Power Management Bus
Implementers Forum

Using The PMBus™ Protocol

Robert V. White
Staff Engineer
Artesyn Technologies
Chair, PMBus™ Specification Working Group
12 October 2005

System Management
Interface Forum

Presentation Overview

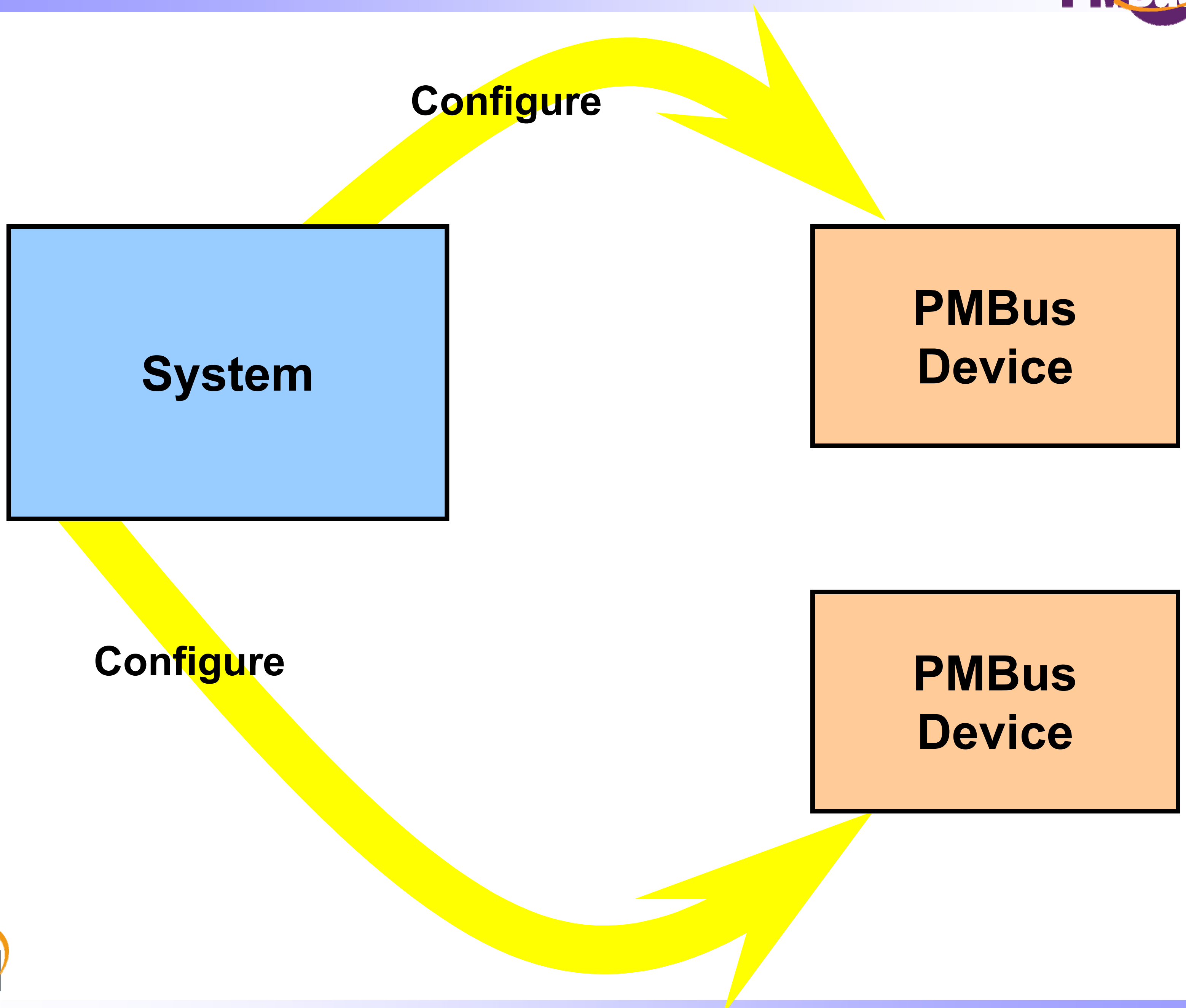
- What Is The PMBus™?
- PMBus Basics
- Using The PMBus In The Lab…
- Implementing PMBus
- Command Language Overview
- Data Formats
- Setting The Output Voltage
- On/Off Control
- Sequencing
- Status Reporting
- Fault Management And Reporting
- Monitoring Voltage, Current And Temperature
- Some Other Topics (As Time Allows)

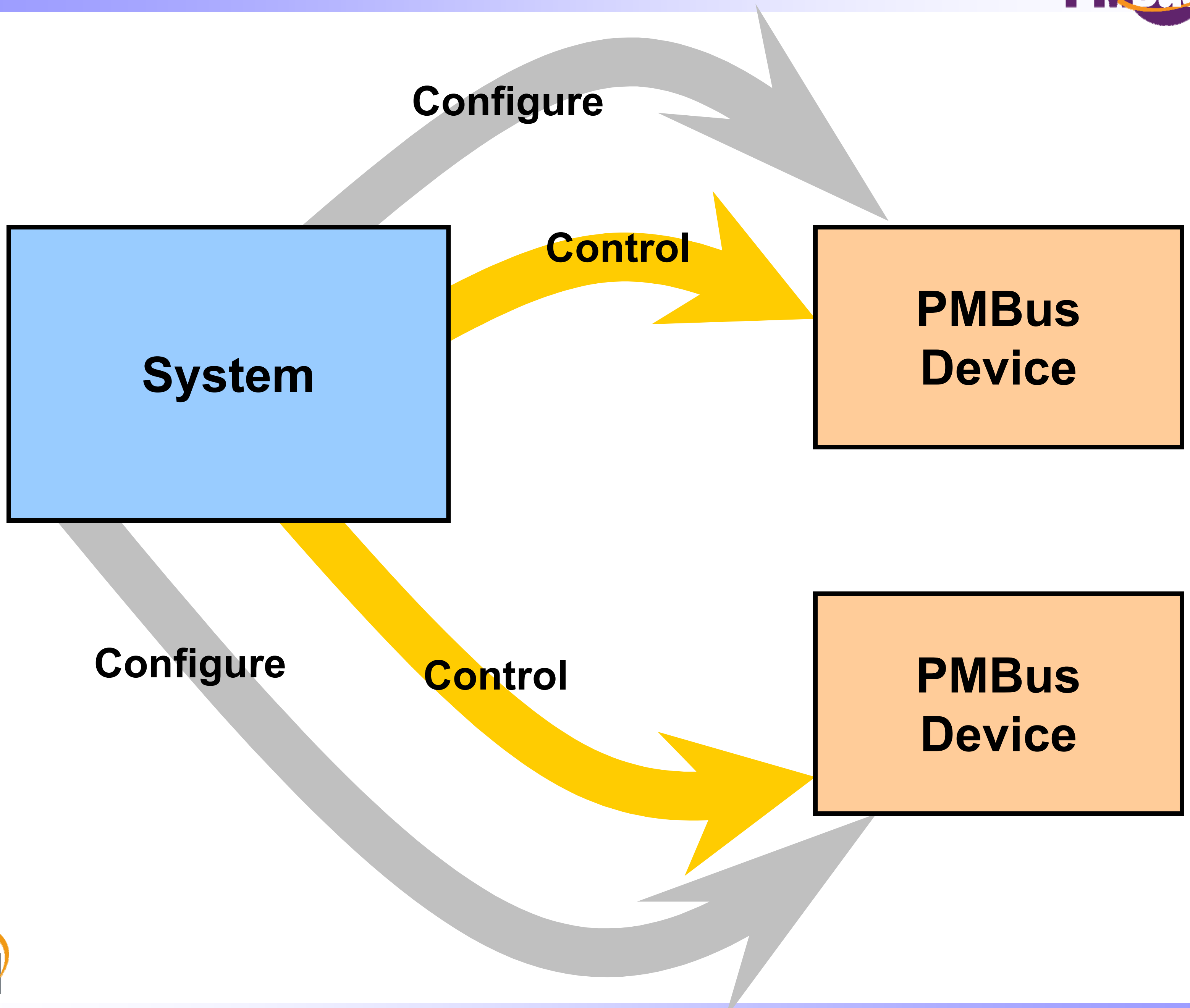
PMBus
Power Management Bus Implementers Forum

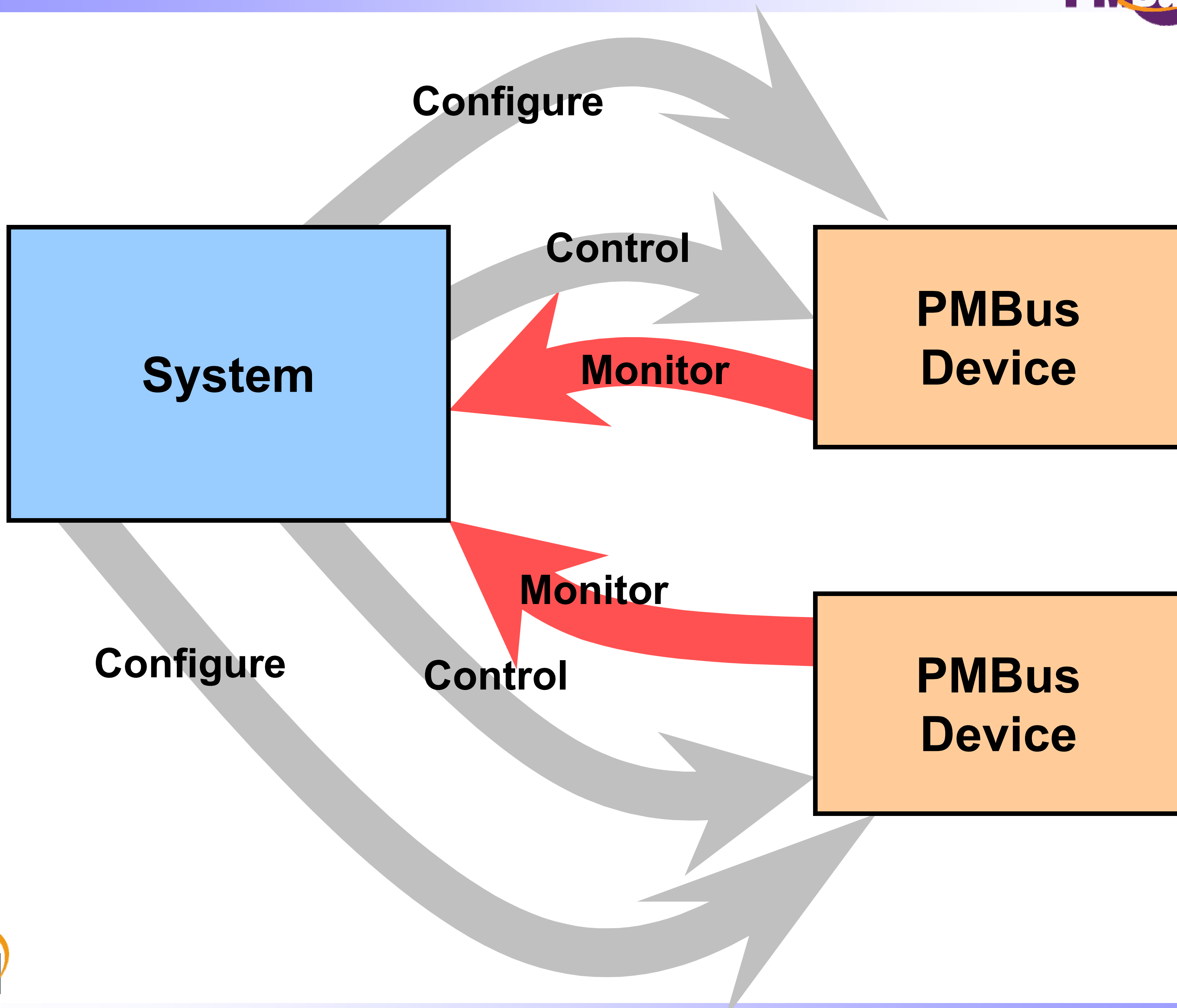
What Is PMBus?

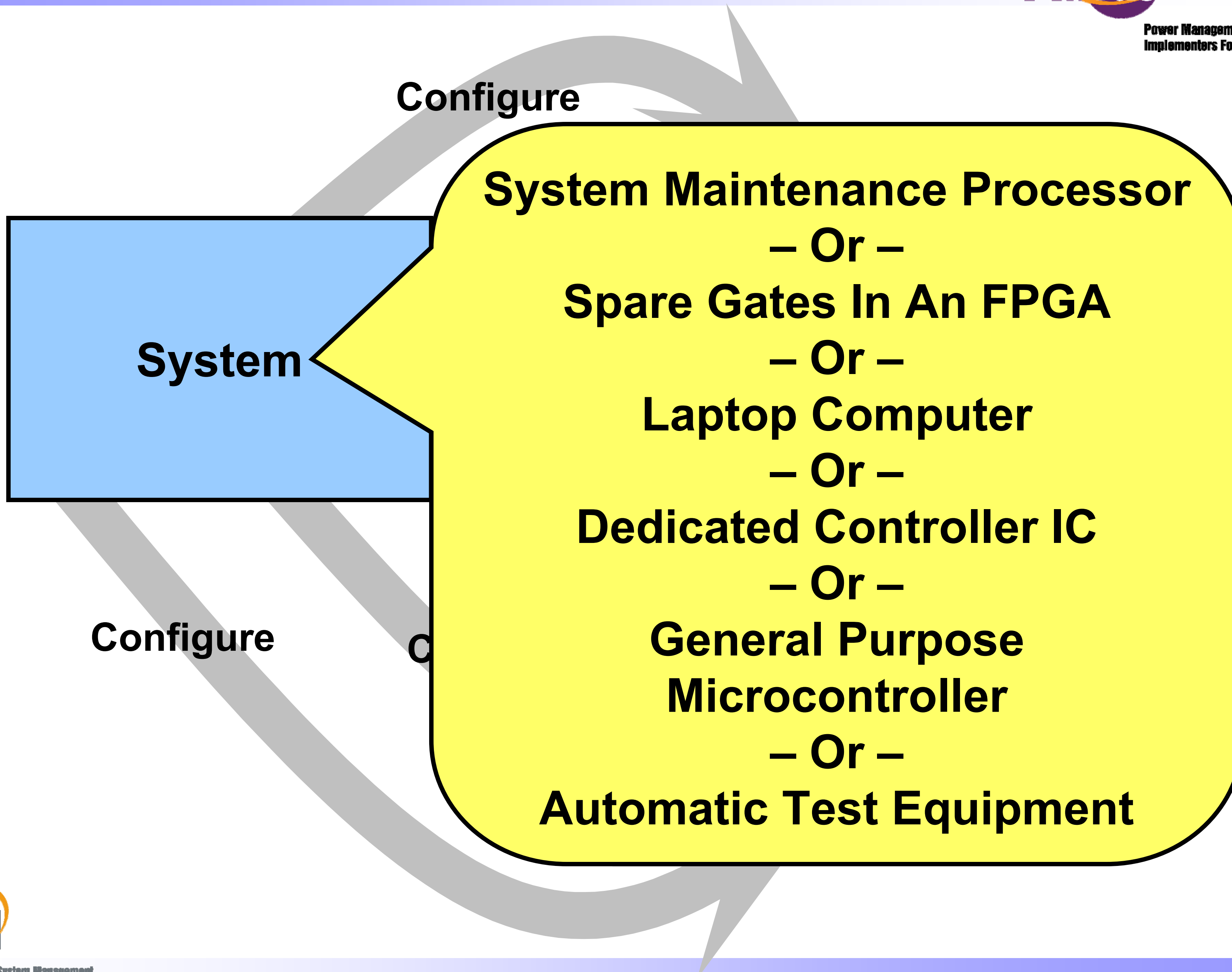
A Standard Way
To Communicate
With Power Converters
Over A Digital
Communications Bus

SMI
System Management Interface Forum


Configure
System
PMBus
Device
Configure
PMBus
Device


Configure
Control
System
PMBus Device
Configure
Control
PMBus Device


Configure
Control
System
Monitor
PMBus
Device
Monitor
Configure
Control
PMBus
Device


Configure
System
Configure
C
System Maintenance Processor
– Or –
Spare Gates In An FPGA
– Or –
Laptop Computer
– Or –
Dedicated Controller IC
– Or –
General Purpose Microcontroller
– Or –
Automatic Test Equipment

PMBus Is An Open Standard

- Owned By The System Management Interface Forum (SM-IF)
  - SM-IF Membership Is Open To All
- Royalty Free
- Released Specifications Freely Available
- Works With All Types Of Power Converters
  - AC-DC Power Supplies
  - Isolated DC-DC And Bus Converters
  - Non-Isolated Point-Of-Load Converters
  - Microprocessor Power Converters

Power Management Bus Implementers Forum

PMBus: What It Is Not

- Not A Product Or Product Line
- Not A Standard For A Power Supplies Or DC-DC Converters
  - No Form Factor, Pin Out, Efficiency, Etc.
  - Alliances Like POLA And DOSA Will Define
- No Converter-To-Converter Communication
  - Such As Current Share And Analog Voltage Tracking
  - Left To The IC And Power Supply Manufacturers
  - Including These Would Inhibit Future Innovation

System Management Interface Forum

Some Basic PMBus Requirements

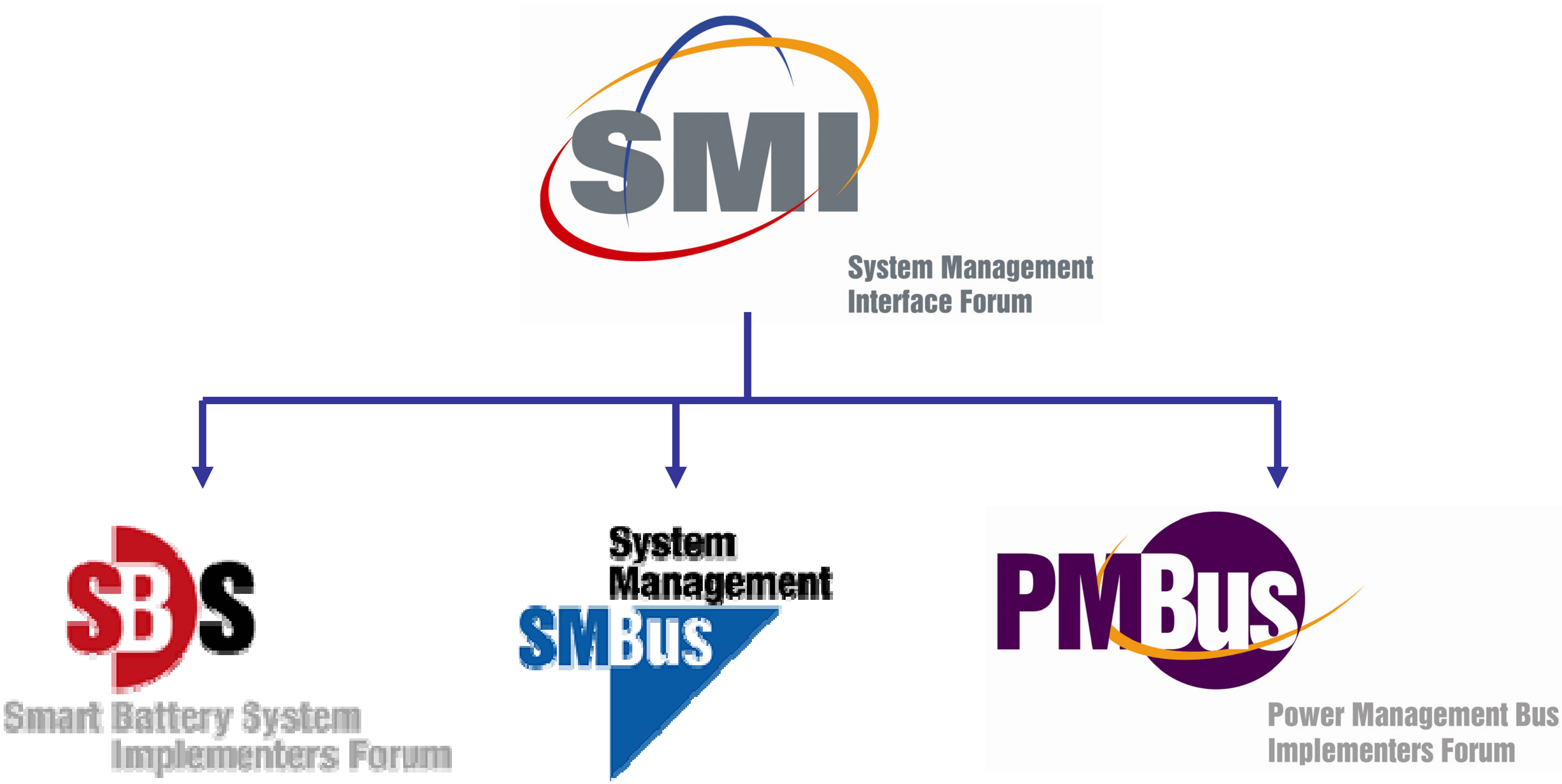
- PMBus Devices Must Start Up Safely Without Bus Communication
- PMBus Devices Can Be Used With Or Without A Power System Manager/Controller
- PMBus Devices Support "Set And Forget"
  - Can Be Programmed Once At Time Of Manufacture
  - Then Operate Forevermore Without Bus Communication
- Defaults From Either/Or
  - Non-Volatile Memory
  - Pin Programming

Who Is PMBus?

PMBus Adopters

- Alliance Semiconductor
- Artesyn Technologies
- Emerson/Astec
- International Rectifier
- Intersil Corporation
- Magnetek, Inc.
- Micro Computer Control Corporation (MCC)
- Microchip Technology
- Primarion
- Silicon Laboratories
- Summit Microelectronics, Inc.
- Texas Instruments
- Tyco Electronics Corp.
- Volterra Semiconductor Corporation
- Zilker Labs

System Management Interface Forum, Inc.



