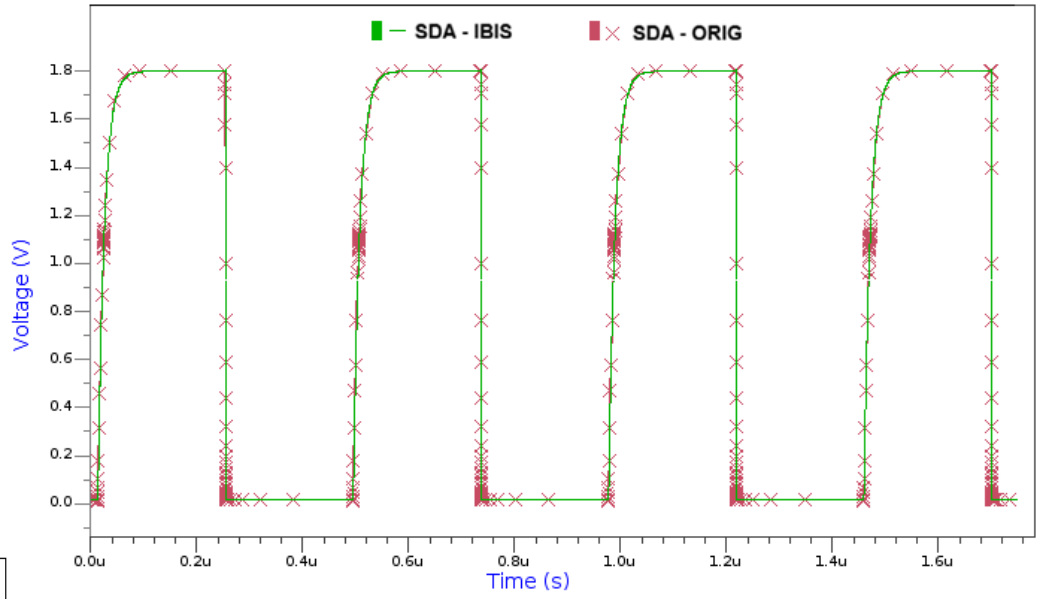
IBIS I/O_open_drain Model

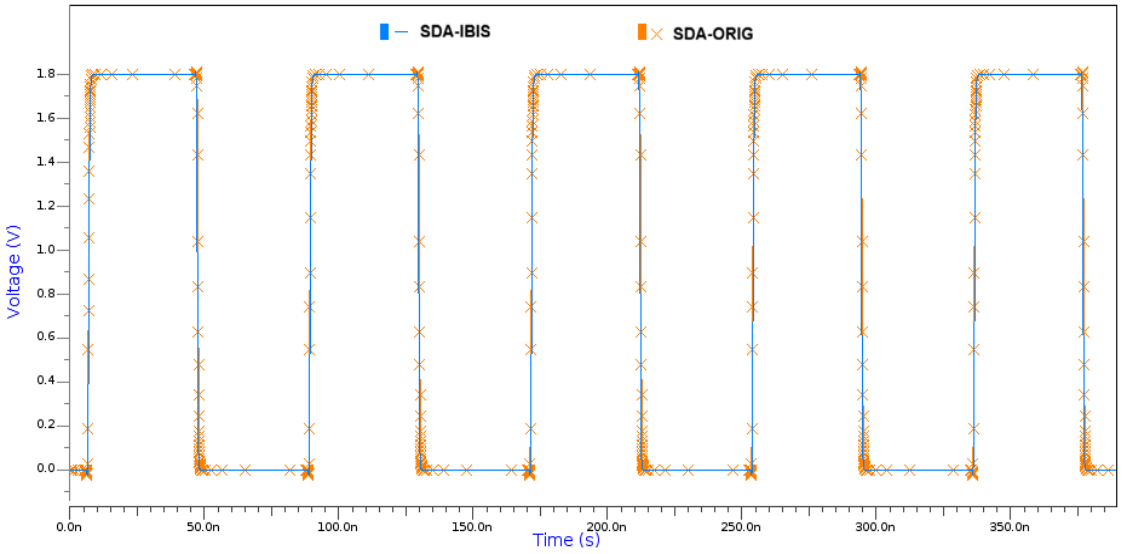
```
|
|************************************************
|            Model mod_sda open_drain
|************************************************
|
[Model]     mod_sda_od
Model_type  I/O_open_drain
Polarity    Inverting
Enable      Active-High
Vinl    =   0.540V
Vinh    =   1.260V
Vmeas   =   0.900V
Cref    =   50pF
Rref    =   10k
Vref    =   1.80V
|
|
C_comp      0.843pF  0.769pF  0.925pF
|
|
[Model Spec]
Vinl    0.540V    0.486V    0.594V
  ...
```