System Management Interface Forum, Inc.

SM-IF Membership Open To Any And All

SMI
System Management Interface Forum

SBS
Smart Battery System Implementers Forum

System Management SMBus

PMBus
Power Management Bus Implementers Forum

www.powerSIG.org

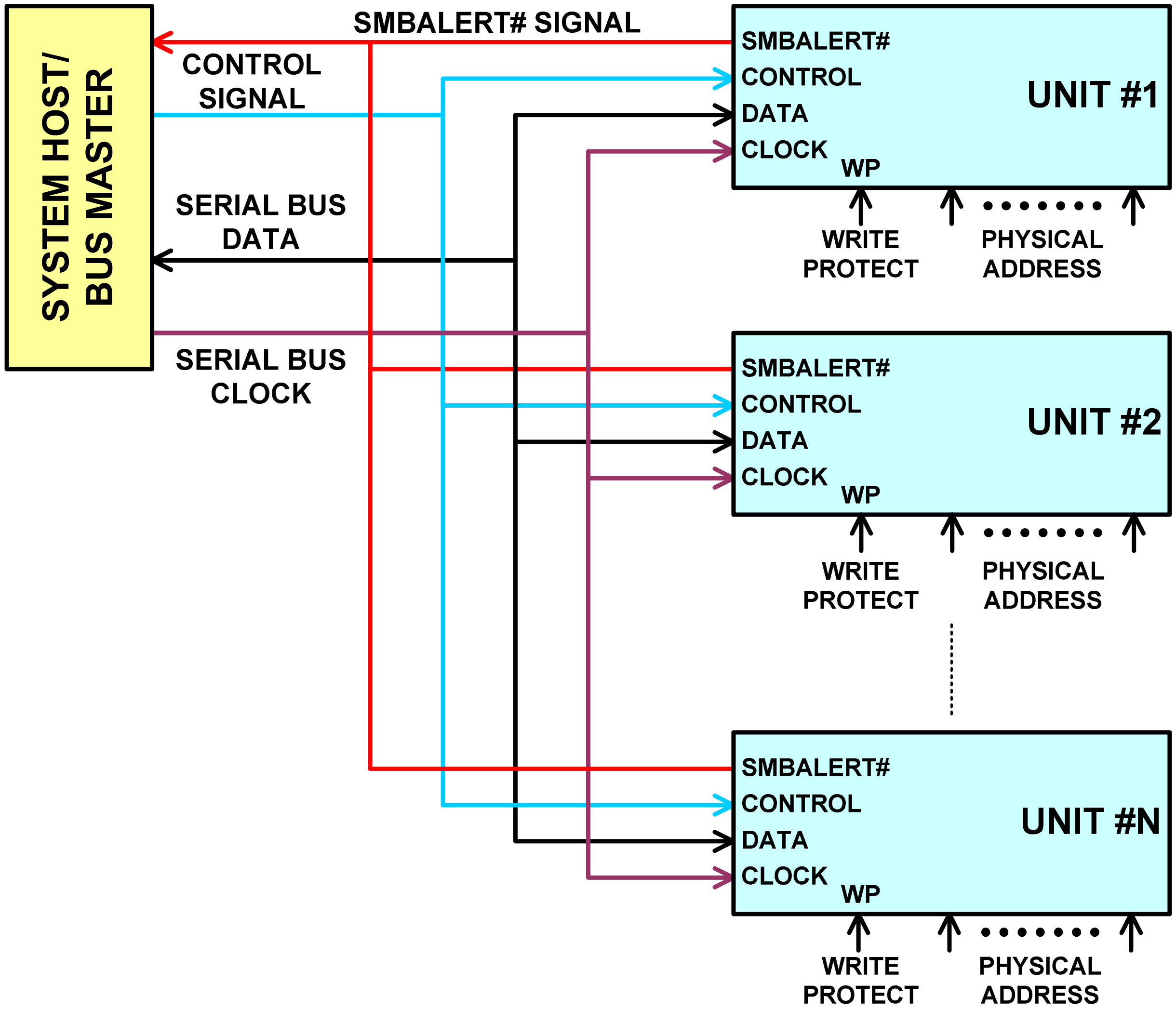
Use Of PMBus™ Logo And Trademark

- Only Adopters Are Permitted To Use The PMBus™ Trademarks And Logo For Commercial Purposes
  - Commercial Purpose Is Anything Related To The Sale Of Products And Services
  - Helps Assure That PMBus Device Manufacturers Understand The Specification
- The Press May Use The Trademarks And Logo In Articles That Do Not Promote Products Or Services

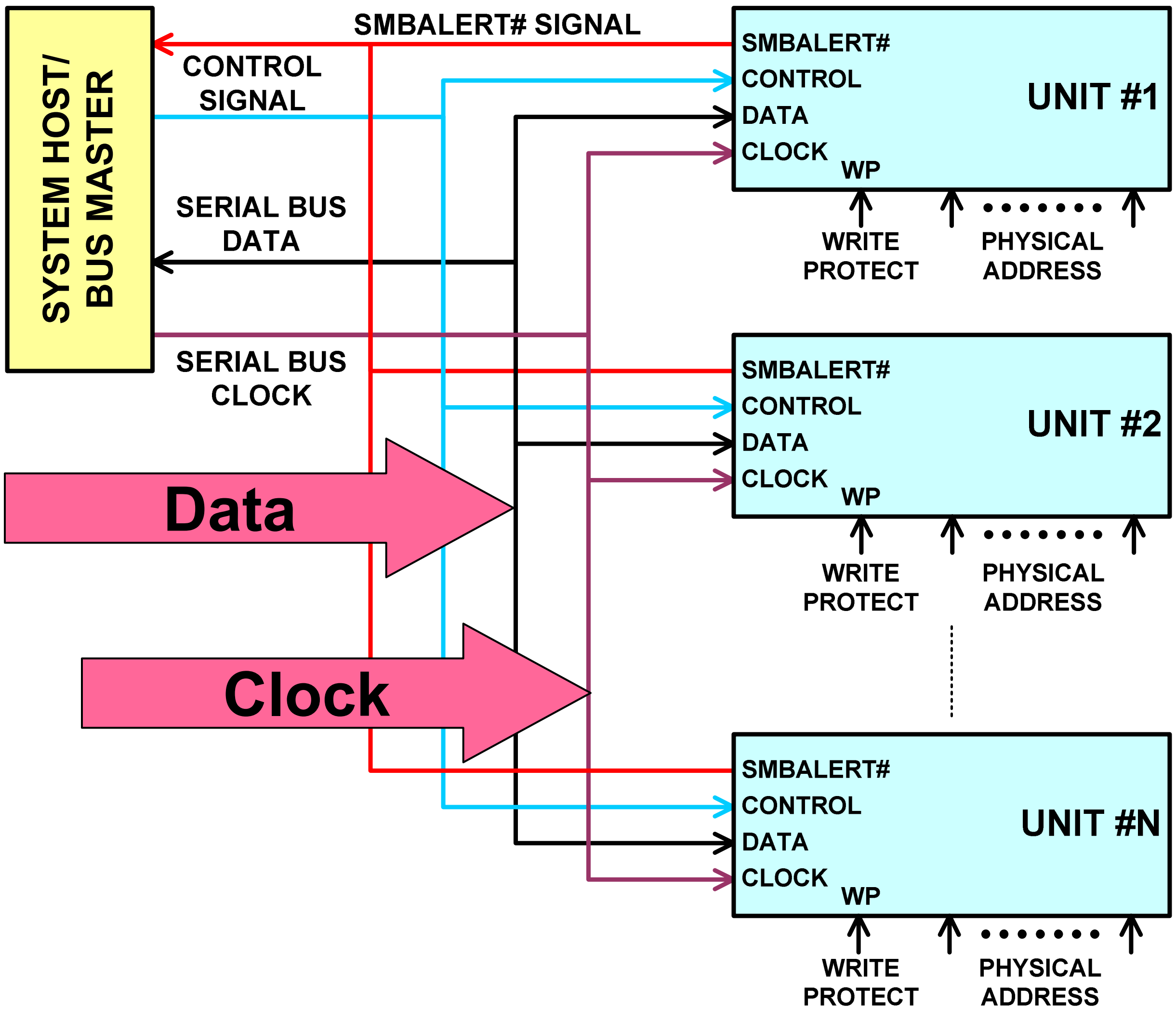
Specification Structure

- Part I – Physical Layer And Transport
  - Bus & Protocols
  - Discrete Signals
  - Electrical Levels
- Part II – Command Language
  - Commands
  - Data Formats
  - Fault Management "Tutorial"
  - Status Reporting "Tutorial"
  - Information Storage

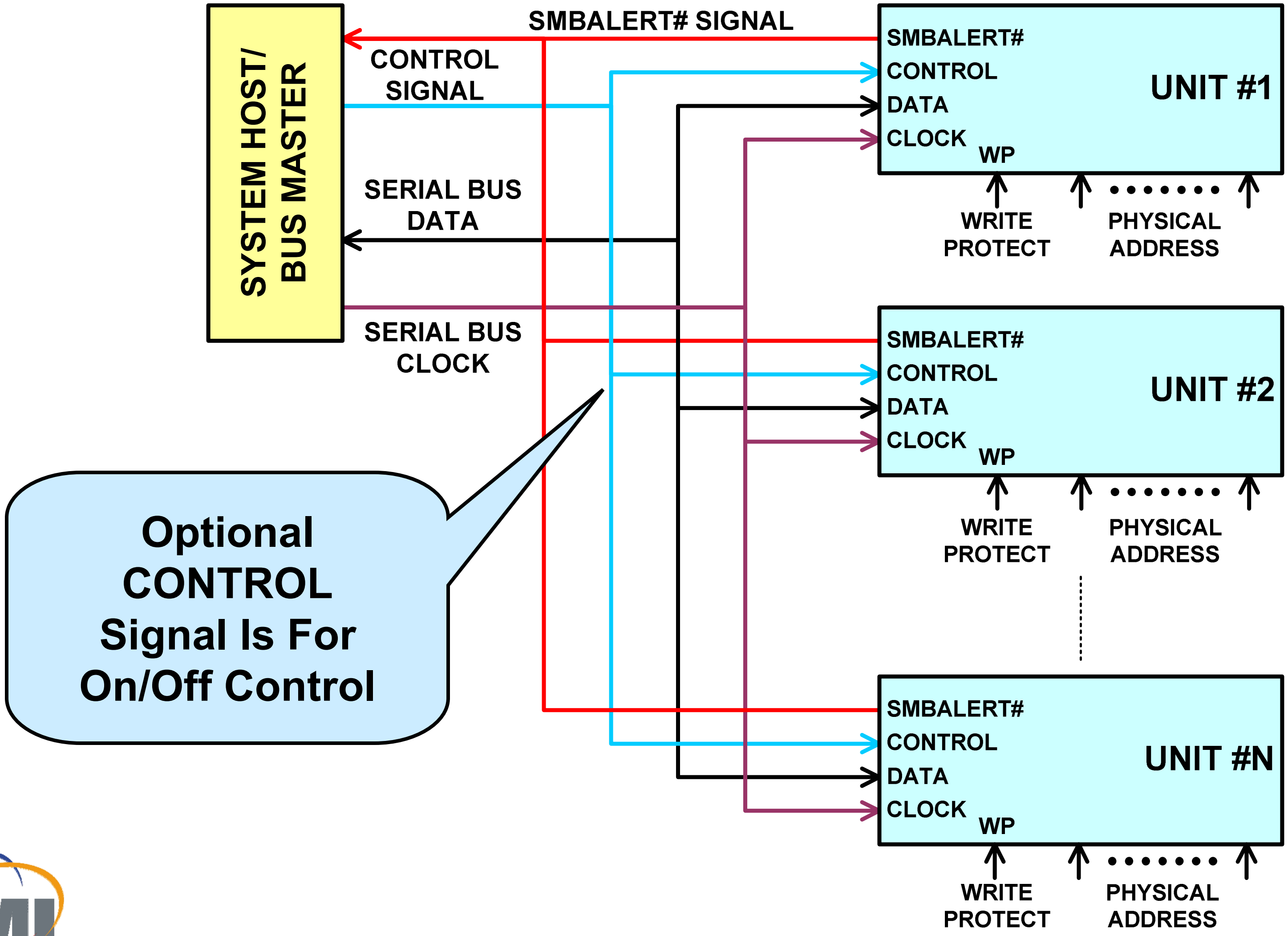
PMBus™ Connections



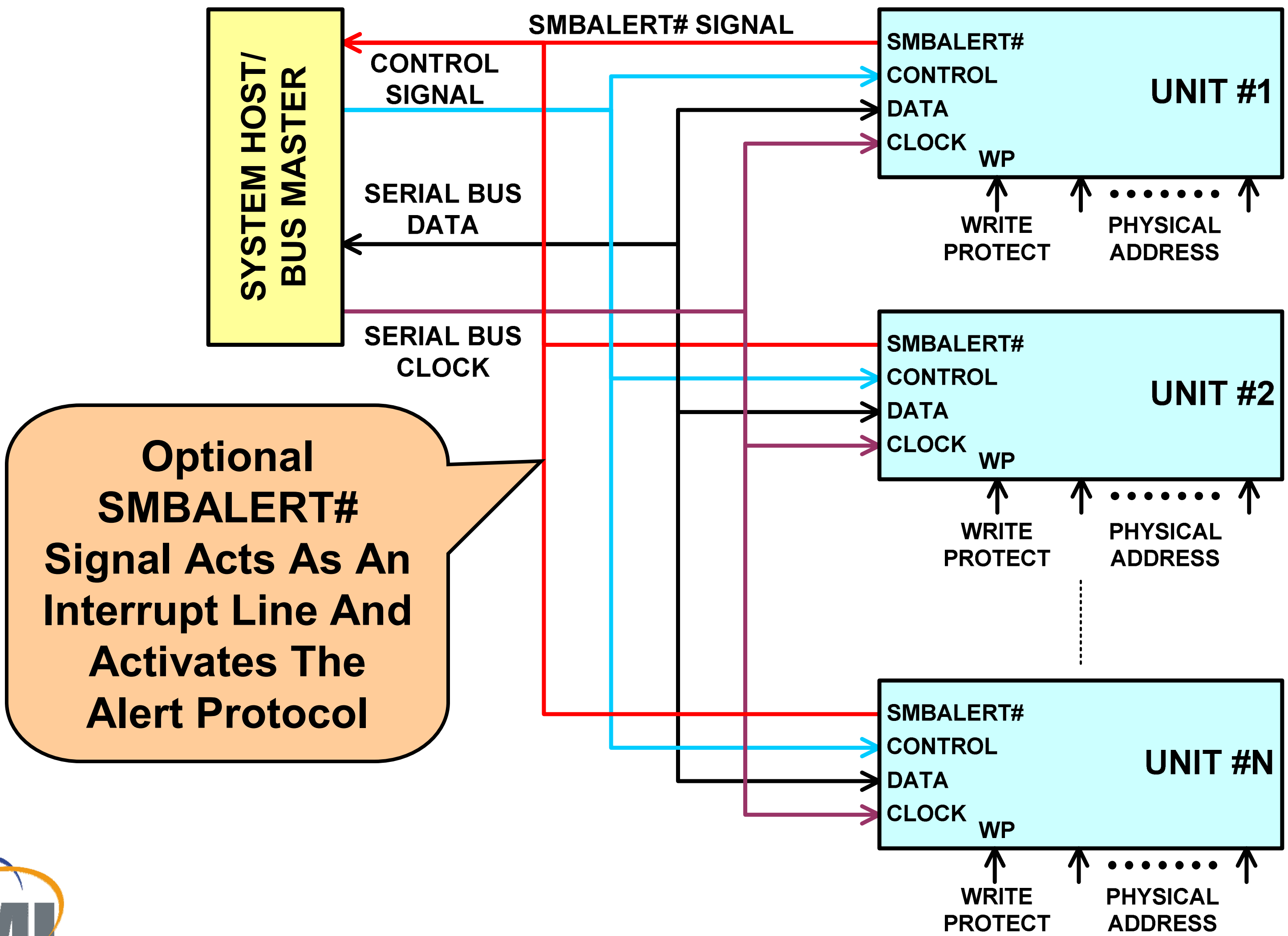
PMBus™ Connections



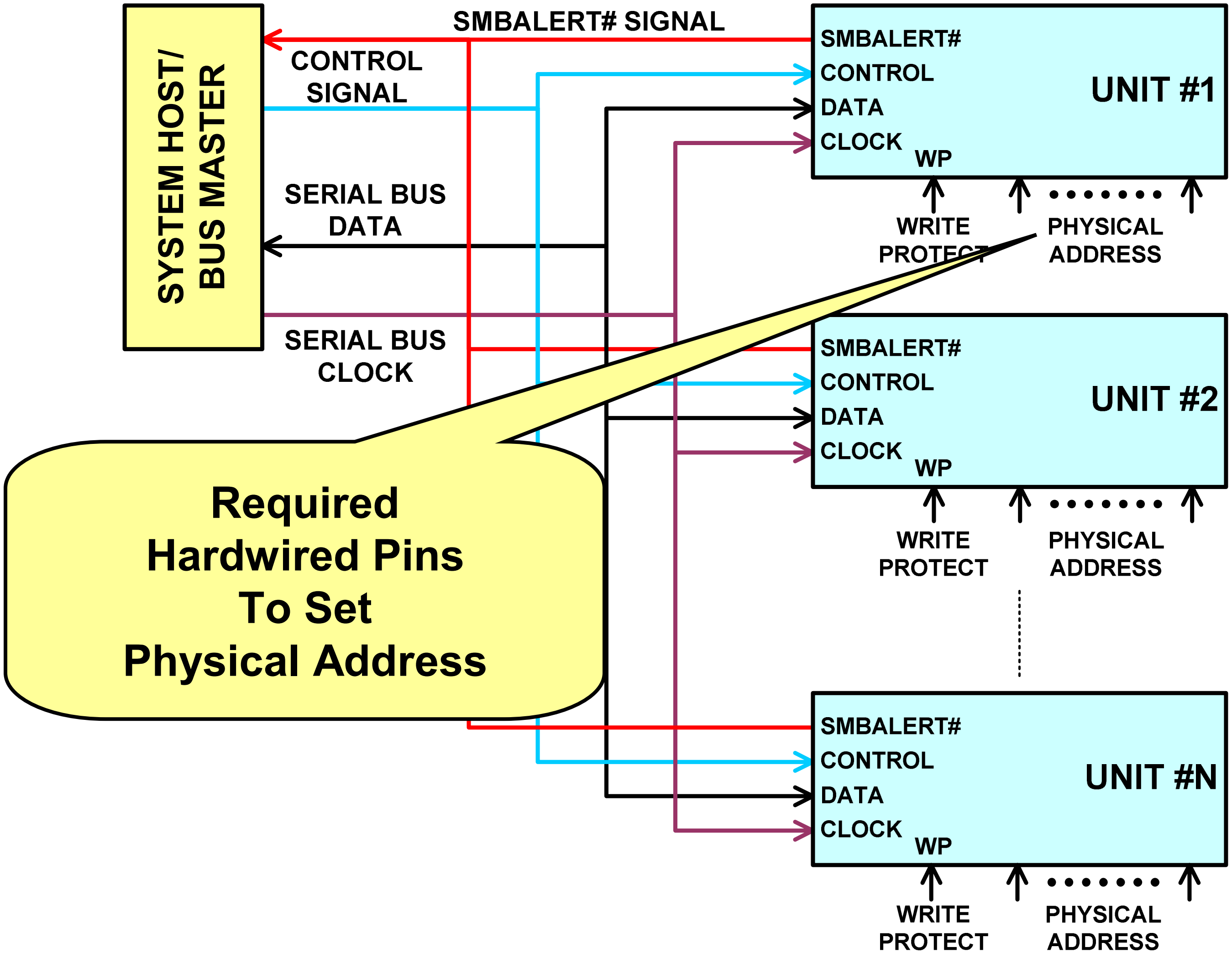
PMBus™ Connections



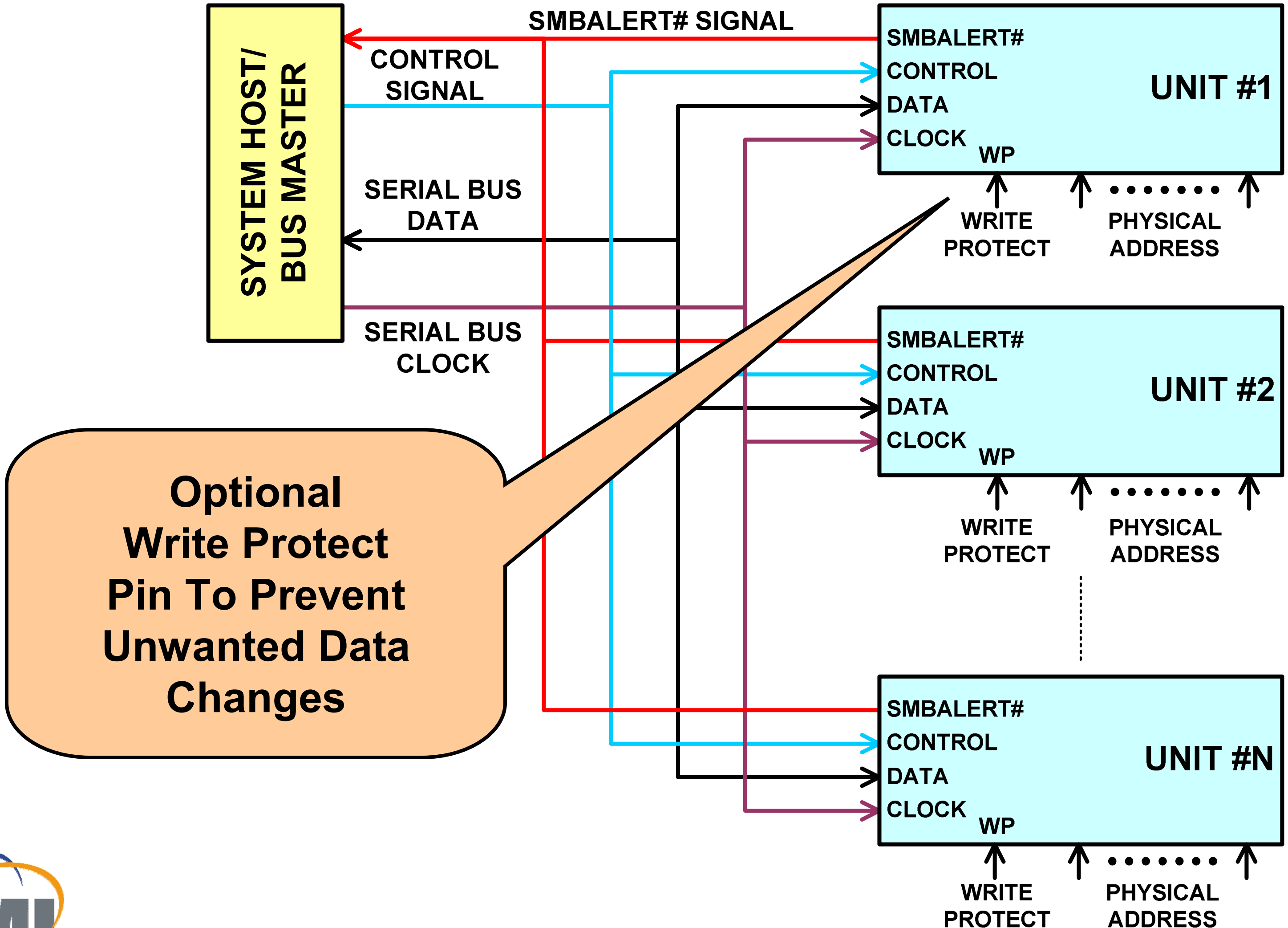
PMBus™ Connections



PMBus™ Connections



PMBus™ Connections



What Is SMBus?

- A Long Existing Standard Bus
- Similar To I²C
  - Synchronous (Clock And Data Lines)
  - Byte Oriented
  - Same Addressing Scheme
  - Same Transmission Control
    - START, STOP, ACK, NACK
- Did Not Require Royalties To Philips

Why SMBus?

- Low Cost Like I²C
- More Robust Than I²C
- More Features Than I²C
  - SMBALERT# Line For Interrupts
  - Packet Error Checking (PEC)
  - Host Notify Protocol
- Generally Electrically Compatible With I²C
- Widely Used In Personal Computers And Small To Medium Servers

SMBus Improvements

- I²C "Noise Sensitivity" – Edge Triggering
  - False START: Timeouts Force Reset
  - False STOP: PMBus Devices Detect Failed Transmissions As Faults
- I²C "Noise Sensitivity" – Corrupt Data
  - Data Rates Permit Digital Filtering
  - Packet Error Checking (PEC)
  - Every Value That Can Be Written Can Be Read

SMBus Improvements

- I^2C Slave Device Hangs Bus
  - Timeouts Force Device Reset
- I^2C Requires Retrieving Device Information By Polling
  - SMBALERT# Line Acts As An Interrupt
  - Automatic, Lossless Bitwise Arbitration Of Simultaneous Requests
- I^2C: 8 Devices Max Of One Type On A Bus
  - No Central Address Control Bureaucracy
  - Over 100 Device Addresses Available

SMBus Limitations

- SMBus and PMBus Specifications Say 100 kHz
  - I²C Says 400 kHz – Which Is Possible If SMBus Setup And Hold Times Are Obeyed
- Capacitance Is A Concern
  - No Explicit Maximum
  - Excessive Capacitance Causes A Violation Of Bus Timing By Slowing Rise Times
  - Minimize Capacitance In Layout
    - Stubs And Branches Not A Concern
  - See SMBus Specification For Details

SMBus Limitations

- SMBus and PMB
  100 kHz
  - I^2C Says 400 kHz
    SMBus Setup And Hold Times Are Obeyed
- Capacitance Is A Concern
  - No Explicit Maximum
  - Excessive Capacitance Causes A Violation Of Bus Timing By Slowing Rise Times
  - Minimize Capacitance In Layout
    - Stubs And Branches Not A Concern
  - See SMBus Specification For Details

PMBus Going To 400 kHz In Revision 1.1

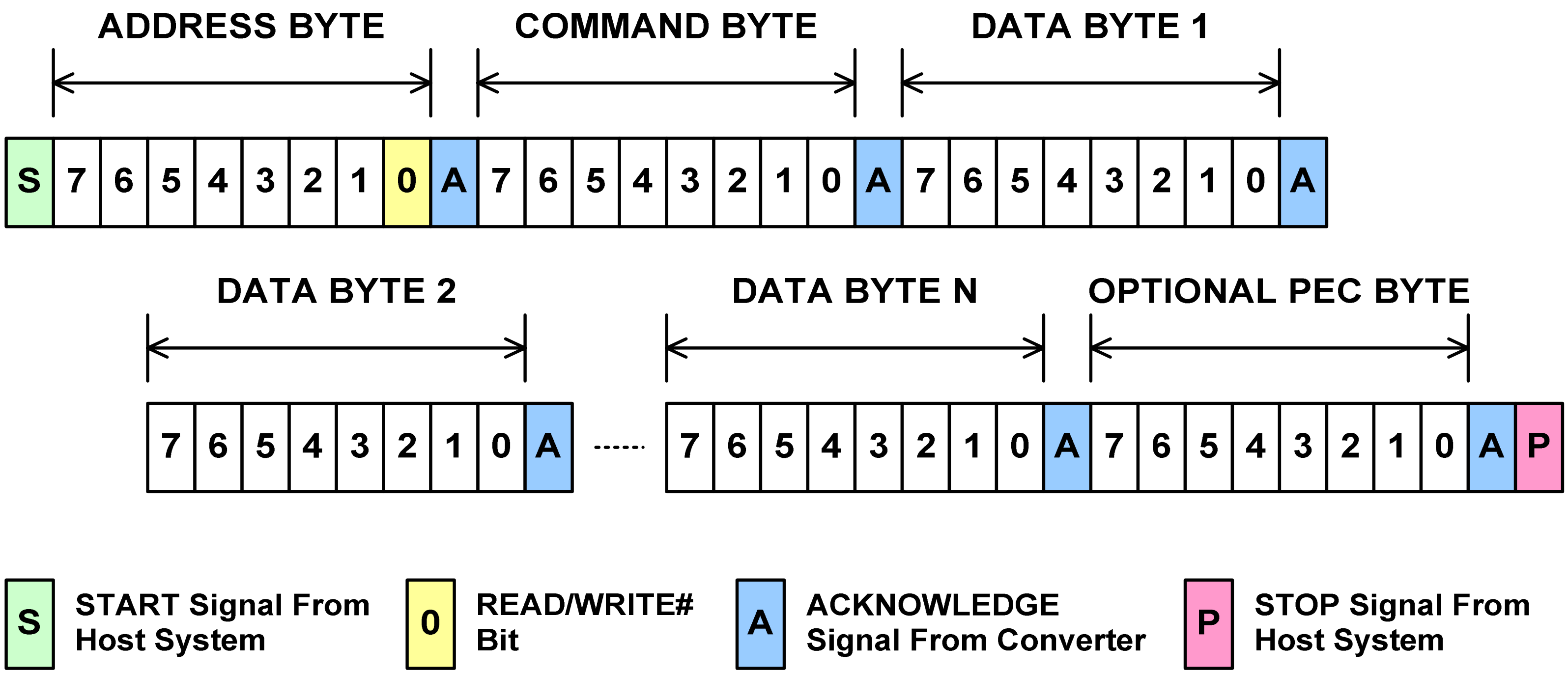
Addressing

- PMBus Devices Use A 7 Bit Address Per The SMBus Specification
  - Provides More Than 100 Possible Device Addresses After Allowing For Reserved Addresses
- No I²C Style Address Control Assignments Or Limitations
- PMBus Users Can Expect Device Addresses To Be Set By A Mix Of:
  - Hardwired Address Pins
  - High Order Address Bits Set By The PMBus Device Manufacturer

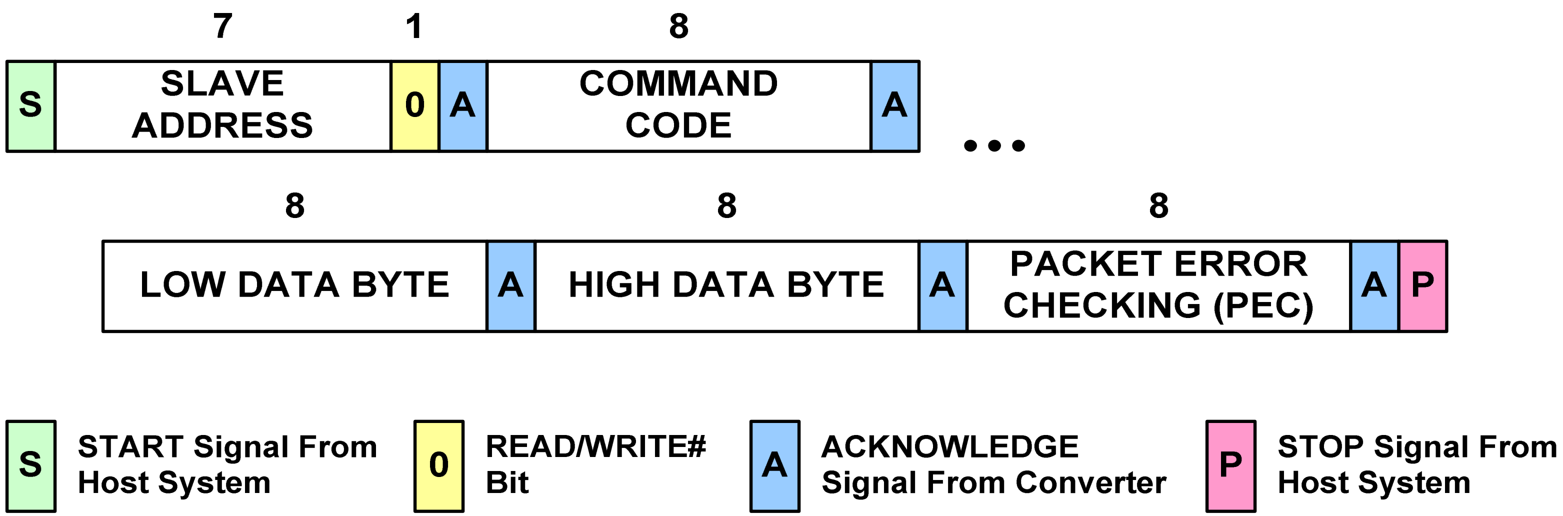
Addressing (cont'd)

- PMBus Device Manufacturers Will Trade Off Cost Of Pins Versus Address Flexibility
- Expect Device Makers To Offer Tri-State Pins Or Resistor Value Programming
- Examples Of The Possibilities
  - 3 Tri-State Pins => 27 Addresses
  - 1 Resistor Programmed Pin => 16–32 Addresses

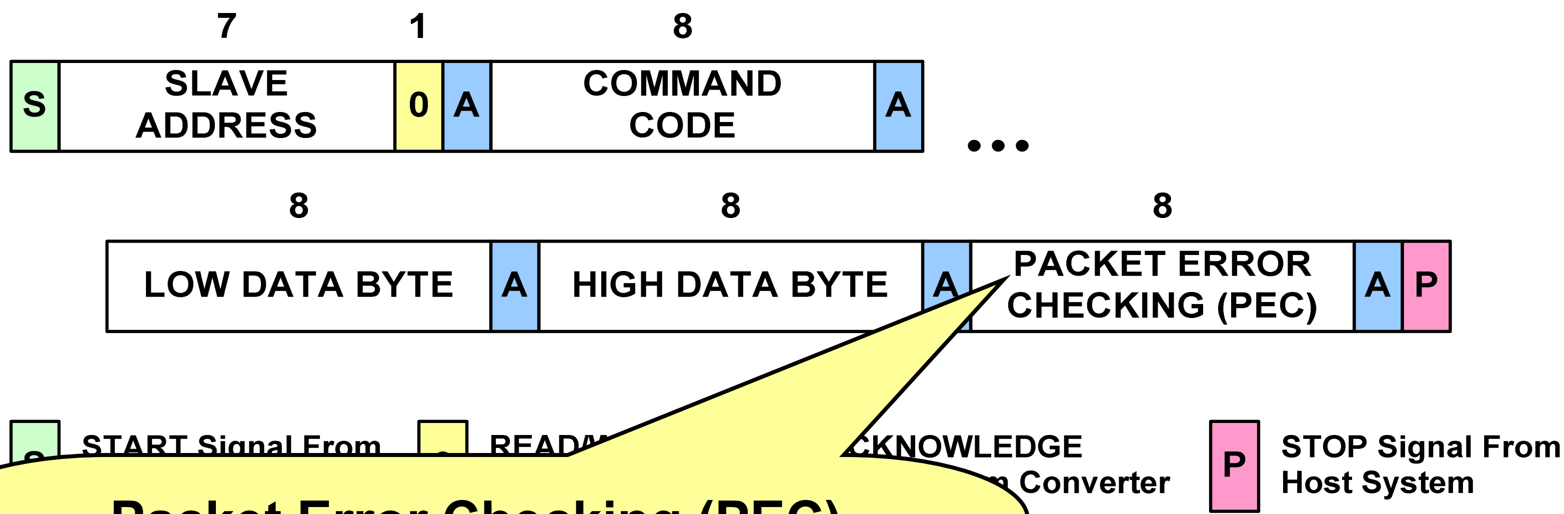
Basic Packet Structure



Write Word Packet



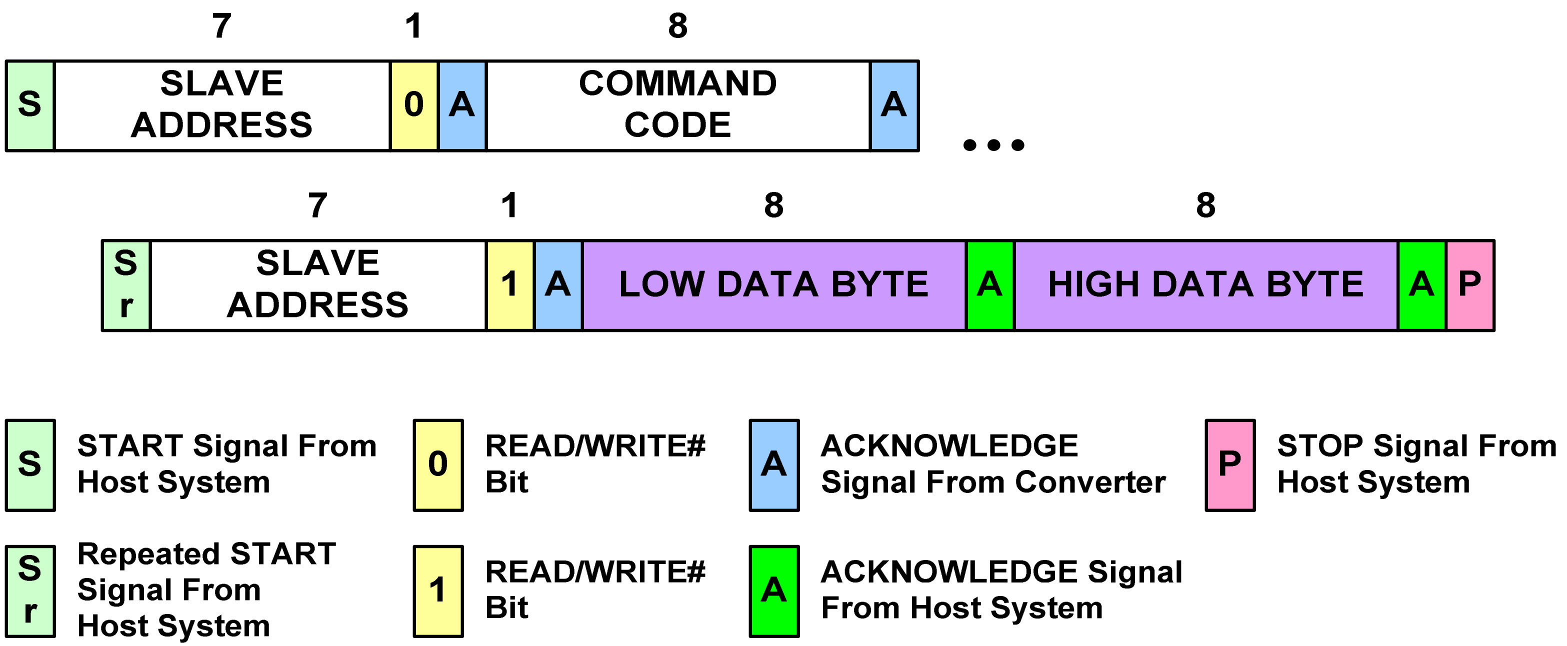
Write Word Packet



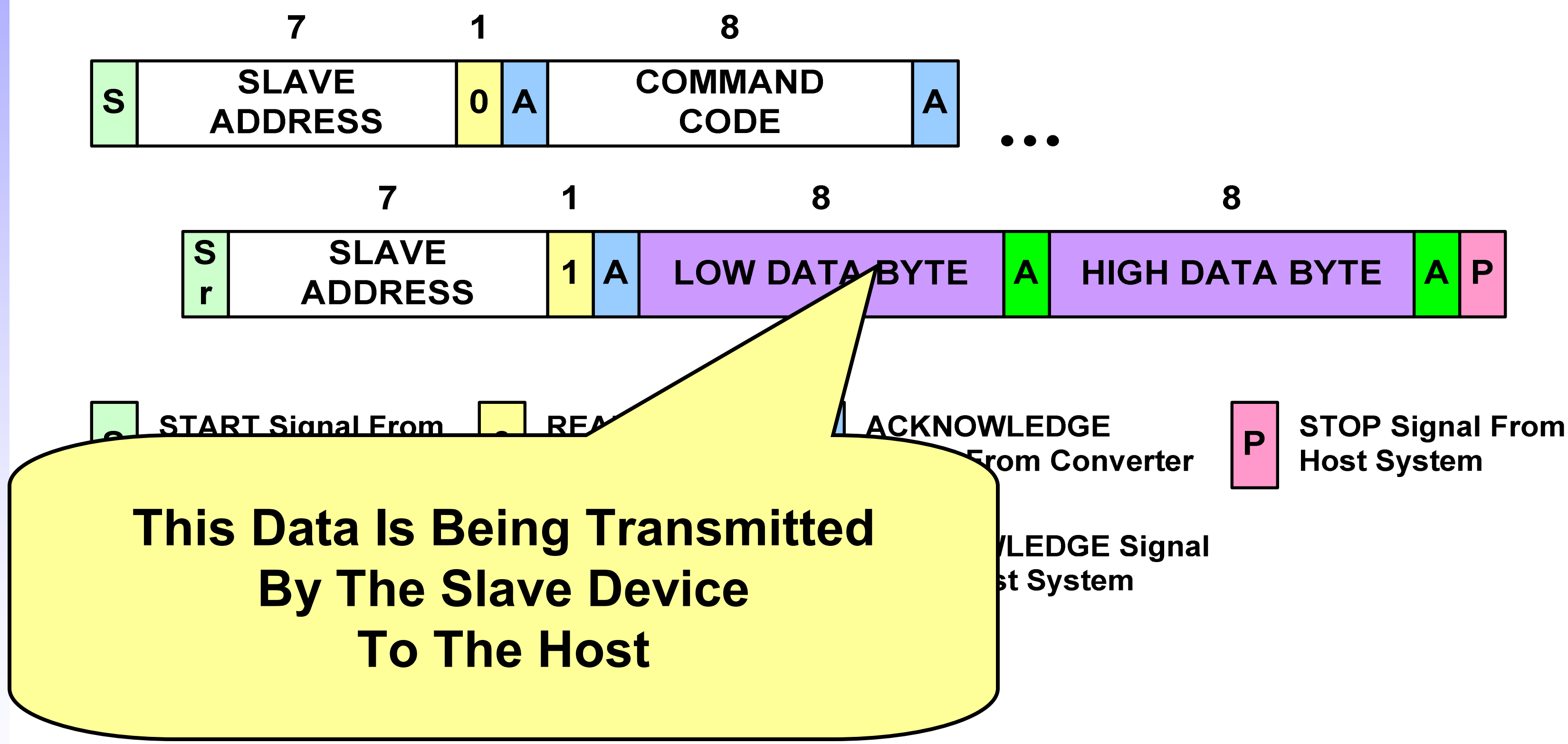
Packet Error Checking (PEC)
Is Optional In The Specification

But It Is Expected To Be
Very Popular With System OEMs!