IBIS models vs Original driver

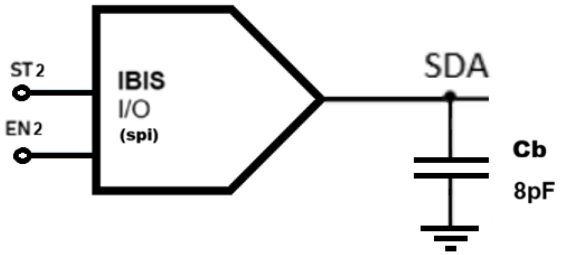
IBIS I/O_open_drain Model



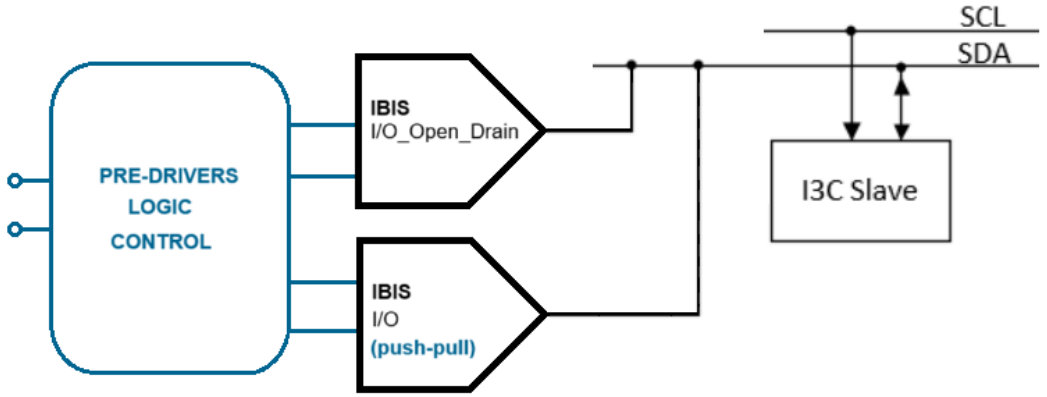




IBIS I/O Model