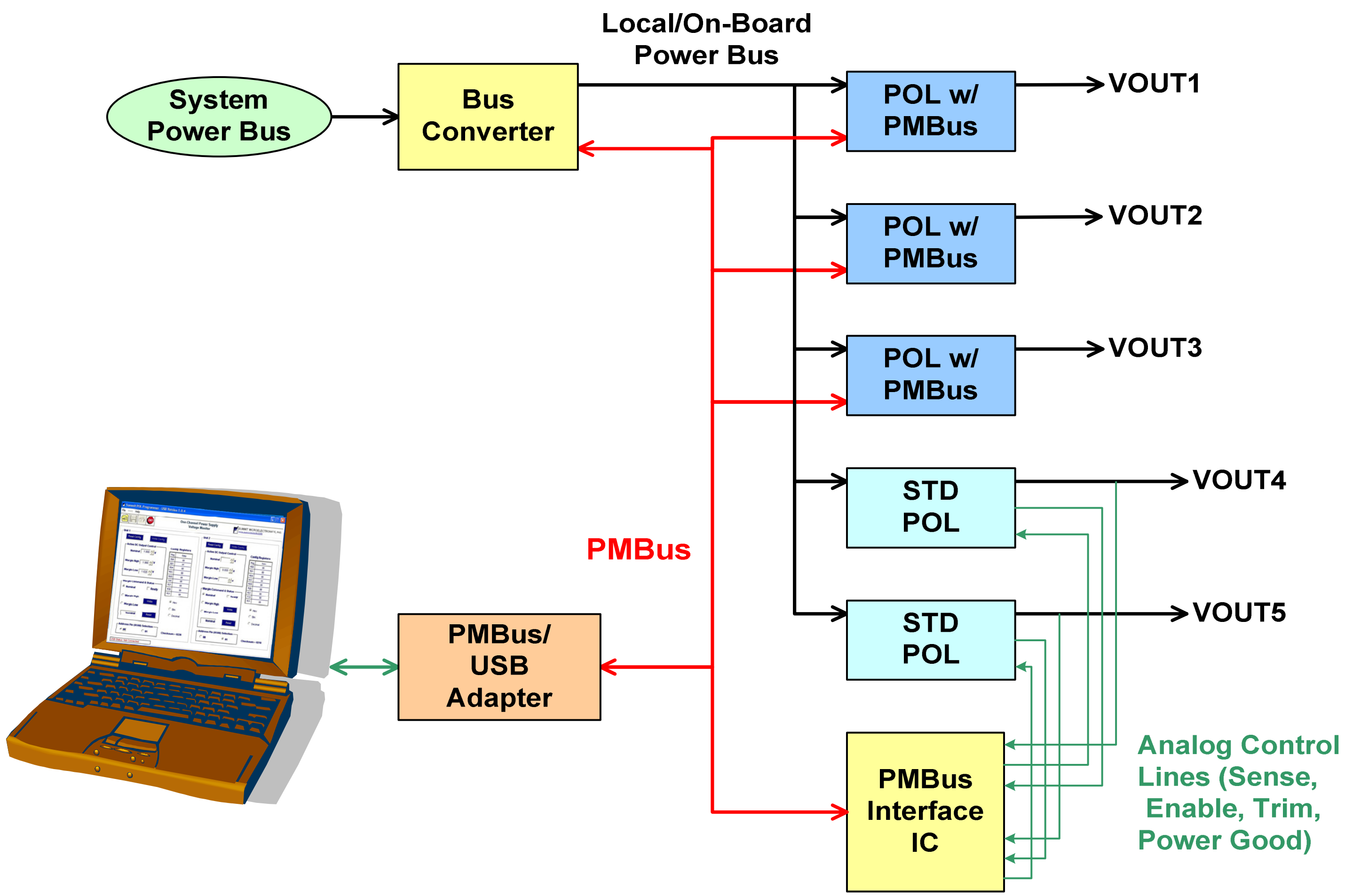
Read Word Packet



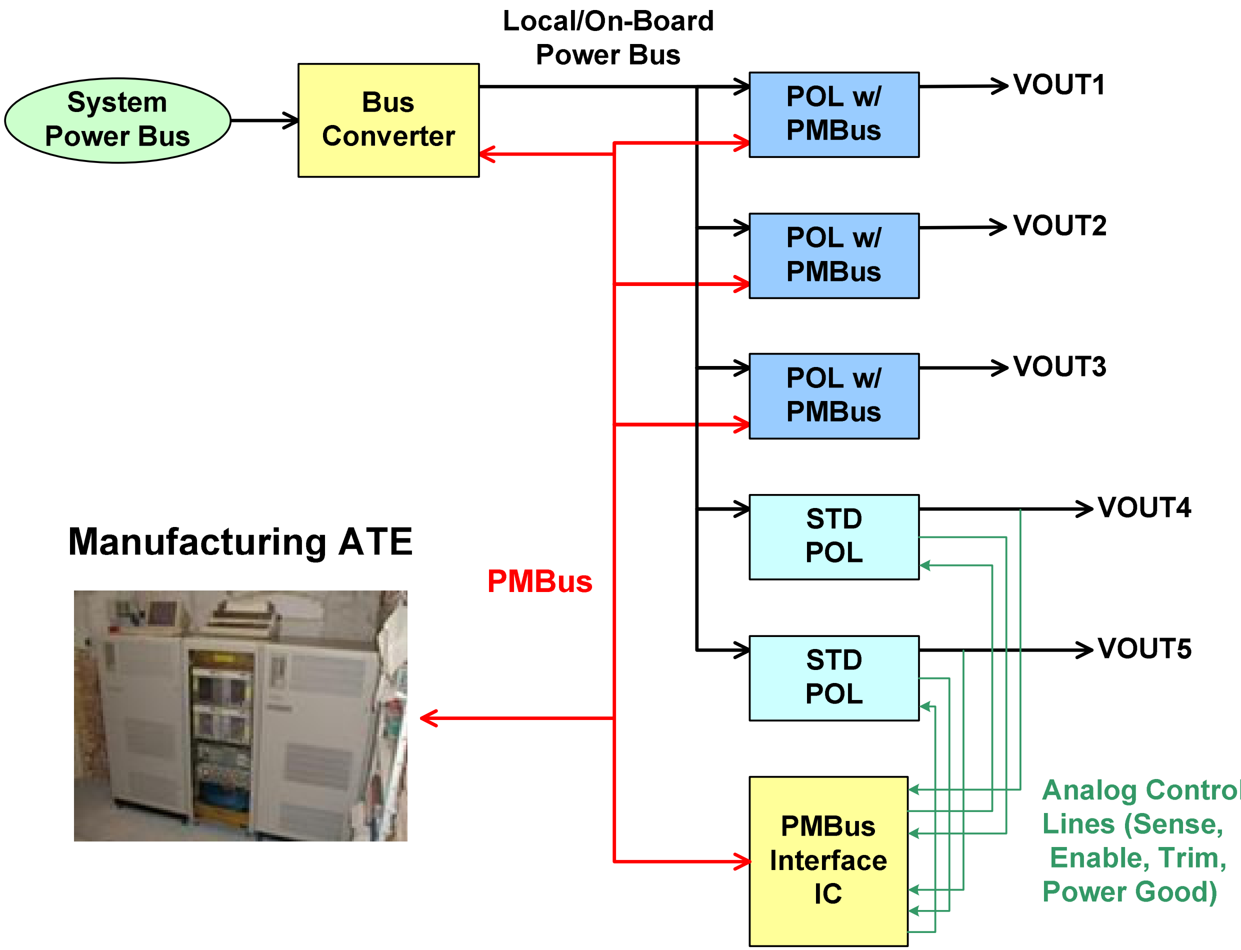
Read Word Packet



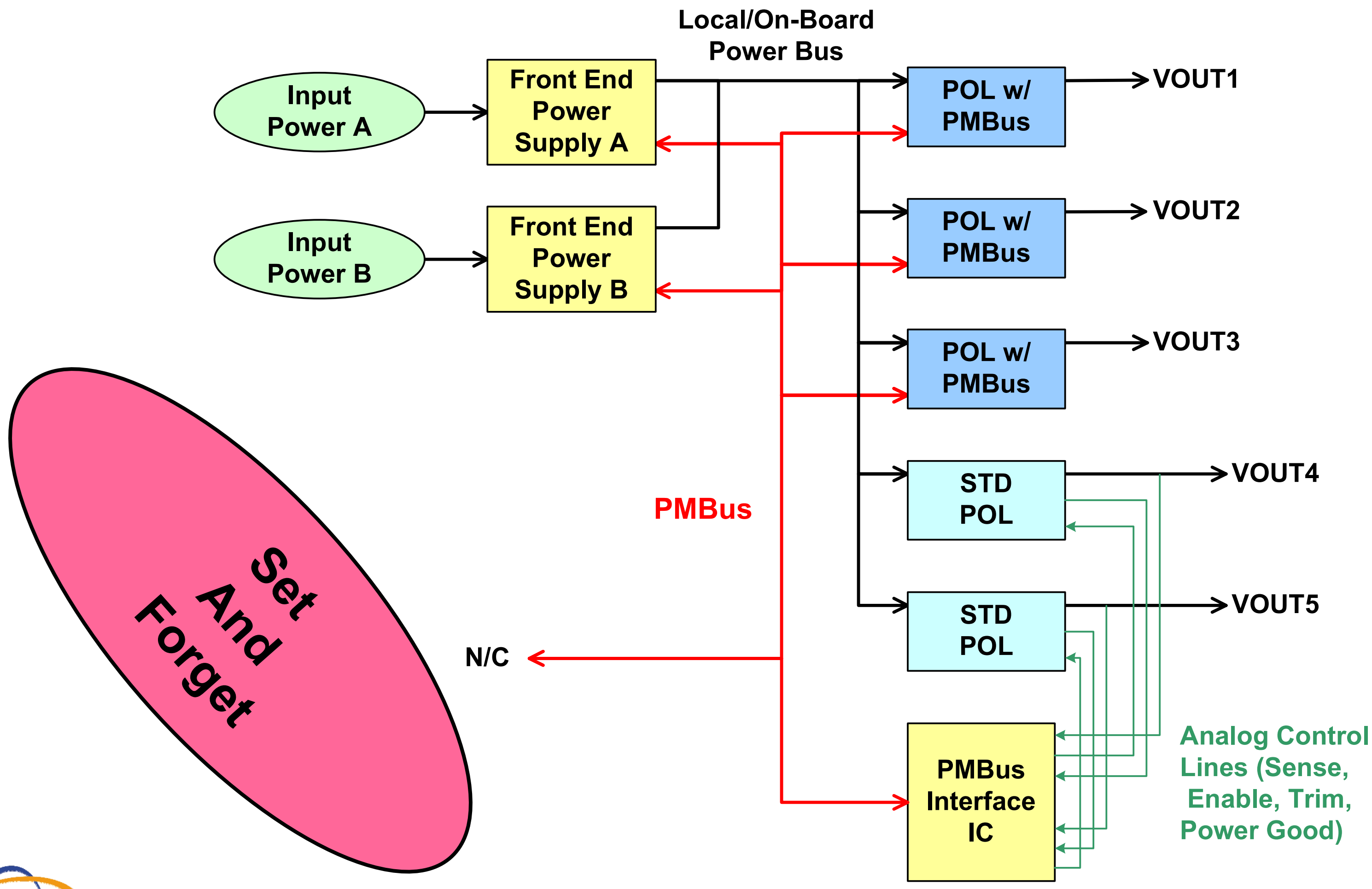
Using PMBus In The Lab



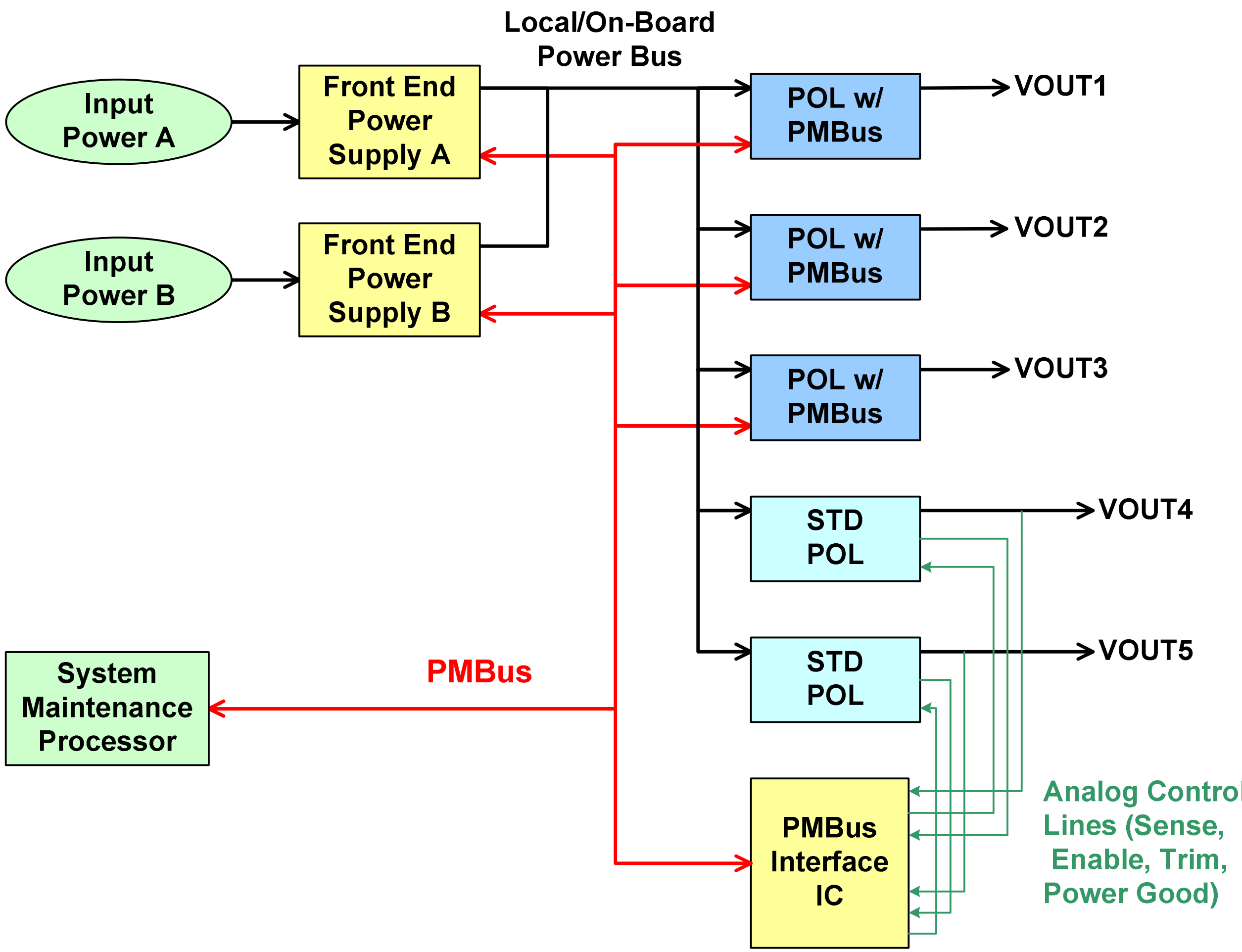
Using PMBus In The Factory



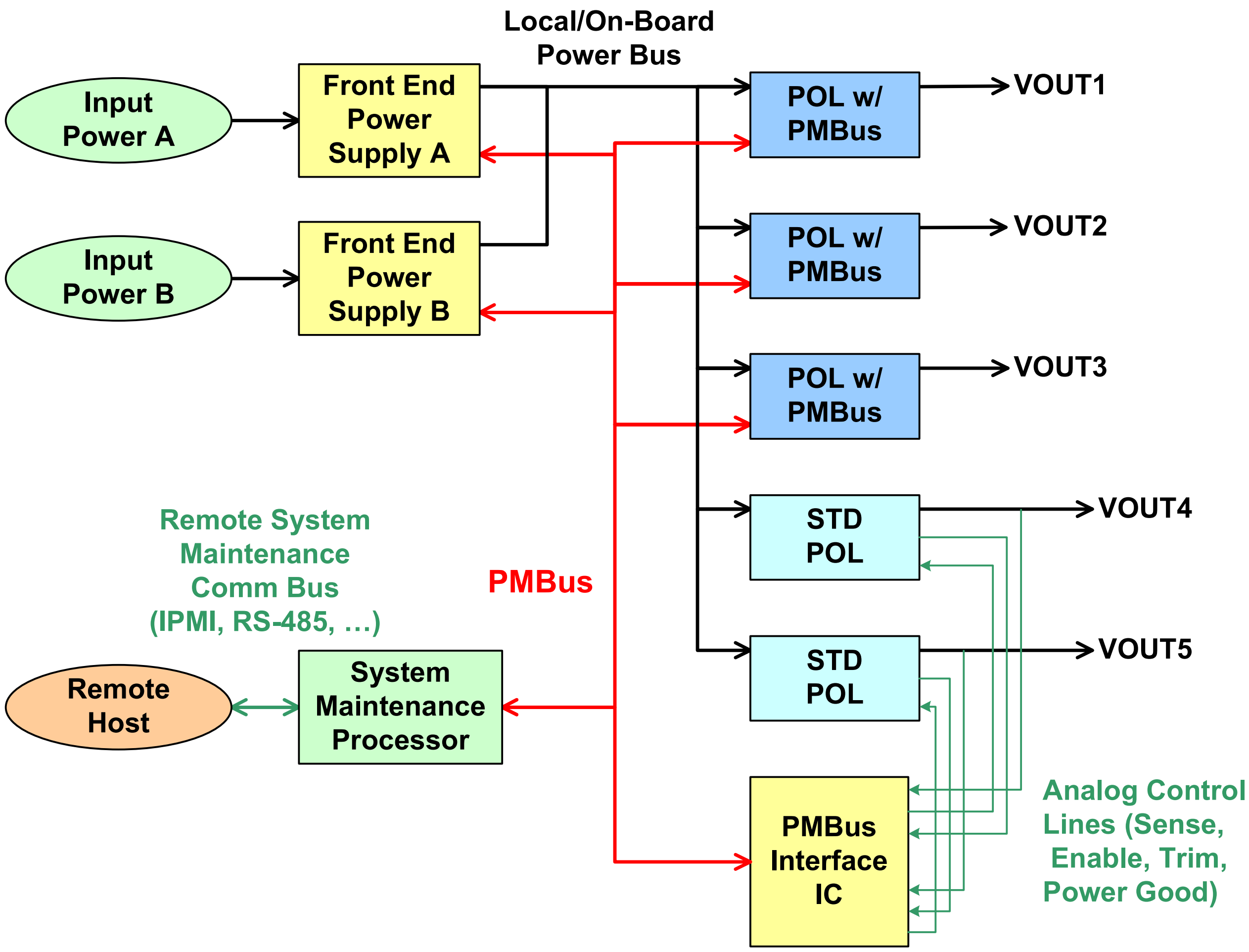
Using PMBus In A System



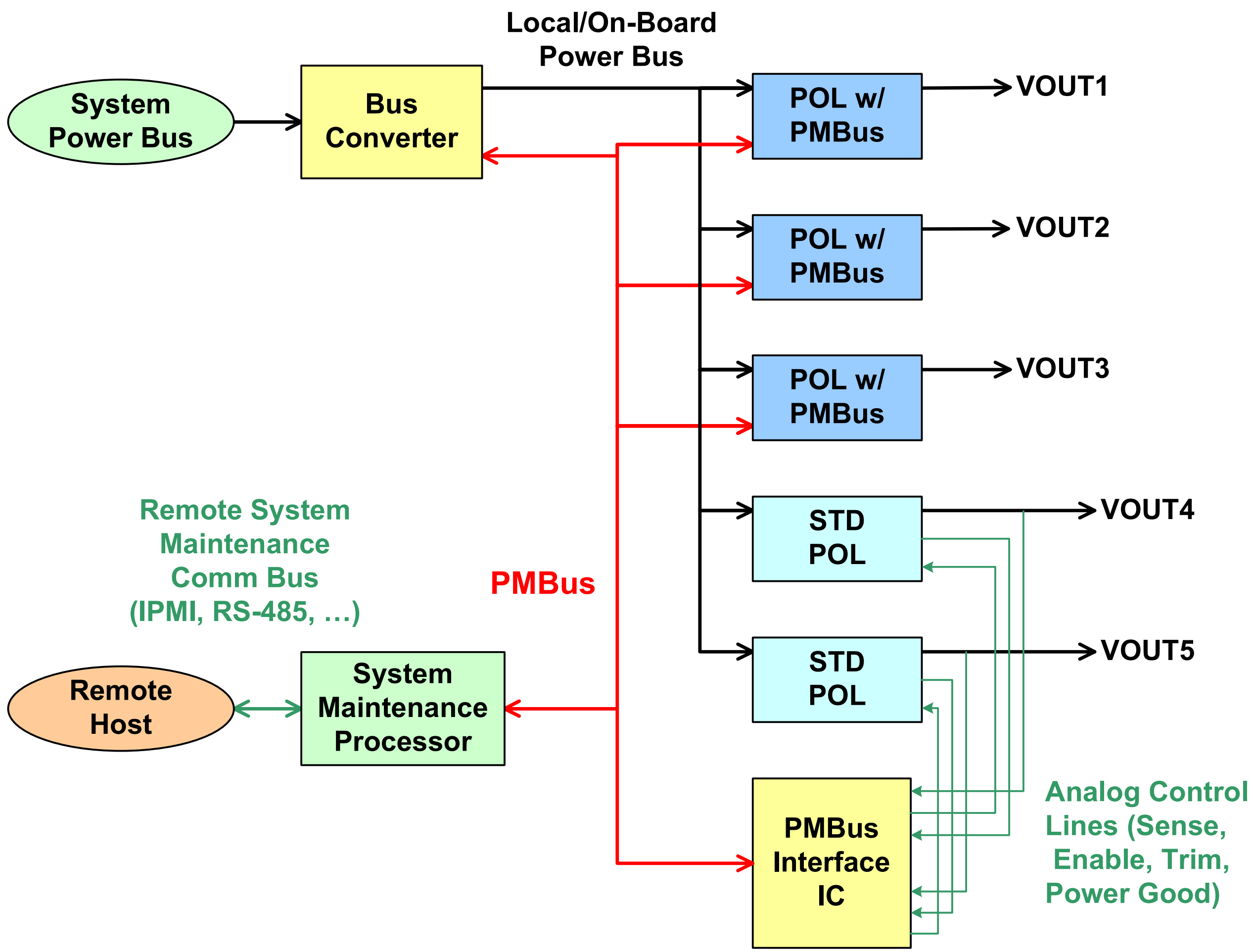
Using PMBus In A System



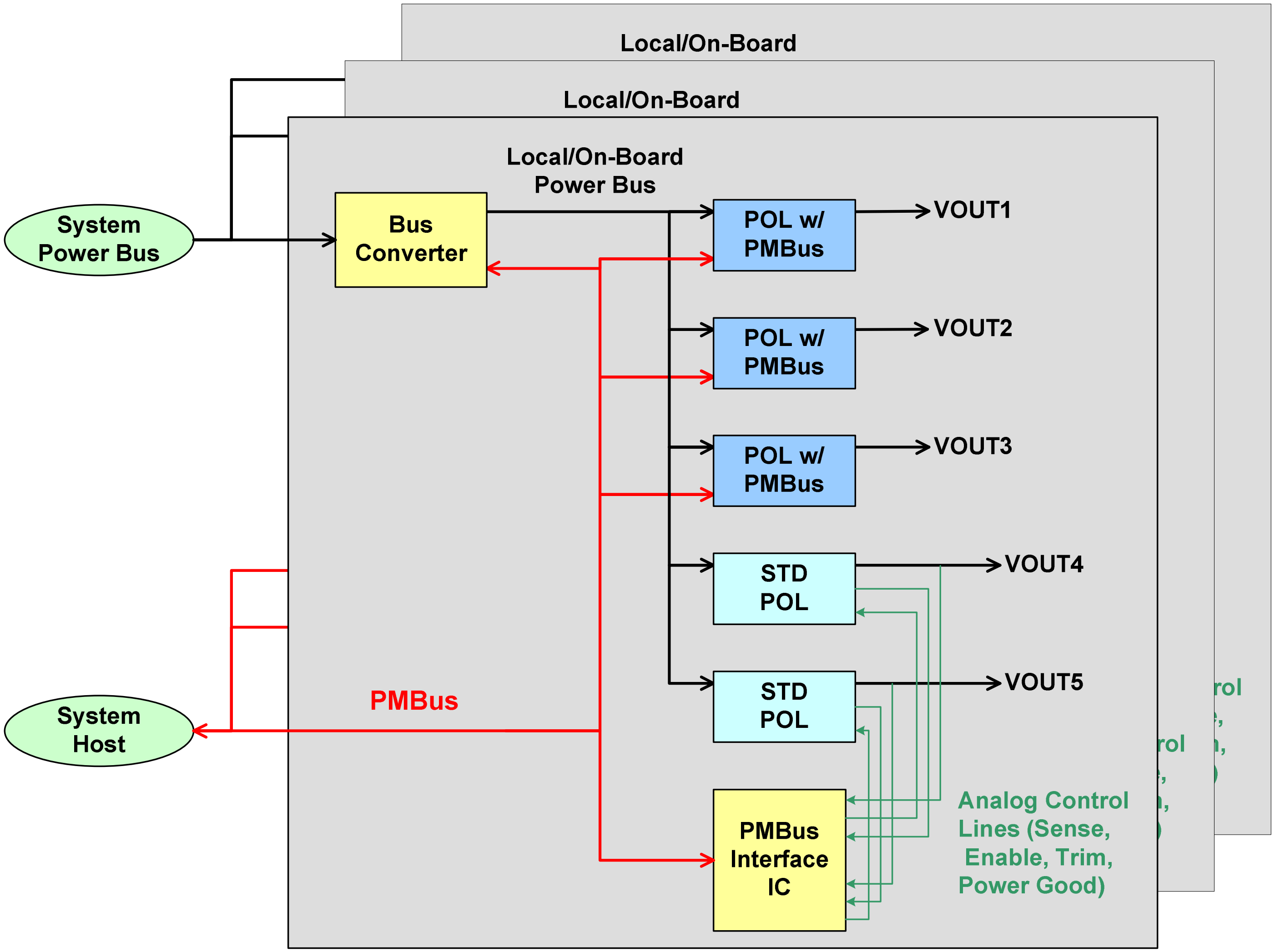
PMBus In A System



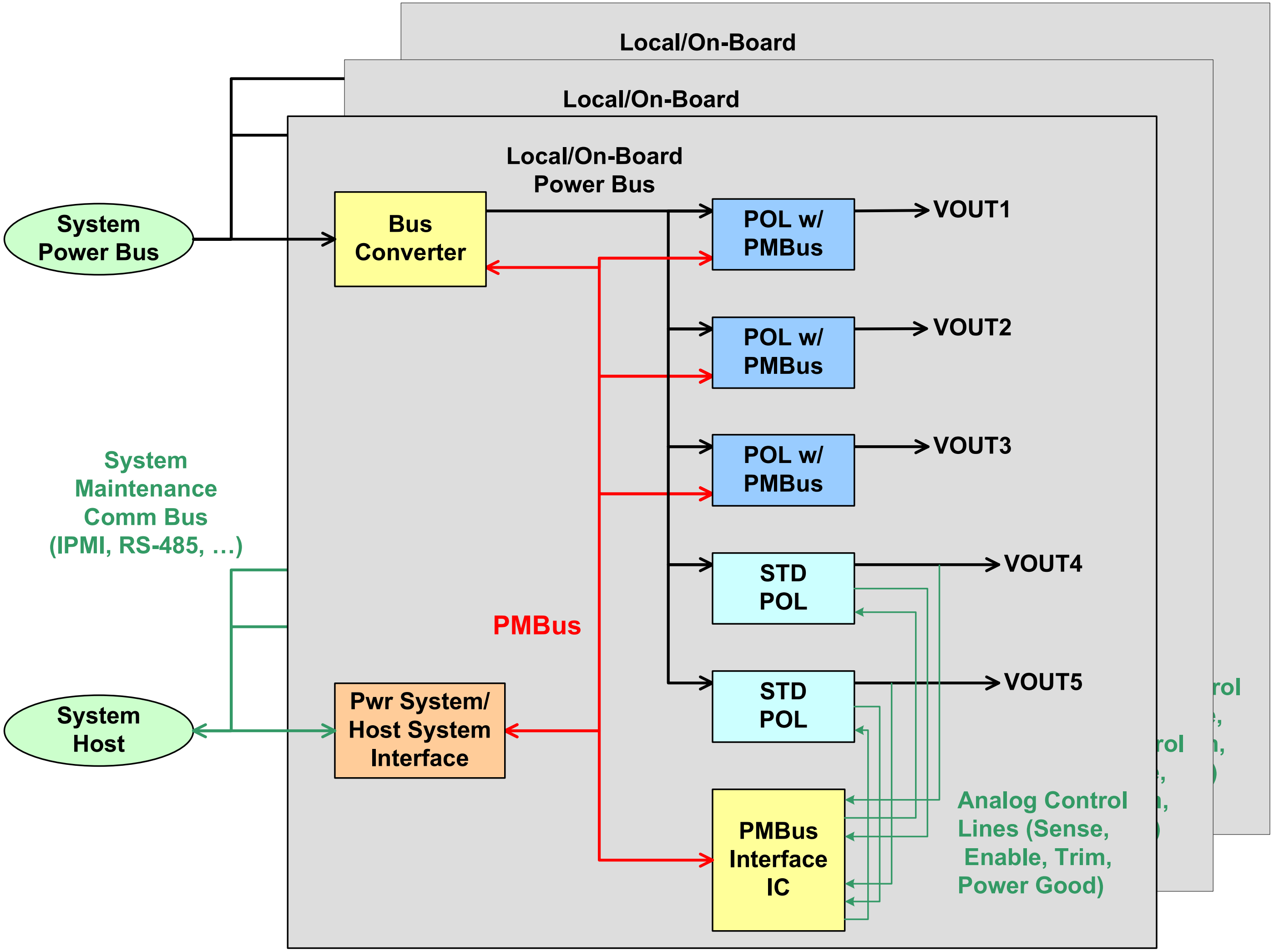
PMBus In A System



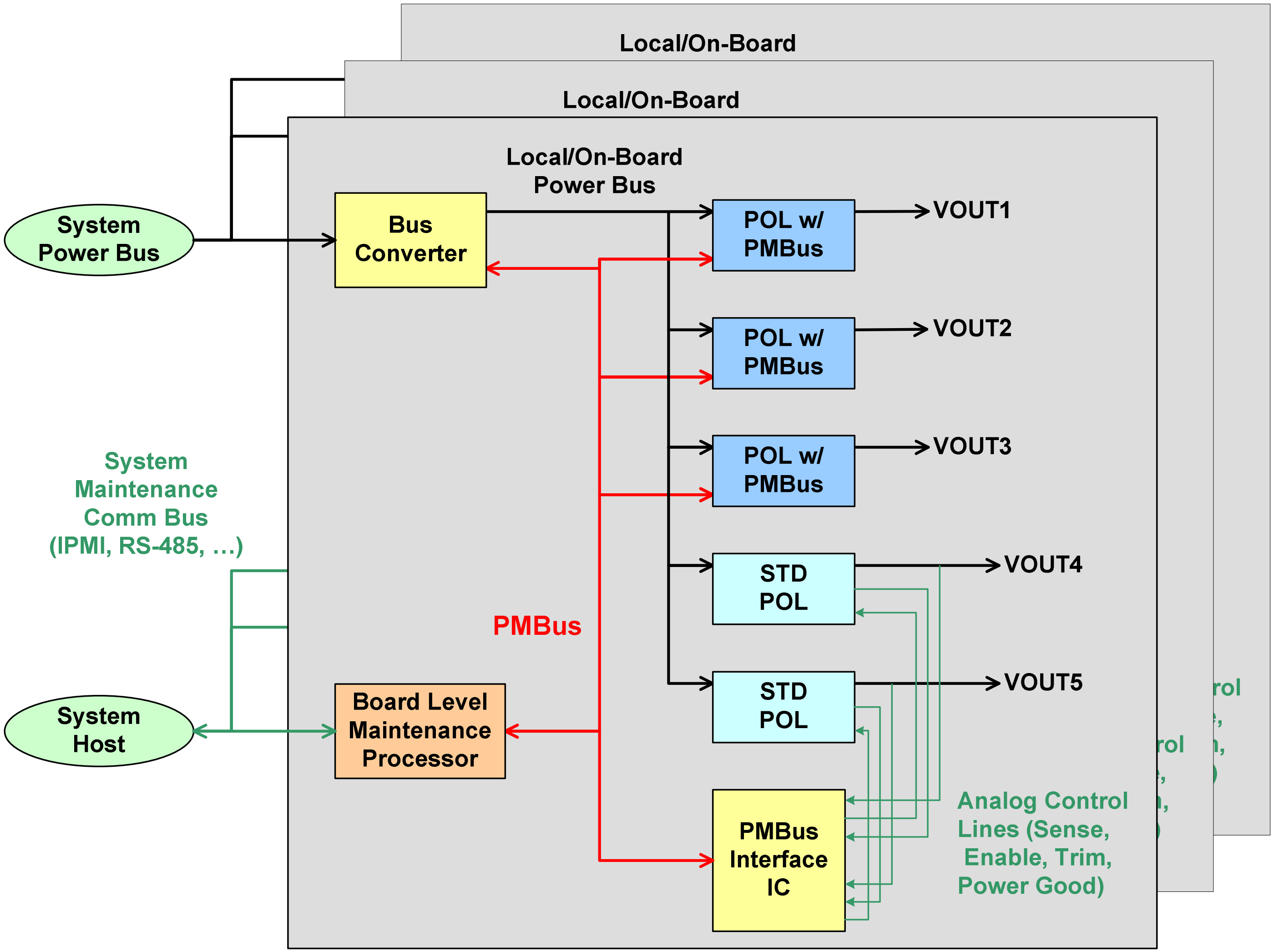
PMBus In A System



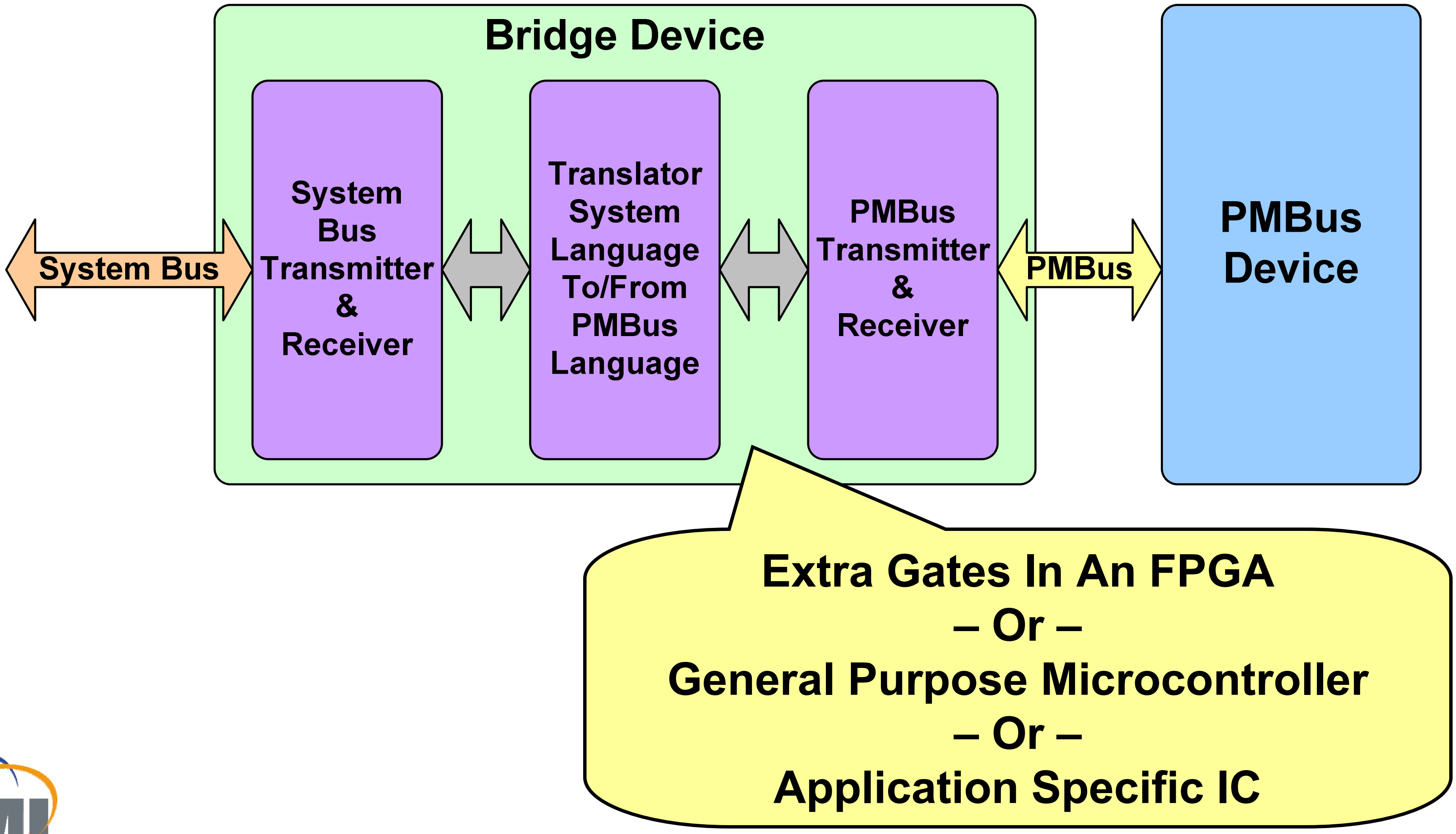
PMBus In A System



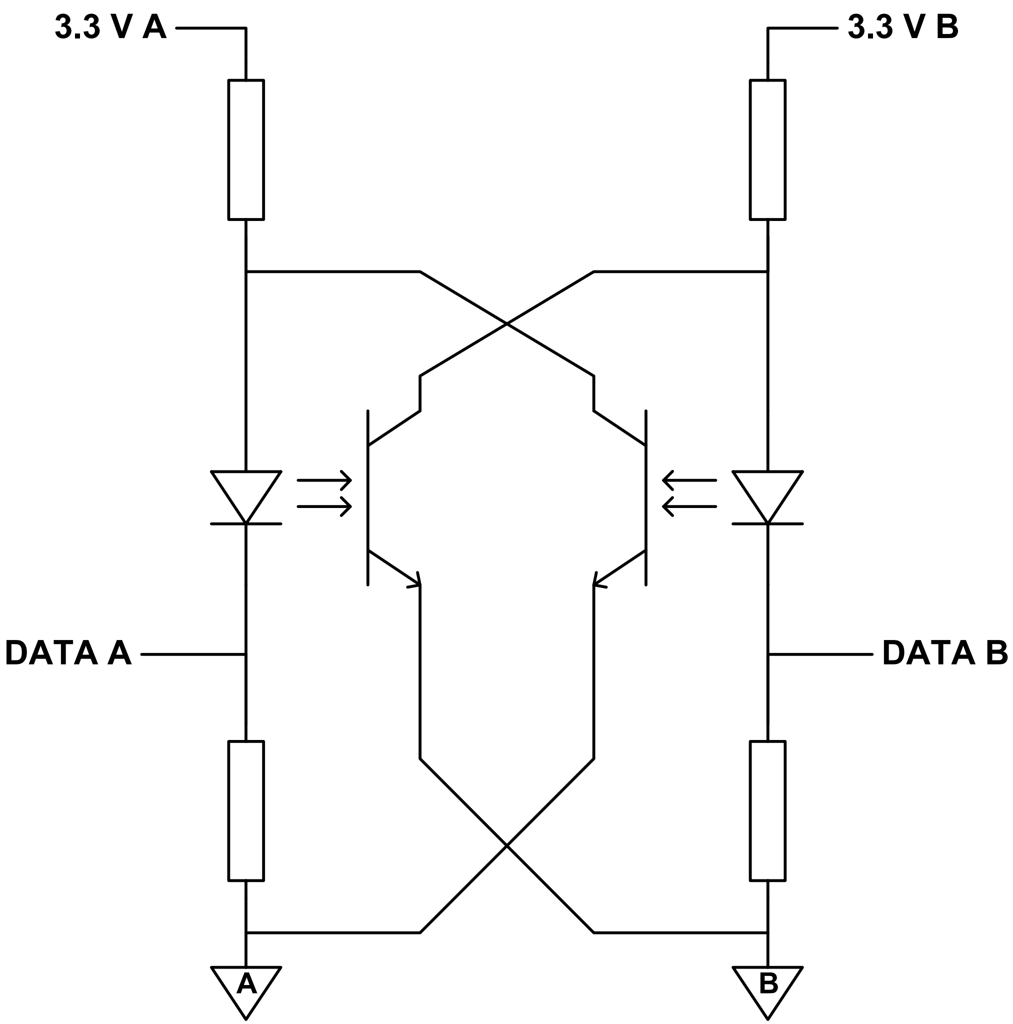