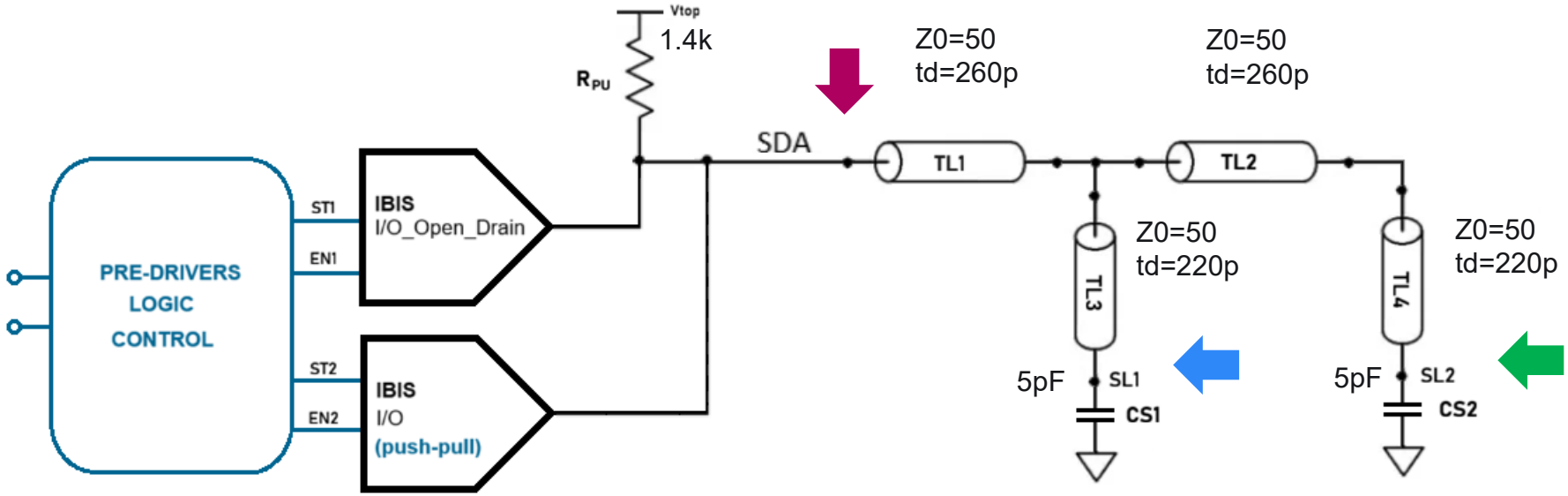


IBIS test-bench for I3C

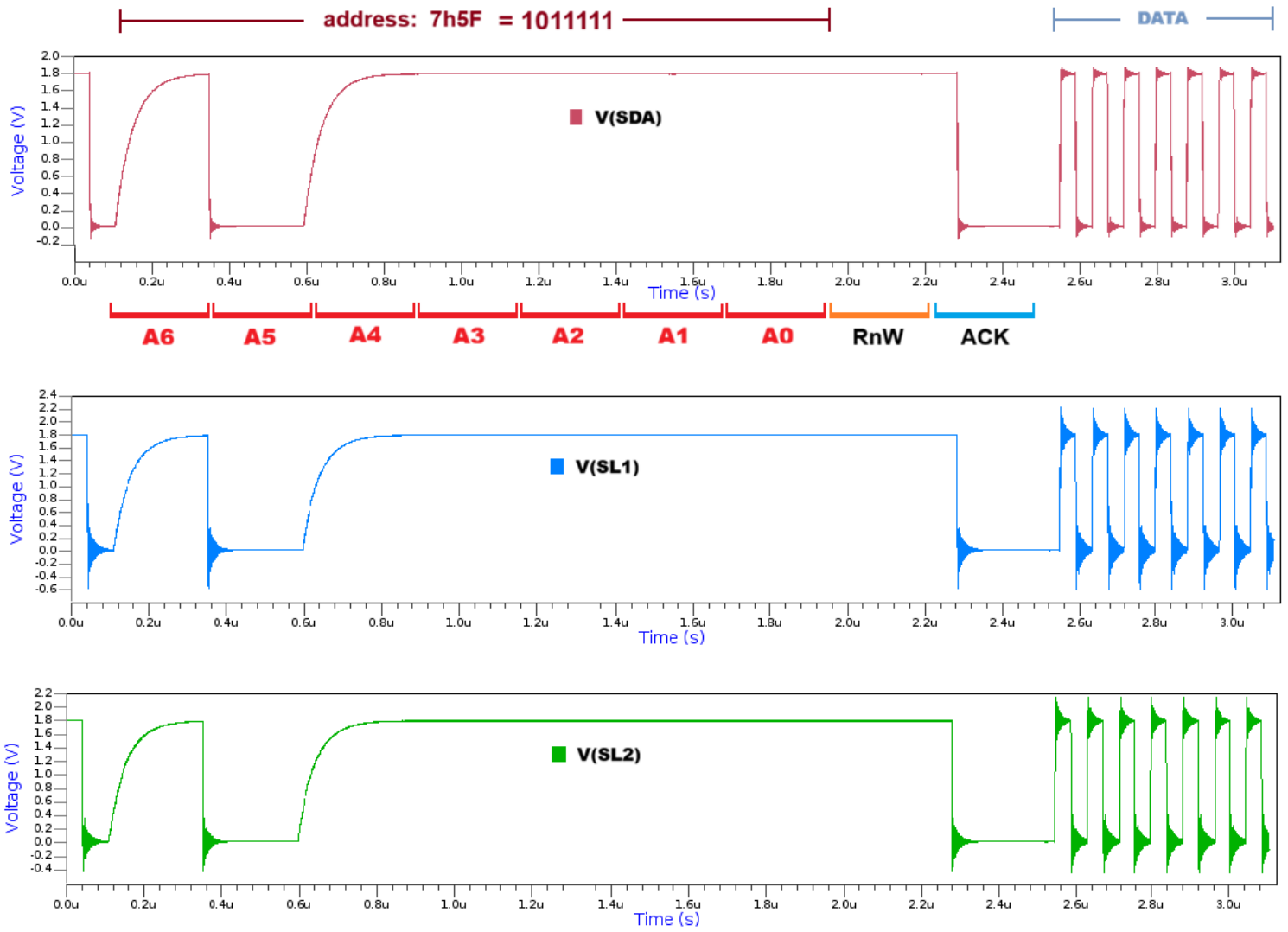


- dynamic model switch -

Consider a more specific test-bench with two slaves:



IBIS test-bench for I3C



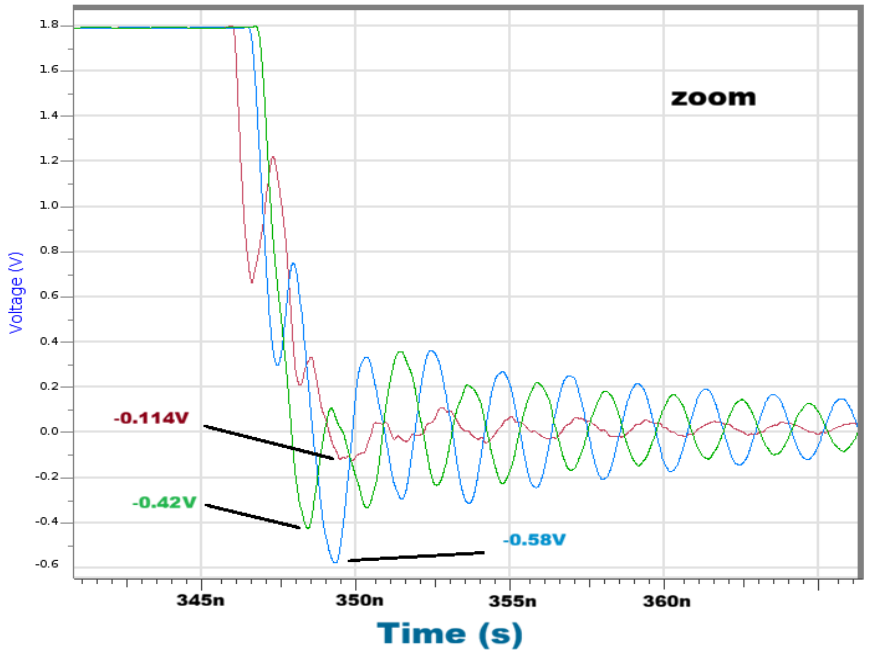
SDA

SL1

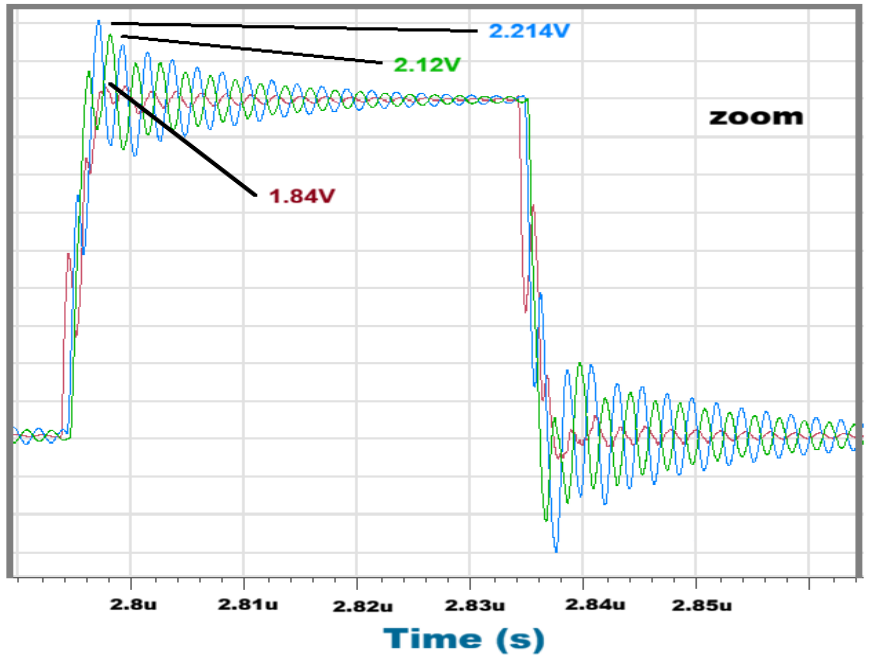
SL2

IBIS test-bench for I3C

address [345n , 365n]



data [2.8u , 2.85u]



Undershoot:

SDA: ΔVu = 0 – (-0.114V) = 0.114V → 6.3%

SL2: ΔVu = 0 – (-0.42V) = 0.42V → 23.3%

SL1: ΔVu = 0 – (-0.58V) = 0.58V → 32.2%

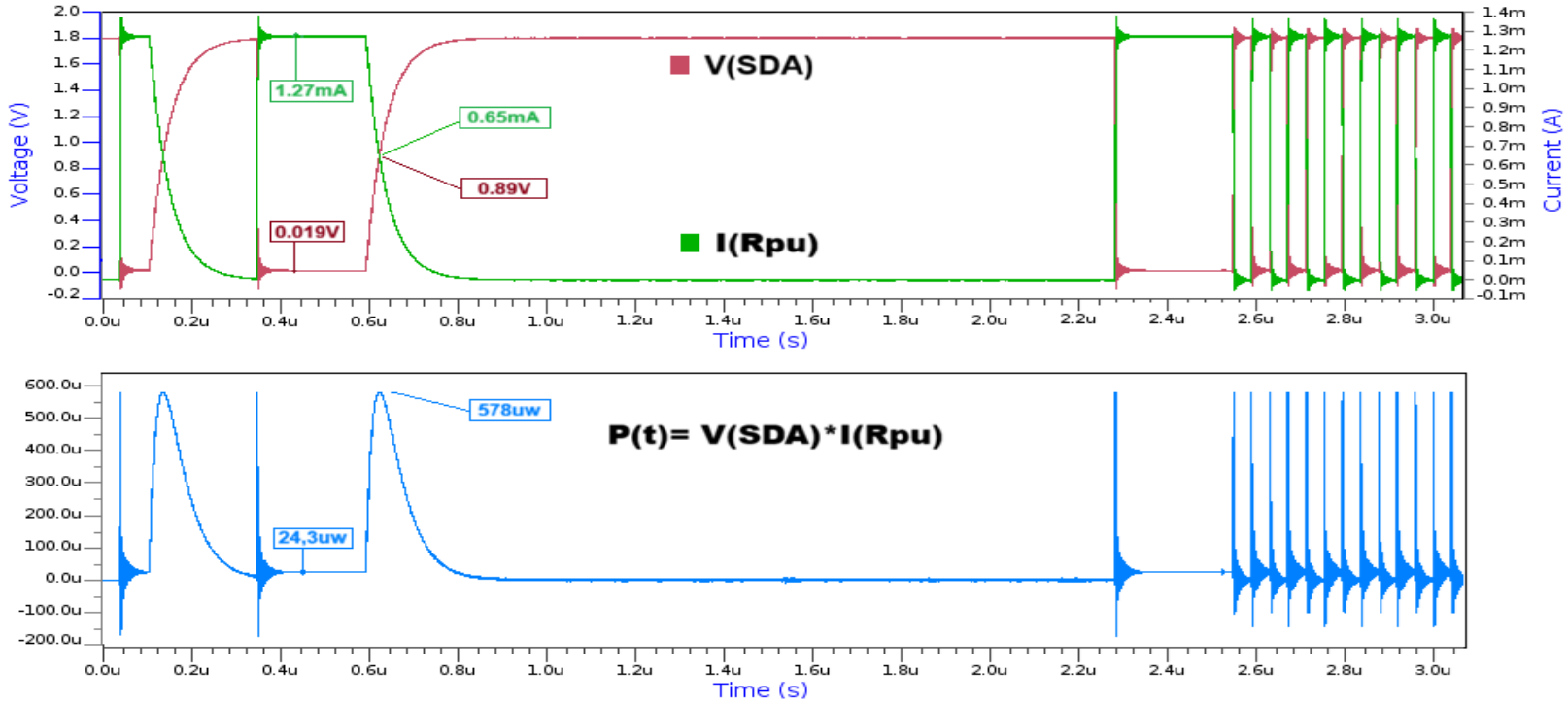
Overshoot:

SDA: ΔVo = 1.84V-1.8V = 0.04V → 2.2%

SL2: ΔVo = 2.12V-1.8V = 0.32V → 17.7%

SL1: ΔVo = 2.214V-1.8V = 0.414V → 23.0%

IBIS power dissipation for I3C

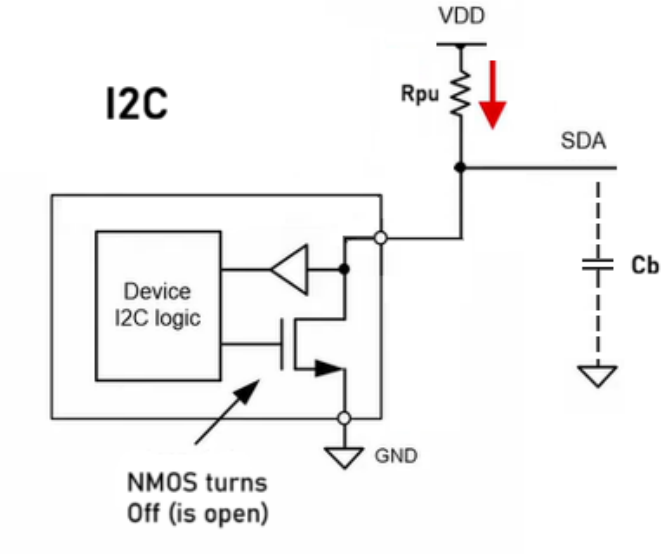


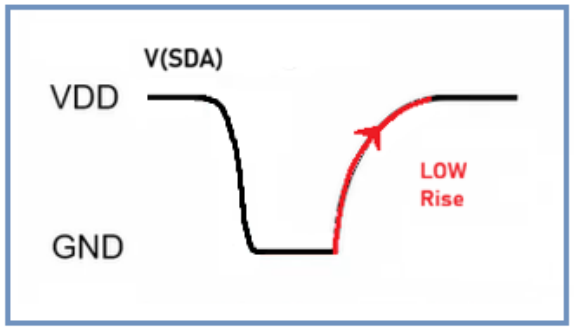
During *Open drain mode* SDA Low Voltage is not zero but 19mV and the current I(Rpu)=1.27mA, so this cause a Static Power (that is around 24uW); The dynamic Power during SDA Rising has a peak of 578uW;

During *I/O mode* SDA Voltage switch from 0V to 1.8V, so here we don't have static Power, only dynamic Power during SDA Rise/Fall edges;

I2C Rpull-up

I2C needs a resistor Rpu to rise signal to VDD:





τ = Rpu * Cb

When SDA il Low:

$$Ipu = VDD / Rpu$$

$$Ppu = \frac{VDD^2}{Rpu}$$

is the Power **wasted** in the Pull-up Resistor

Pull-up Resistance:

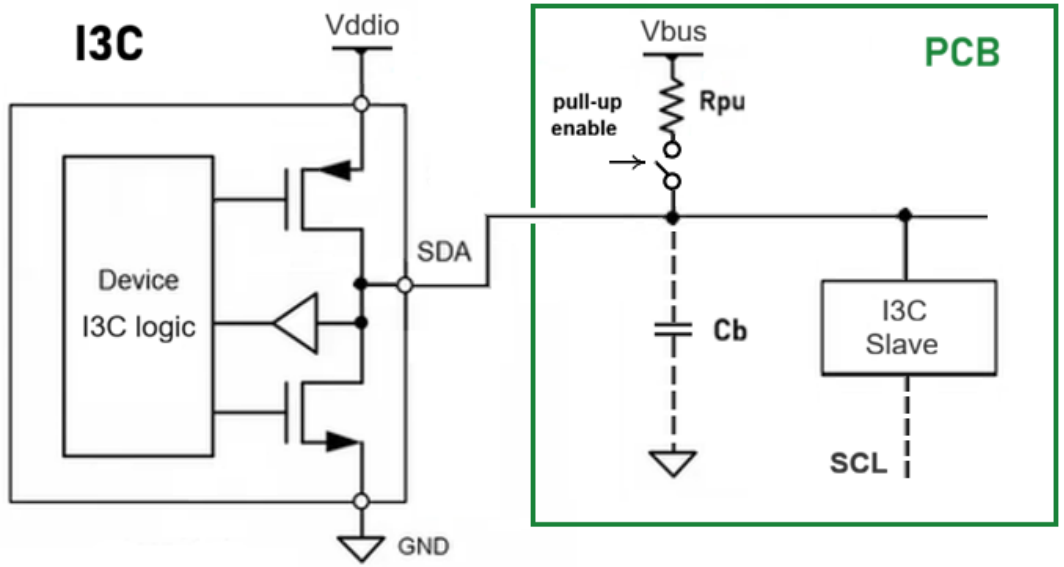
$$Rpu(min) < Rpu < Rpu(max)$$

$$Rpu(min) = \frac{Vdd - V_{OL}}{I_{OL}}$$

$$Rpu(max) = \frac{tr}{0.8473\ Cb}$$

I3C Rpull-up

I3C pull-up the SDA to **High** through a **PMOS (in push-pull mode)**



when I3C is in Master Mode the Rpu can be Enabled (On), if the I3C is in Slave Mode, the Rpu is disabled;

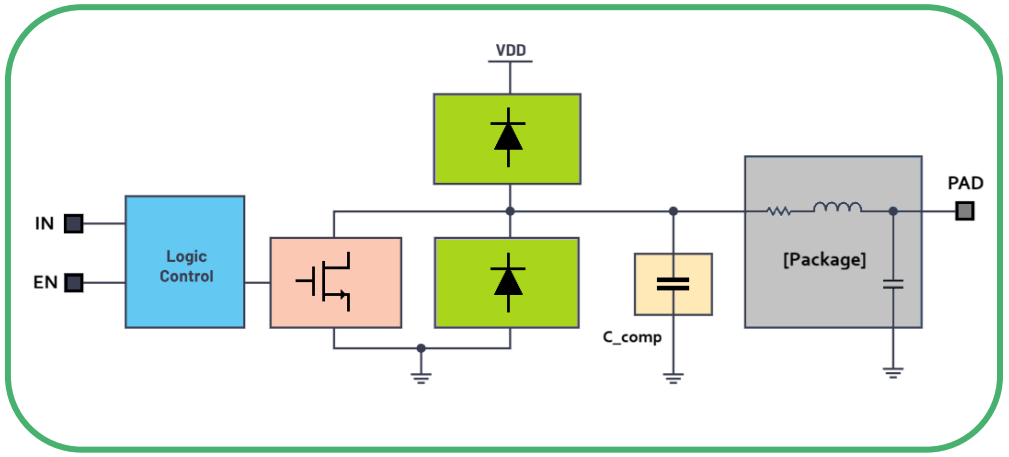
Rpu = Rpu_int

Rpu = Rpu_int // Rpu_ext

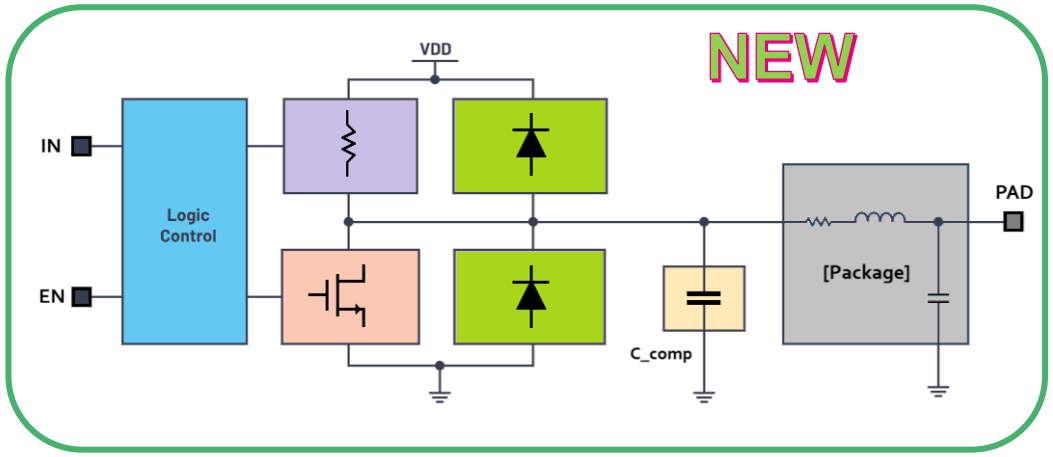
I3C usually have an *Internal* resistance of pull-up, but in base to the **Cb** (total bus capacitance) the Pull-up resistance could be not sufficient; So often *external* resistance is defined in the PCB and allows to have a correct speed during the open drain mode.