PMBus In A System



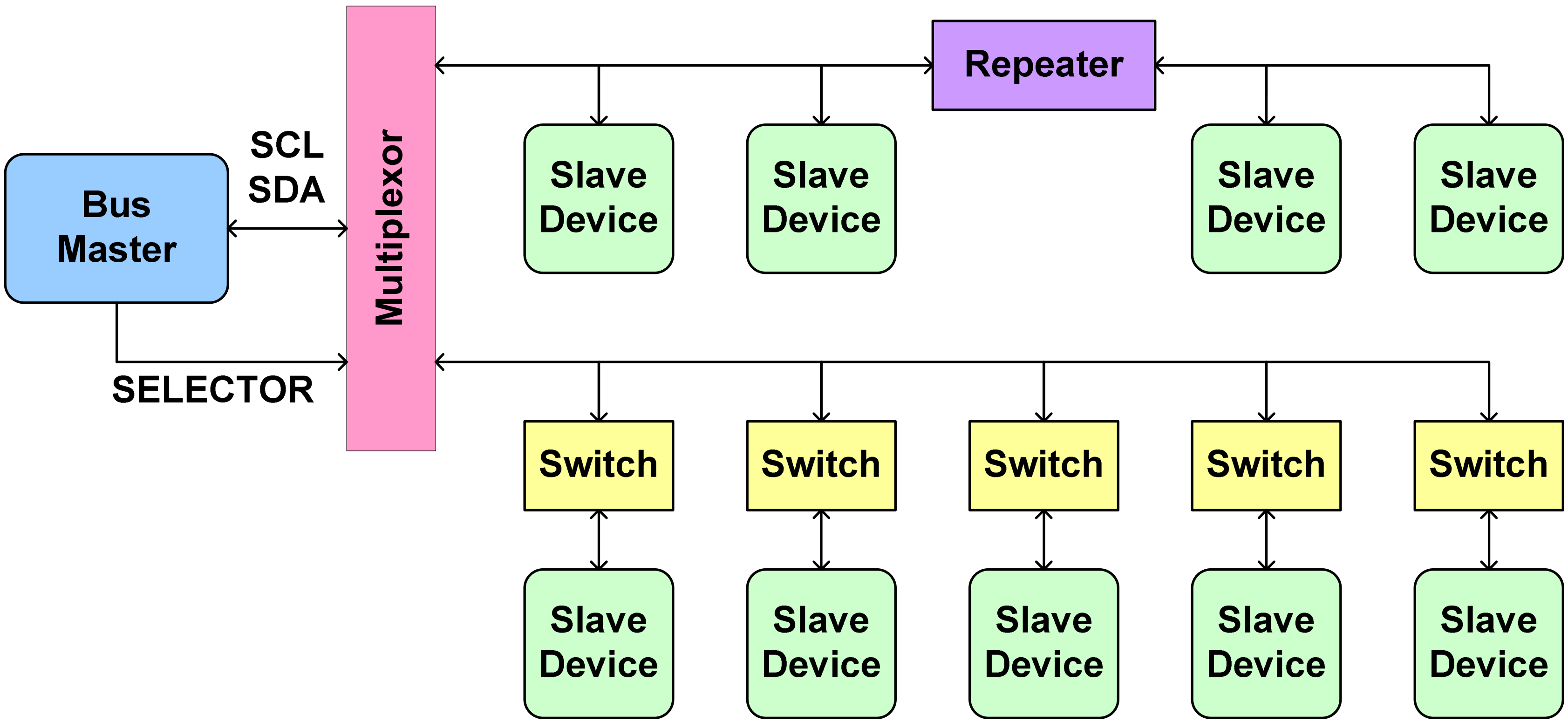
PMBus Bridge To Other Buses



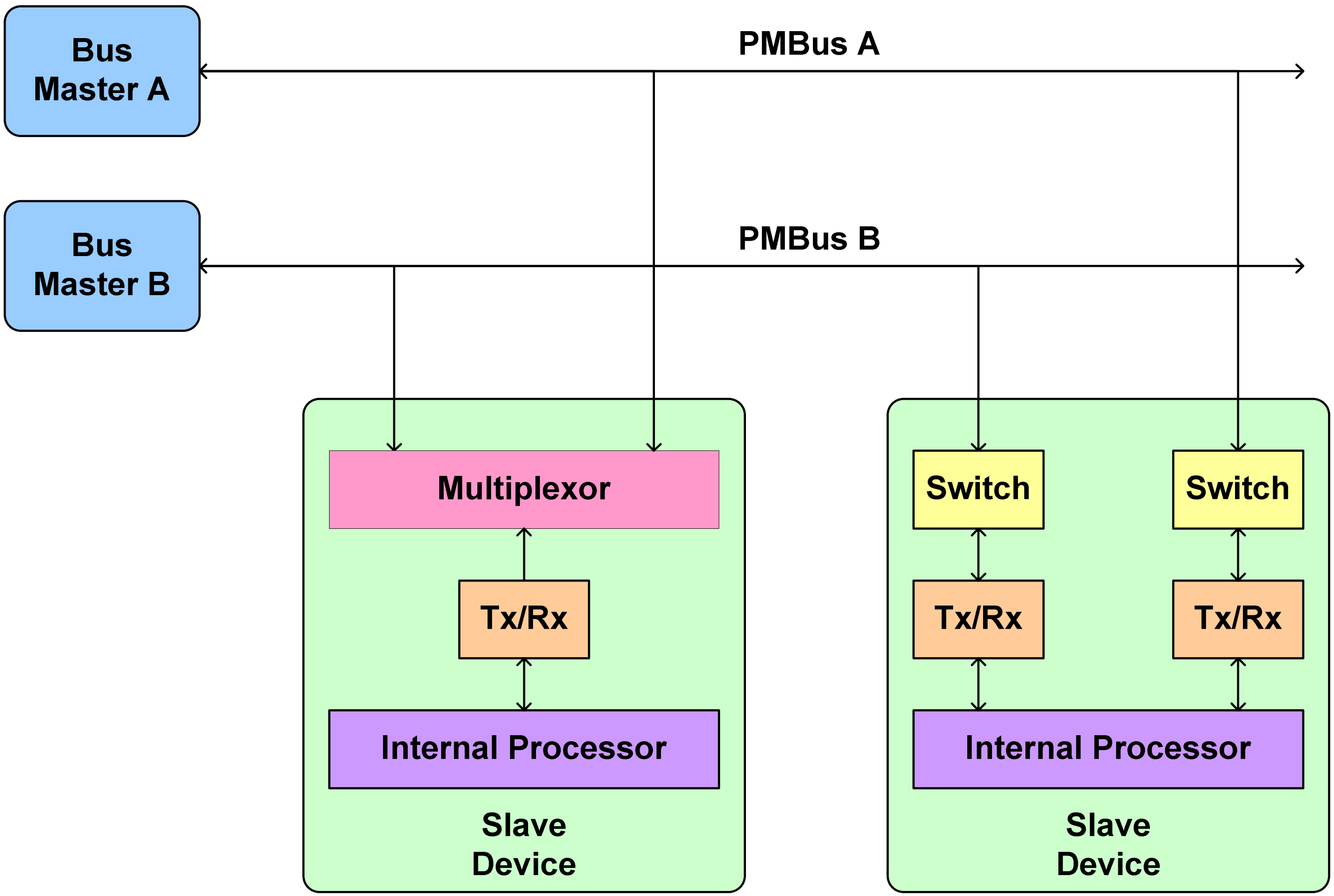
Simple Electrical Bi-Directional Isolation



Bus Extensions



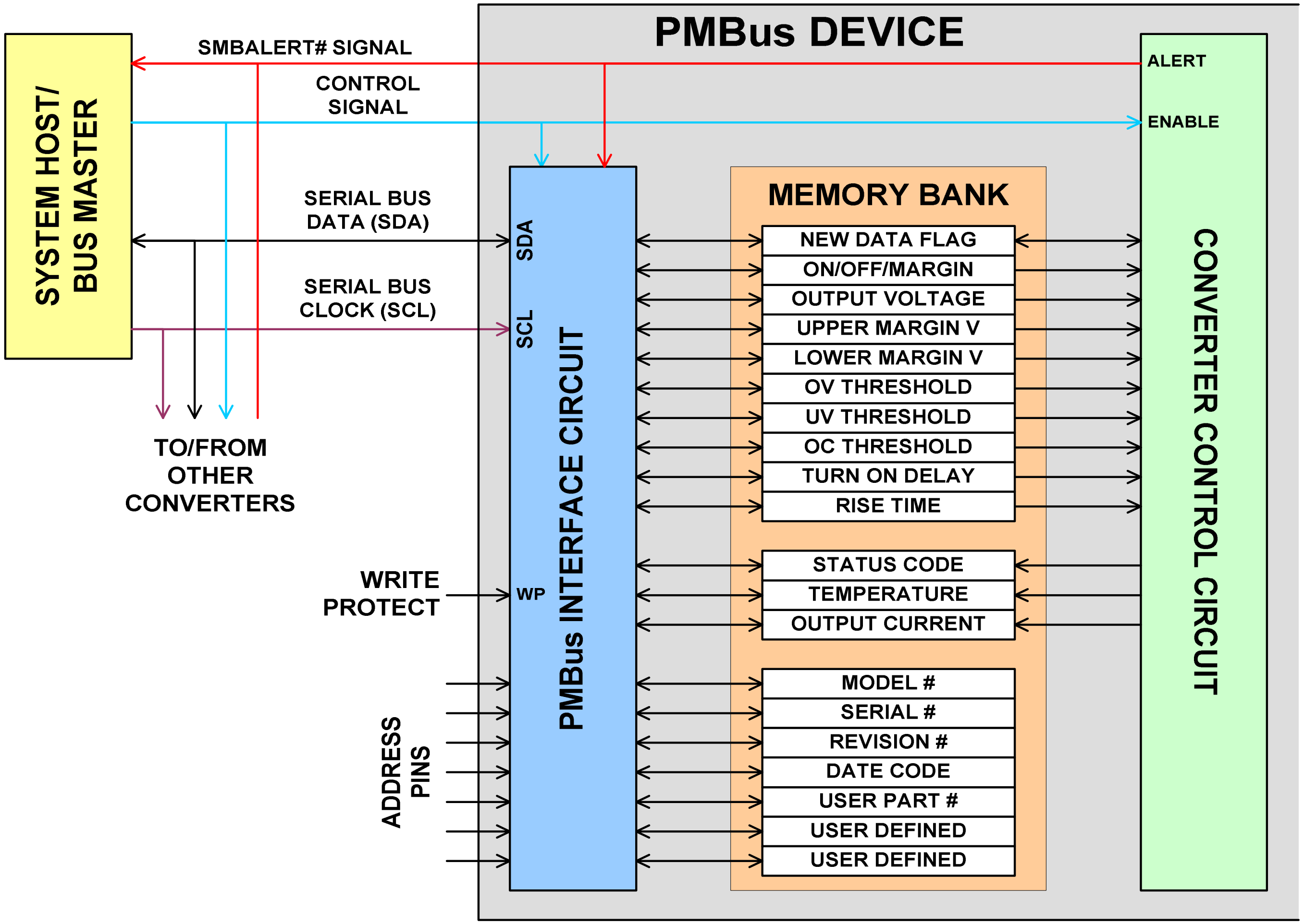
Redundant Buses



What's Needed To Make A PMBus Device?

- Physical/Data Link Layer To Receive & Send Data Over The Bus
  - Plus CONTROL, SMBALERT#, WP, Address Pin Interface
- Memory
  - Received Configuration
  - Device Status And Parametric Information
- The Rest Of The Device
  - Such As Power Control And Conversion Circuits That Use/Supply Stored Information
  - Note That PMBus Does Not Depend On The Type Of Controller: Analog, Digital, Hybrid