IBIS new Model Type (suggestion)

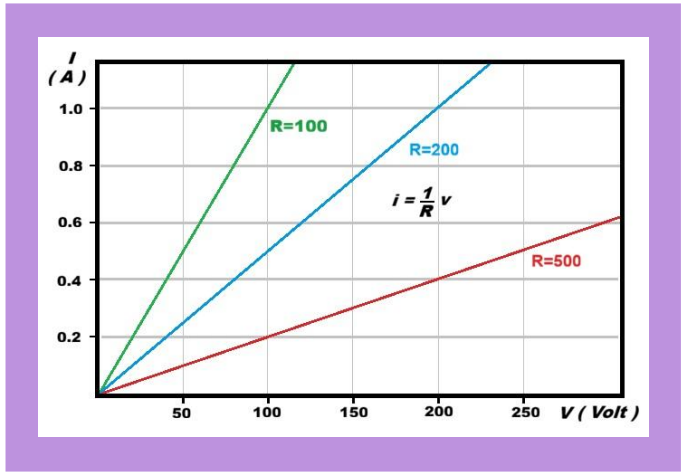
IBIS I/O_open_drain



IBIS I/O_open_drain_rpu

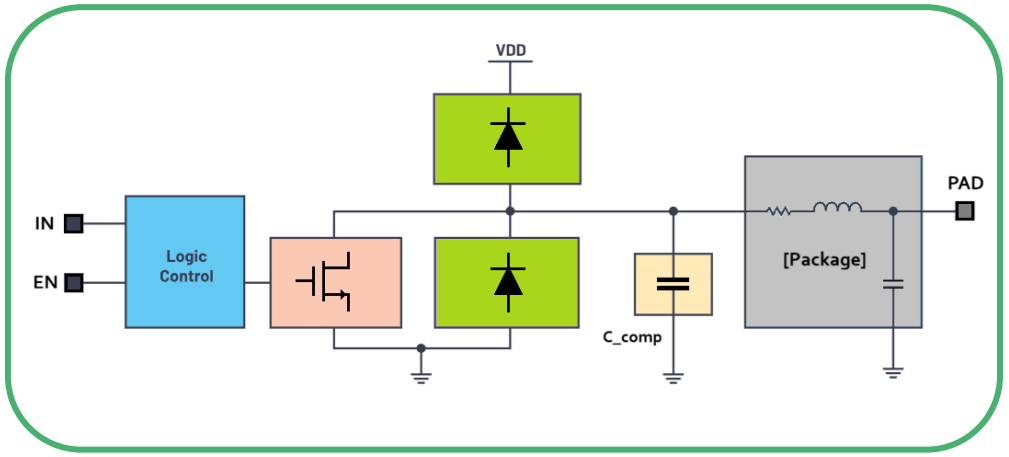


New **Model type**: I/O_open_drain with **Internal Rpu** resistance:
I/O_open_drain_rpu
(or with a set of values)

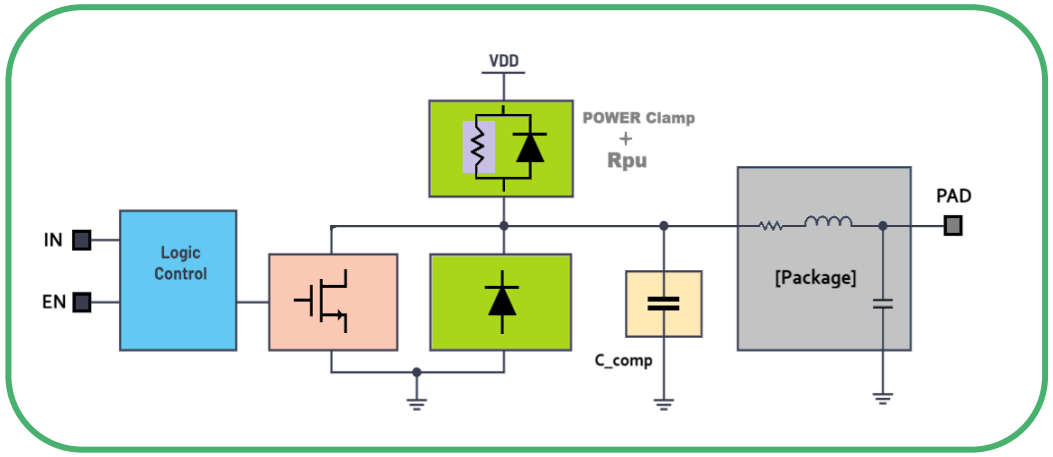


IBIS Model Type (suggestion)