PMBus Device Concept



How to Make A PMBus Device

Integrated Solution

- ASSP
- ASIC

Piece Part Solution

- Bus Interface
- Control & Monitor
- ASIC
- FPGA
- GP Microcontroller

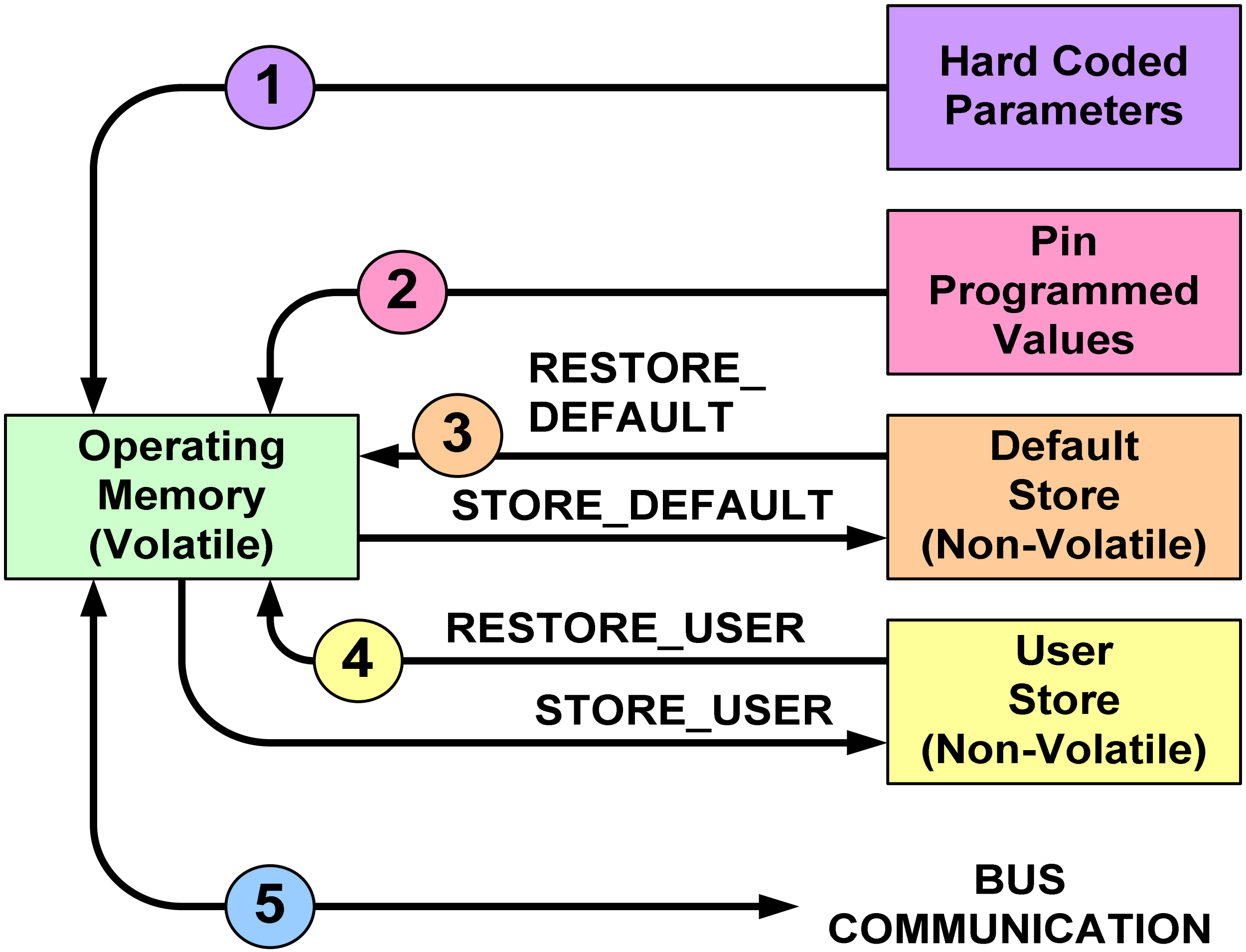
PMBus
Power Management Bus Implementers Forum

Making A PMBus Device

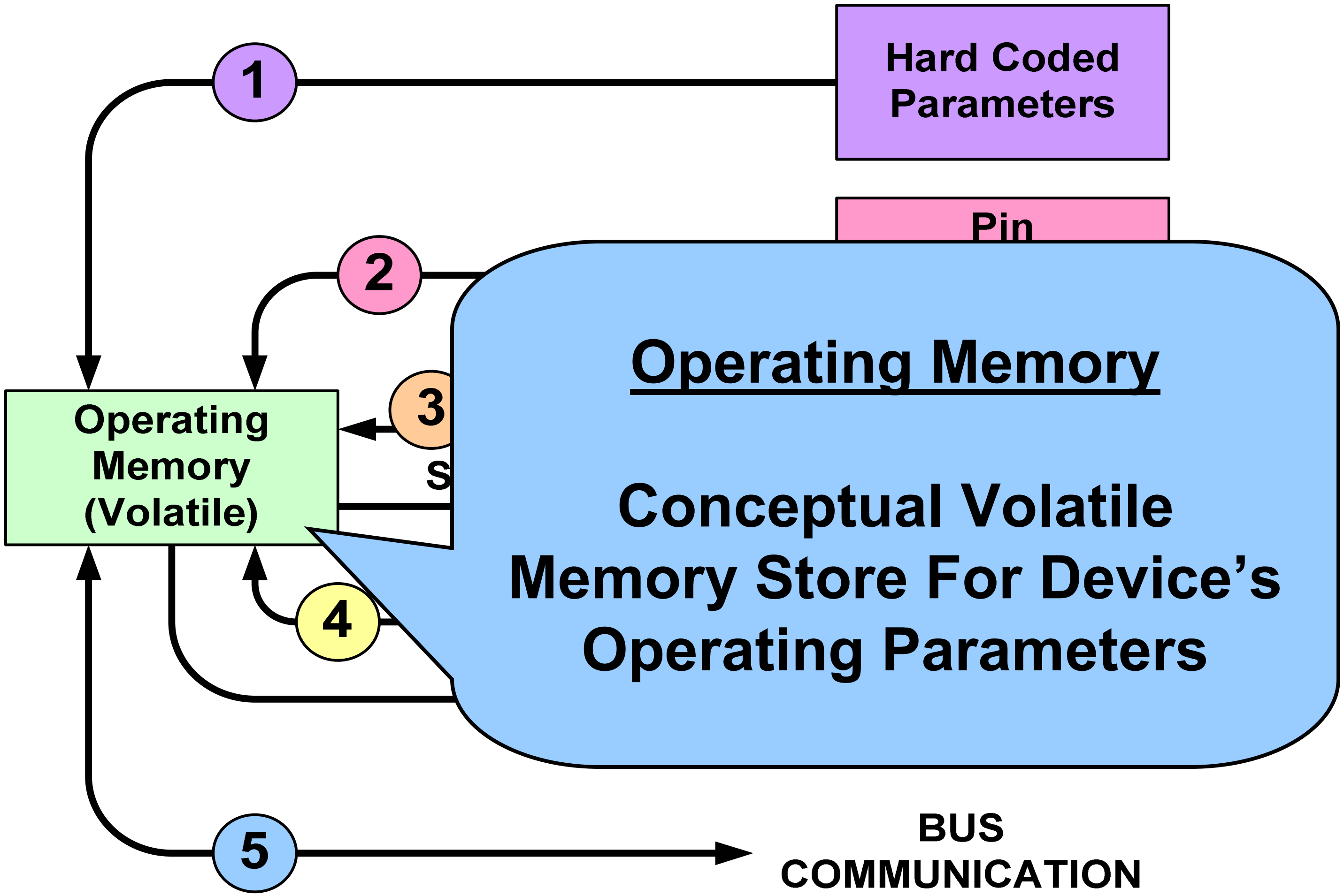
	Pro	Con	Risk
ASSP	Minimal Investment	Hard To Differentiate	Low Technical Medium Economic
ASIC	Have It Your Way	$$$$$ Design Skills	Medium Technical High Economic
FPGA	Have It Your Way	$$$ Design Skills	Medium Technical Medium Economic
General Purpose Microcontroller	Flexibility & Added Functionality	$ Programming Skills	Medium Technical Medium Economic

SMI
System Management Interface Forum

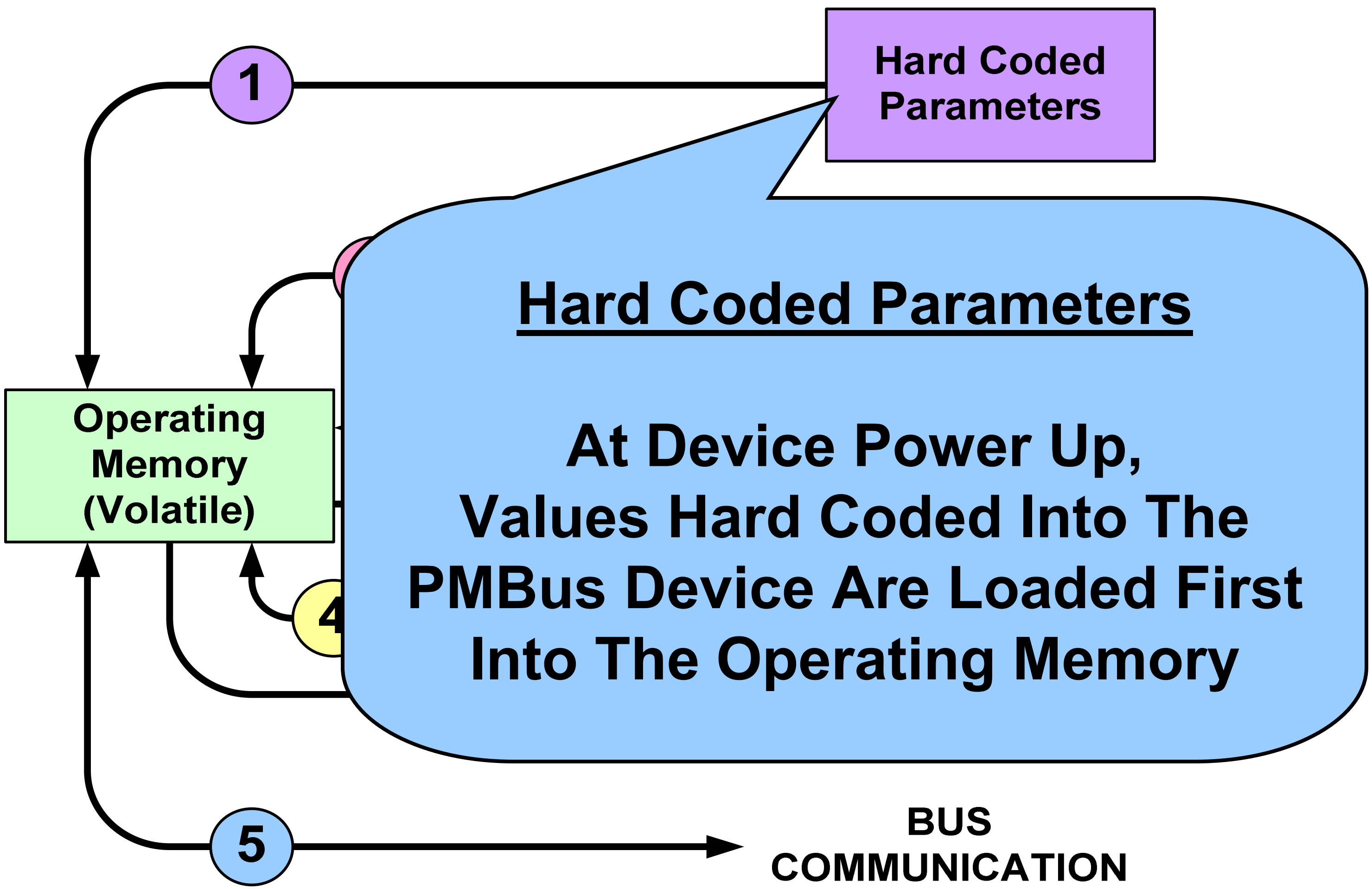
Conceptual View Of Memory And Startup



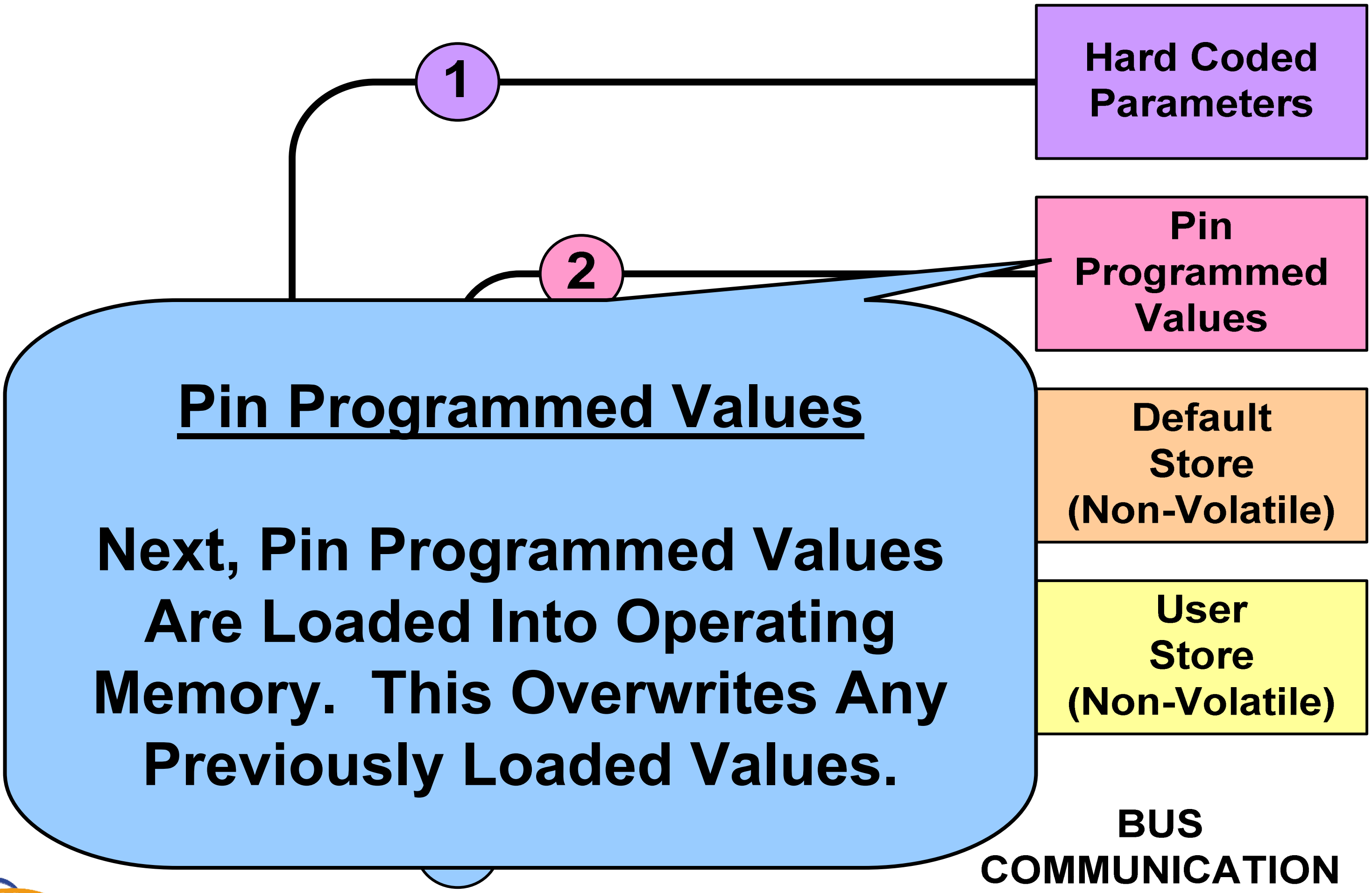
Conceptual View Of Memory And Startup



Conceptual View Of Memory And Startup



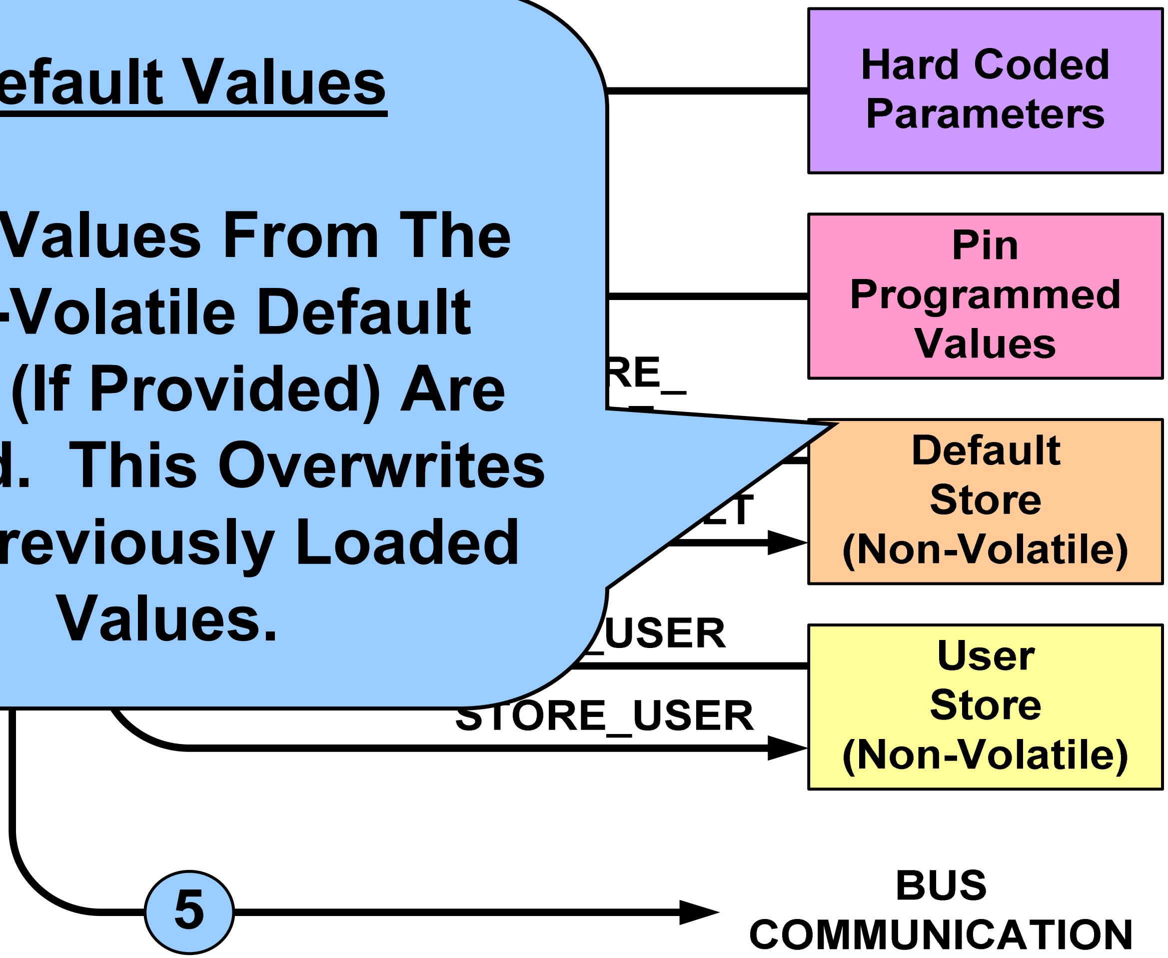
Conceptual View Of Memory And Startup



Conceptual View Of Memory And Startup

Default Values

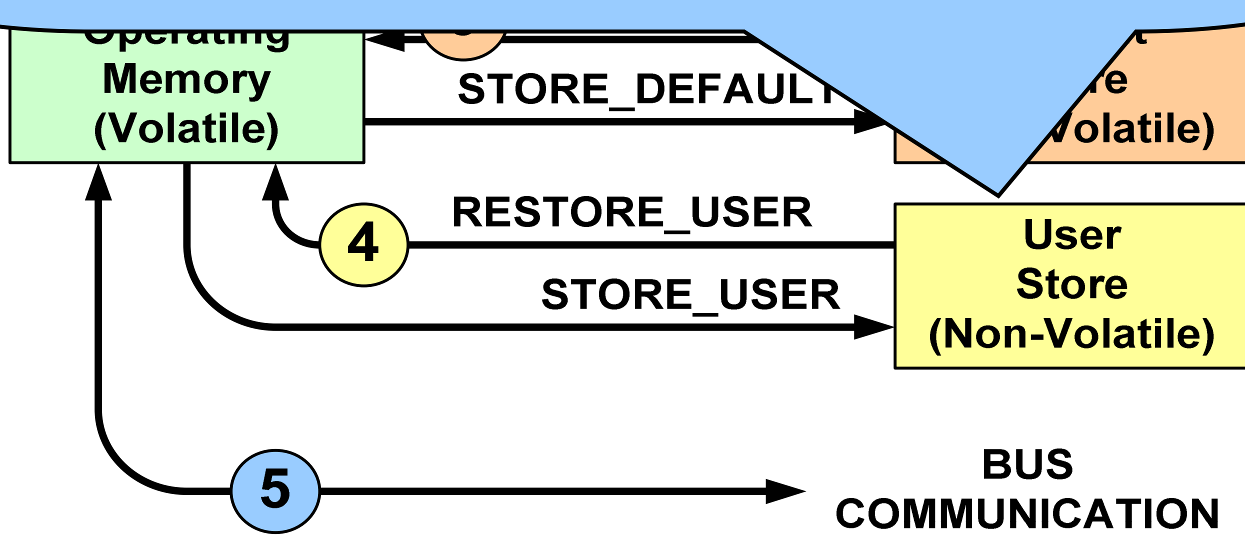
Next, Values From The Non-Volatile Default Store (If Provided) Are Loaded. This Overwrites Any Previously Loaded Values.



Conceptual View Of Memory And Startup

User Stored Values

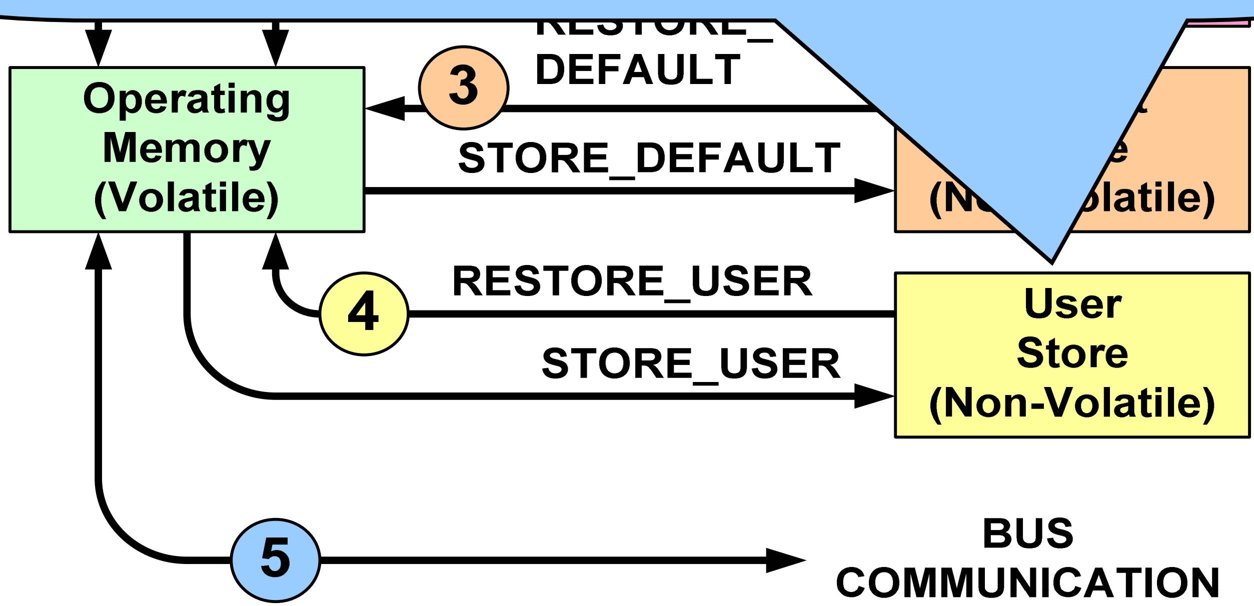
Next, Values From The Non-Volatile User Store (If Provided) Are Loaded. This Overwrites Any Previously Loaded Values.



Conceptual View Of Memory And Startup

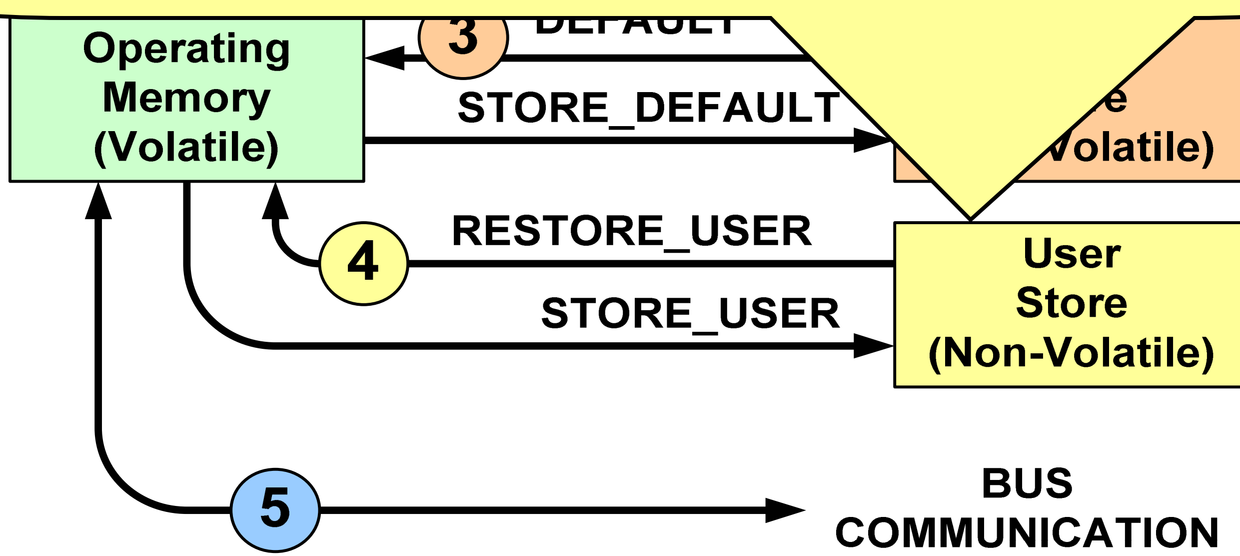
Bus Communication

Next, Values Sent Via The SMBus Are Loaded. This Overwrites Any Previously Loaded Values.

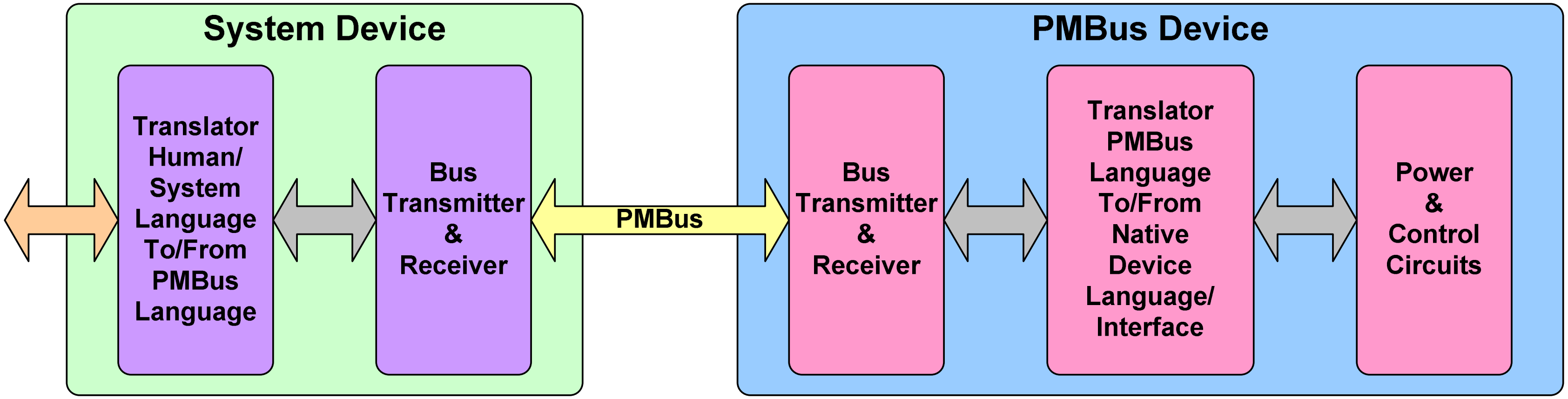


Conceptual View Of Memory And Startup

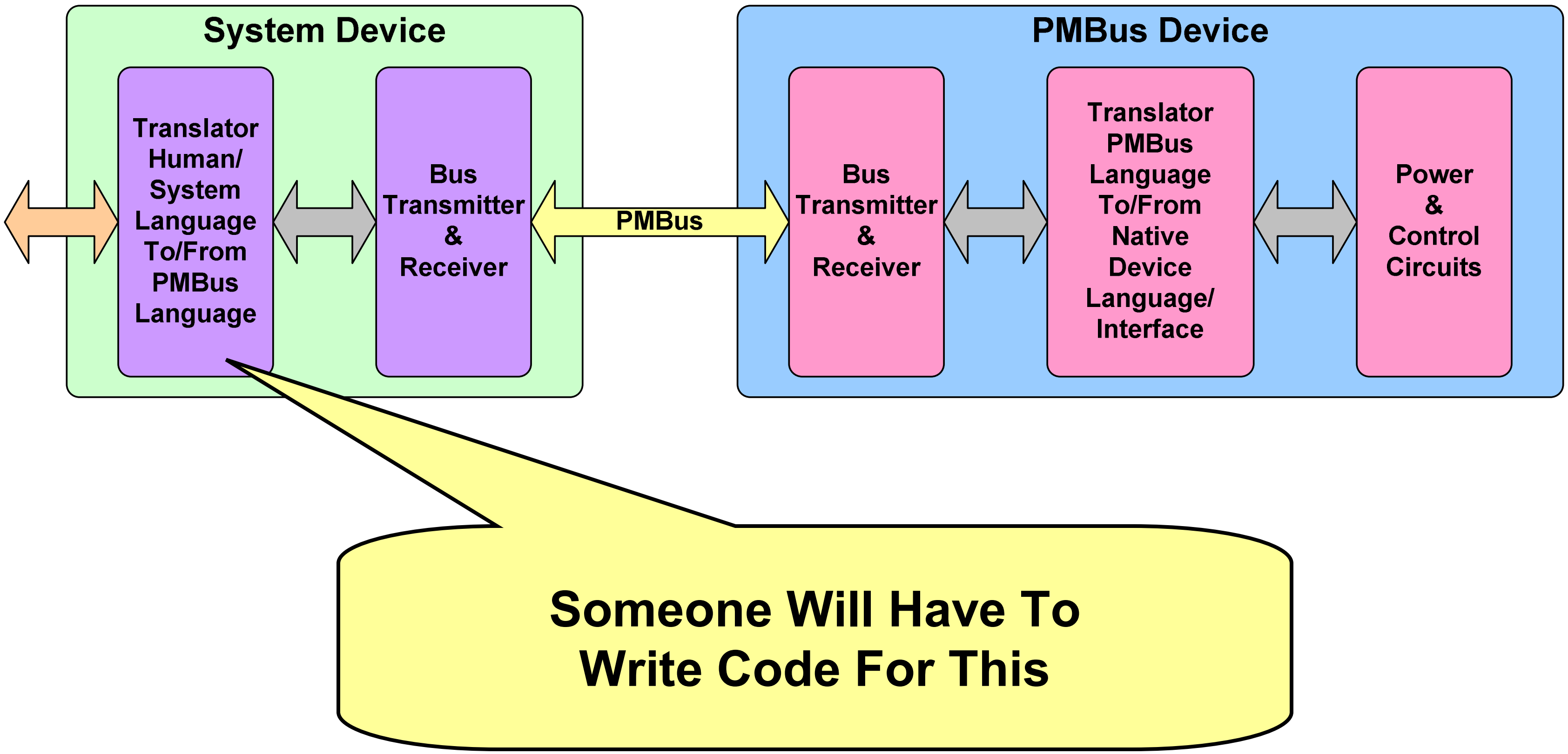
Used To Store A Snapshot Of The Device's Operating State. When Power Removed And Restored, Device Can Resume Operation From Its Last Programmed State.