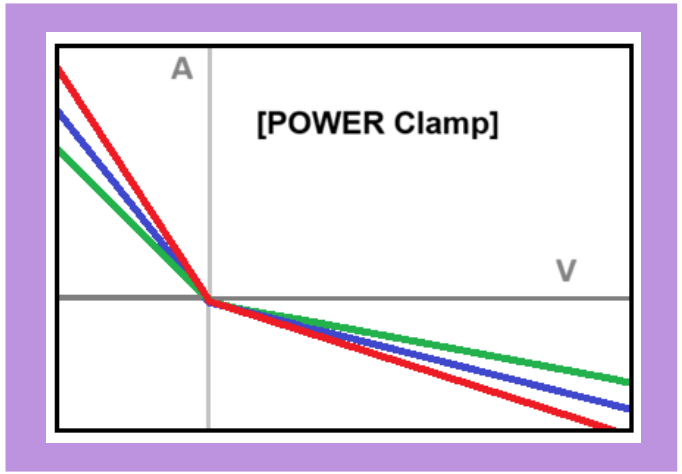
IBIS I/O_open_drain



IBIS I/O_open_drain*

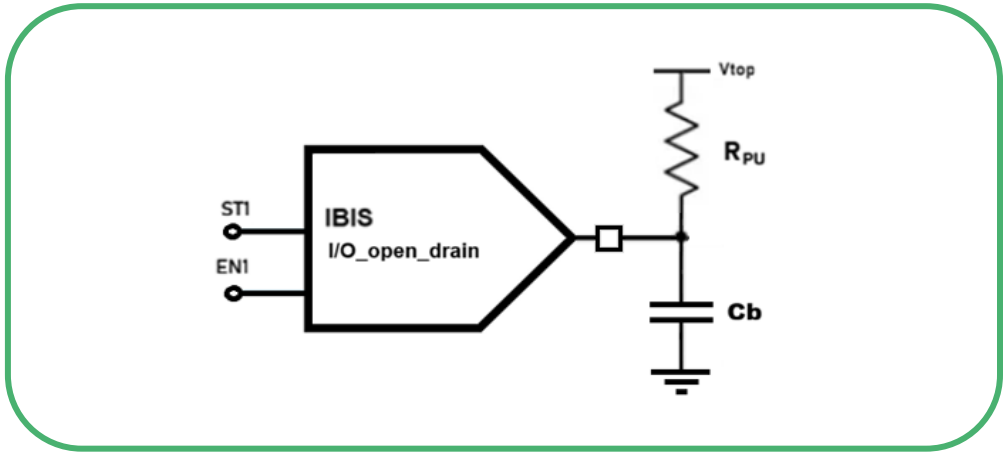


Or **Model type**: I/O_open_drain and insert the **Rpu** resistance into the [POWER Clamp] curve

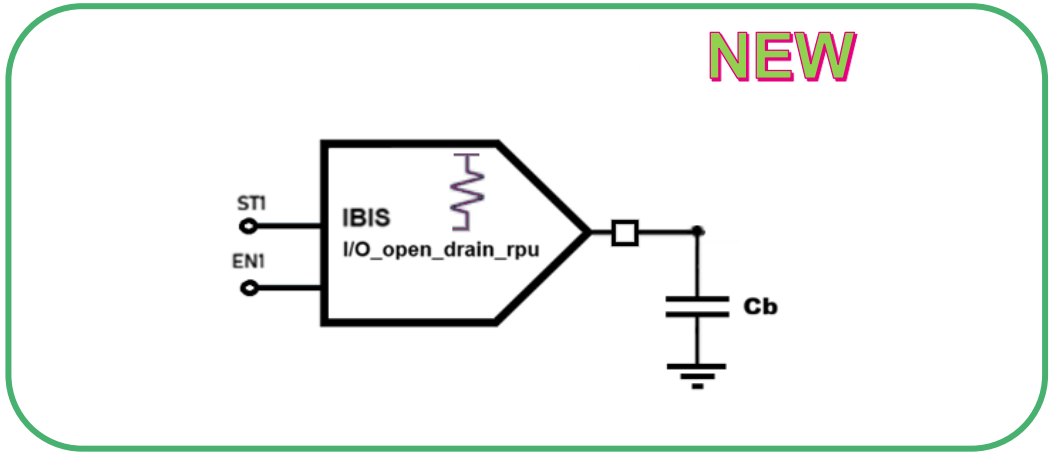


IBIS new Model Type (suggestion)

IBIS I/O_open_drain



IBIS I/O_open_drain_rpu



the **Model type**: I/O_open_drain with **Internal Rpu** resistance allows the user to consider the Output pad directly with the Cb (or the T-line)

Thank you

Fabio Brina

Our technology starts with You

Find out more at www.st.com