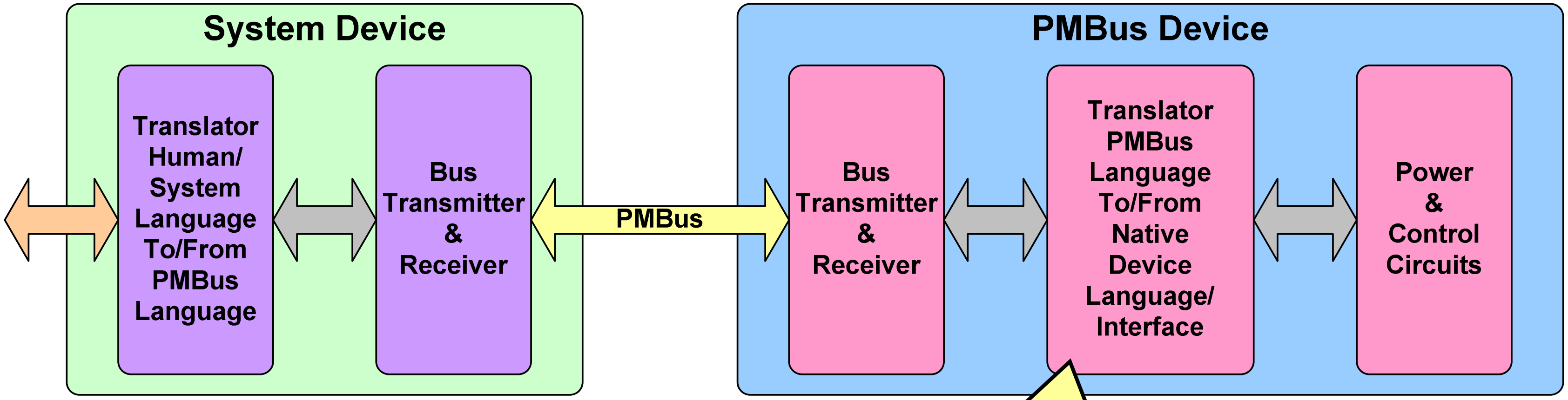
PMBus–Host Interface



PMBus–Host Interface

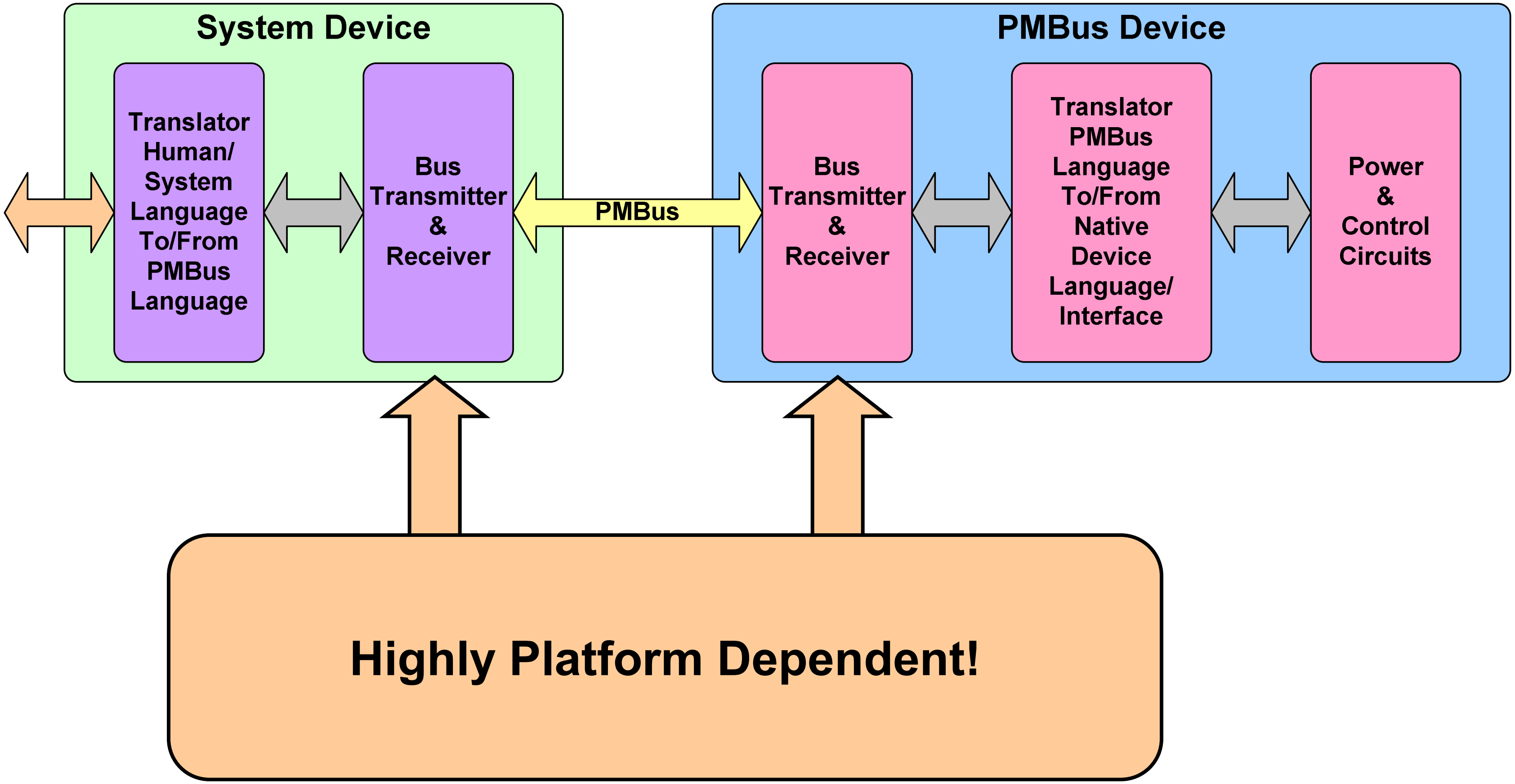


PMBus–Host Interface



Calculation And Conversion For Input To D/A Converters And Output From A/D Converters

PMBus–Host Interface



PMBus/SMBus Interface

- "Bit Banging" With A General Purpose I/O Port On A Microcontroller
  - Can Be Done & Can Be Done Well
  - Pay Attention To The Specification
  - Timing Is Important
- Integrated Into Silicon
  - Many Microcontrollers Have An I²C Port That Can Be Used To Drive SMBus
  - Look For PMBus To Be Built Into I²Cs For Power Conversion And System Monitoring

Command Language

- Commands Consist Of:
  - A Command Code
    - 256 Command Codes (00h To FFh)
  - Zero Or More Data Bytes
- Command Code
  - Not A Register Location!
  - Devices Must Map Command Code To Memory Location Themselves
- Data Byte(s)
  - Defined In The Specification

Data Formats

- More Time Spent On This By Specification Working Group Than Any Other Topic!
- Challenges
  - Wide Range Of Values (Millivolts To Kiloamperes)
  - Wide Range Of Resolution
    - Millivolts For Microprocessors
    - Volts And Amperes For AC Power
  - Positive And Negative Values
  - Limited Computing Power In PMBus Devices

Data Format Choices

Resolution

General Purpose: 10 Bits

Output Voltage Related: 16 Bits

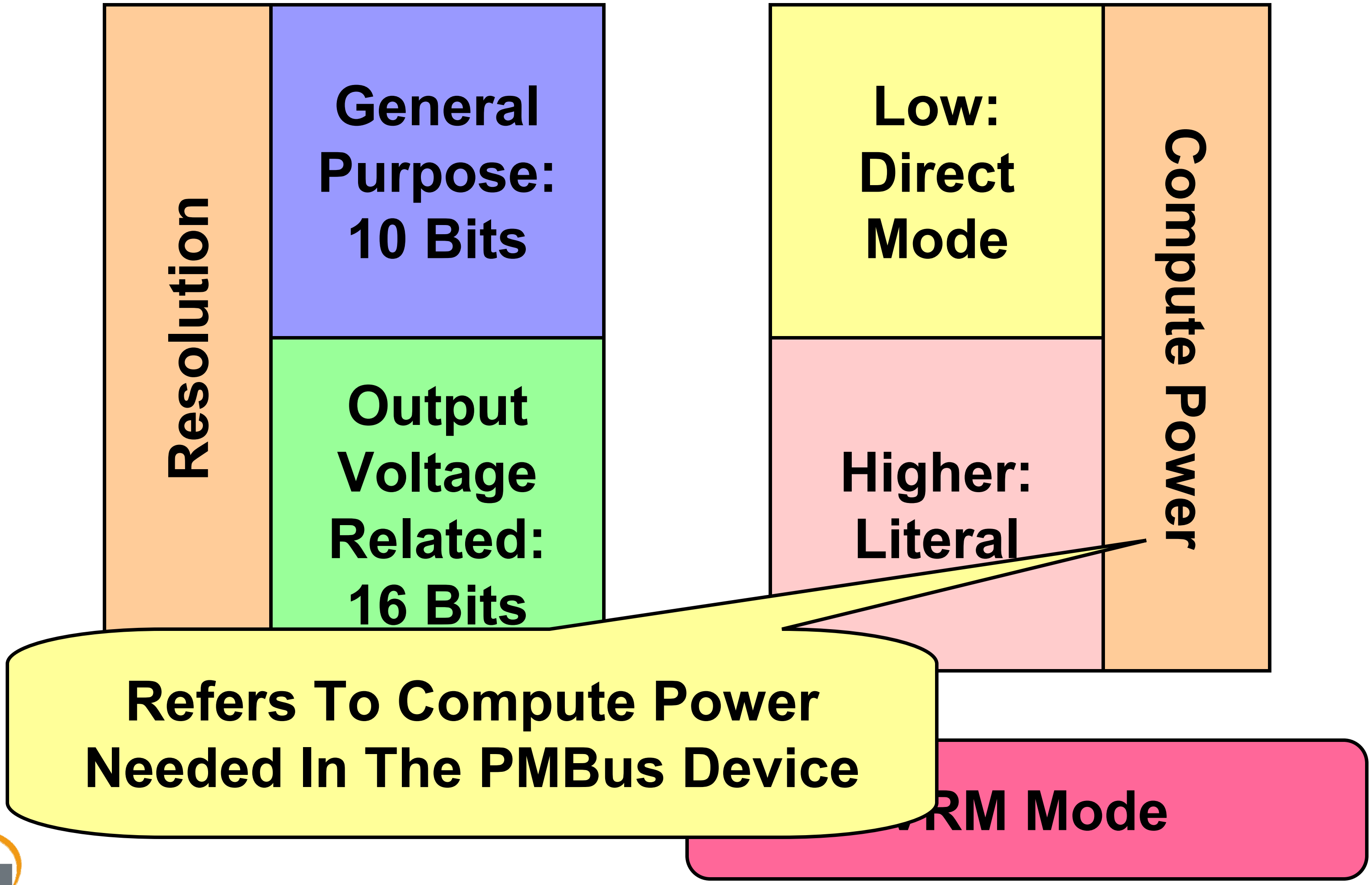
Low: Direct Mode

Higher: Literal

Compute Power

VRM Mode

Data Format Choices



Literal Format

- $X = Y \cdot 2^{N} \Leftrightarrow Y = X \cdot 2^{-N}$
  - X = "Real World" Value (Example: 3.3)
  - Y = Binary Value Sent Over The PMBus
  - N = Scale Factor
- Y (Binary Signed Integer)
  - General Purpose Case: 11 Bits
  - Output Voltage Related Data: 16 Bits
- N (Binary Signed Integer)
  - 5 Bits In Both General Purpose Data And Output Voltage Related Data

Literal Format

NOTICE!

This Is The Form That Will Appear In The PMBus Specification Revision 1.1

This Is "Backwards" From What Is In Specification 1.0 Section 7

– 5 Bits In Both General Purpose Data And Output Voltage Related Data

Literal Format

- Two Ways to Think Of This Format
- "Binary Floating Point"

$$X = Y \cdot 2^N$$

Y = Mantissa

N = Exponent

Literal Format

- Two Ways to Think Of This Format
- "Binary Floating Point"
  - Y = Mantissa
  - N = Exponent
- Number Of LSBs

$$X = Y \cdot 2^N$$

Y = Number Of LSBs

2^N Equals Size Of LSB

Literal Format

- Two Ways to Think Of This Format
- “Binary Floatin
  - Y = Mantissa
  - N = Exponent
- Number Of LSBs

Example: N = –10

$2^{-10} = 1/1024 = 9.766 \times 10^{-4}$ =>

977 microunits/bit

$$X = Y \cdot 2^N$$

Y = Number Of LSBs

2^N Equals Size Of LSB

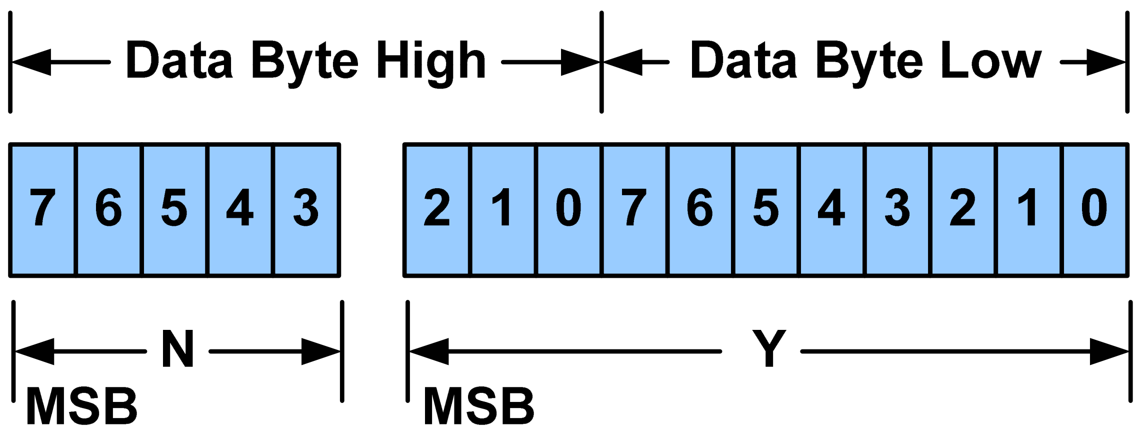
Power Management Bus Implementers Forum

System Management Interface Forum

General Purpose Literal Format

- 10 Bit Resolution
- Wide Range Of Values Possible
  - Maximum Positive: $1023 \times 2^{15} = 33.5217 \times 10^{6}$
  - Minimum Value: $\pm 1 \times 2^{-16} = \pm 1.526 \times 10^{-5}$
  - Maximum Negative: $-1024 \times 2^{15} = -33.5544 \times 10^{6}$

General Purpose (10 Bit) Literal Format



Example Conversion With Maximum Resolution

- Given $X = 3.3$ V;
  Calculate Y And N
- Maximum Resolution With Largest Possible Y
  - $Y_{max} = 1023$
- Largest Possible Y => Smallest 2^{-N}
  - Smallest LSB
  - Largest $|N|$

Example Conversion With Maximum Resolution

- Start By Finding N
- Can Solve Directly
  - But Complicated
- Or: Start By Dividing Y_{max} By X
- Examine Result And Find Largest 2^{-N} That Is Less Than The Result
  - This Gives N
- Multiply X By 2^{-N} To Get Y
  - Convert To 11 Bit Signed Binary Integer

$$N = \mathrm{int}\left(\log_2\left(\frac{X}{Y_{MAX}}\right)\right) = \mathrm{int}\left(\frac{\ln\left(\frac{X}{Y_{MAX}}\right)}{\ln 2}\right)$$

$$\frac{Y_{MAX}}{X} = \frac{1023}{3.3} = 310.0$$

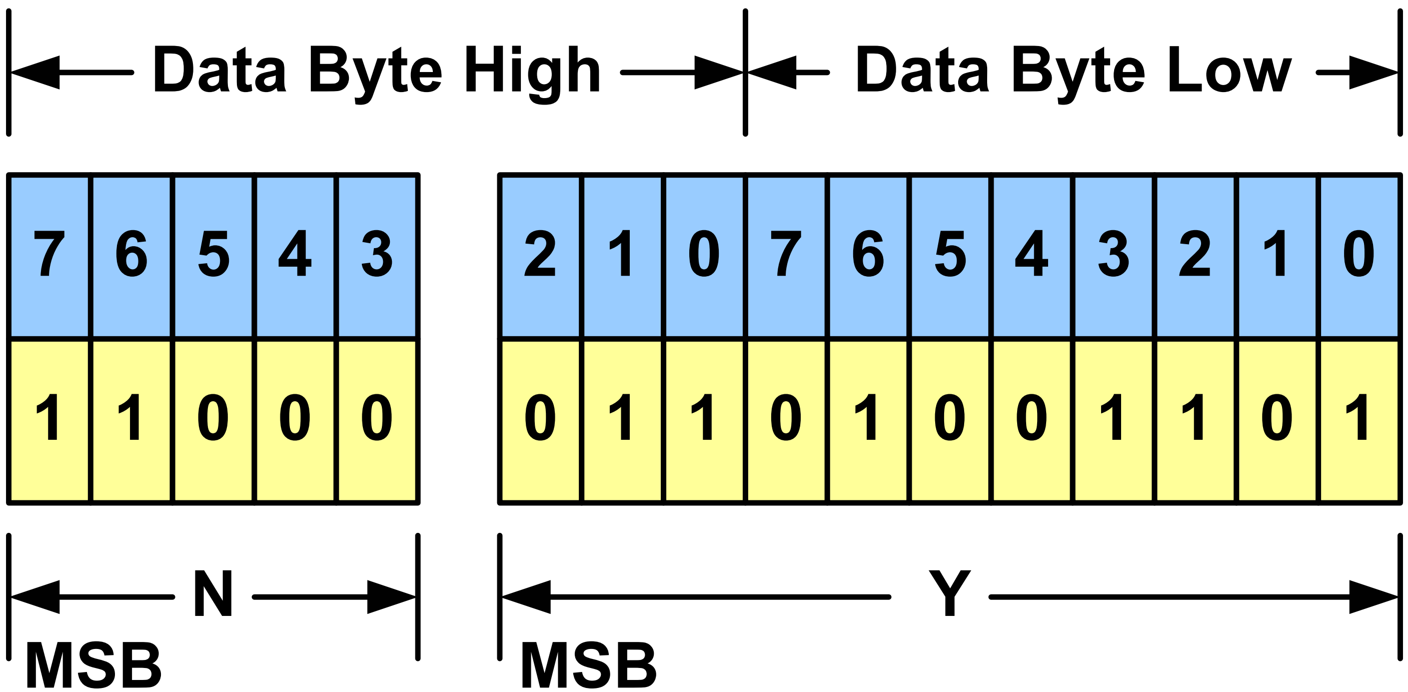
$$\max(2^{-N}) < 310.0 \Rightarrow 256$$
$$\Rightarrow N = -8 = 11000\mathrm{b}$$

$$3.3 \times 2^{-N} = 3.3 \times 2^{8}$$
$$= 3.3 \times 256$$
$$= 844.8 \Rightarrow 845$$
$$845 \Rightarrow 01101001101\mathrm{b} = Y$$

PMBus Power Management Bus Implementers Forum

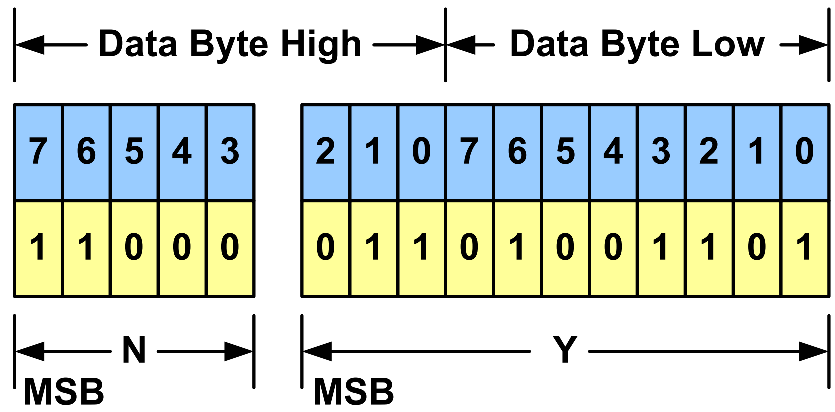
SMI System Management Interface Forum

Literal Mode
Result Sent Over The PMBus



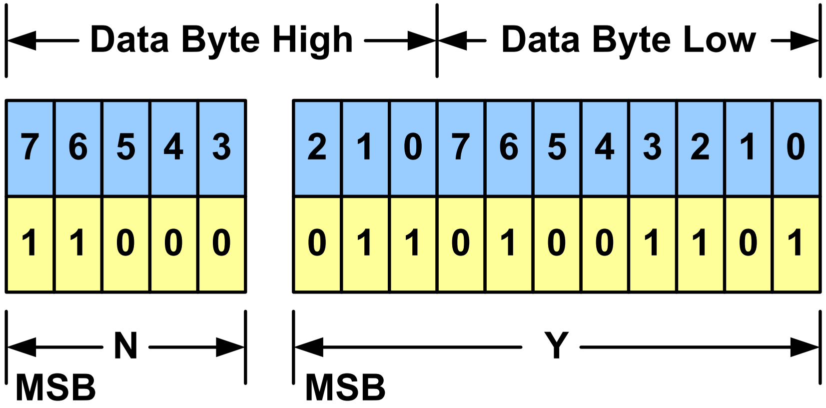
Literal Mode As A Non-Integer Binary Value

- Can Think Of N As Telling The Device How Many Binary Places To Move The Binary Point: 11.0100110b



Literal Mode As A Non-Integer Binary Value

- Can Think Of N As Telling The Device How Many Binary Places To Move The Binary Point: 11.0100110b

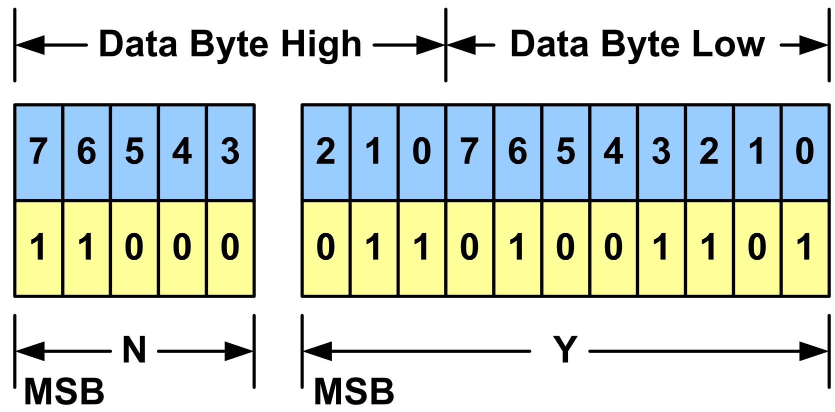


N = –8

Move Binary Point 8 Places To The Left

Literal Mode As A Non-Integer Binary Value

- Can Think Of N As Telling The Device How Many Binary Places To Move The Binary Point: 11.0100110b



- Can Also Think Of This As A Sum Of Powers Of 2

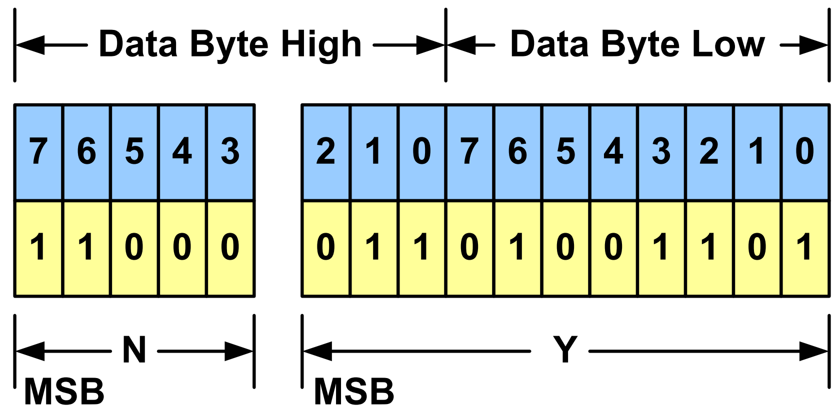
$$1 \bullet 2 + 1 \bullet 1 + 0 \bullet \frac{1}{2} + 1 \bullet \frac{1}{4} + \ldots$$

$$0 \bullet \frac{1}{8} + 0 \bullet \frac{1}{16} + 1 \bullet \frac{1}{32} + \cdots$$

$$1 \bullet \frac{1}{64} + 0 \bullet \frac{1}{128} + 1 \bullet \frac{1}{256}$$

Literal Mode As A Non-Integer Binary Value

- Can Think Of N As Telling The Device How Many Binary Places To Move The Binary Point: 11.0100110b



- Can Also Think Of This As A Sum Of Powers Of 2

$$1 \bullet 2 + 1 \bullet 1 + 0 \bullet \frac{1}{2} + 1 \bullet \frac{1}{4} + \ldots$$

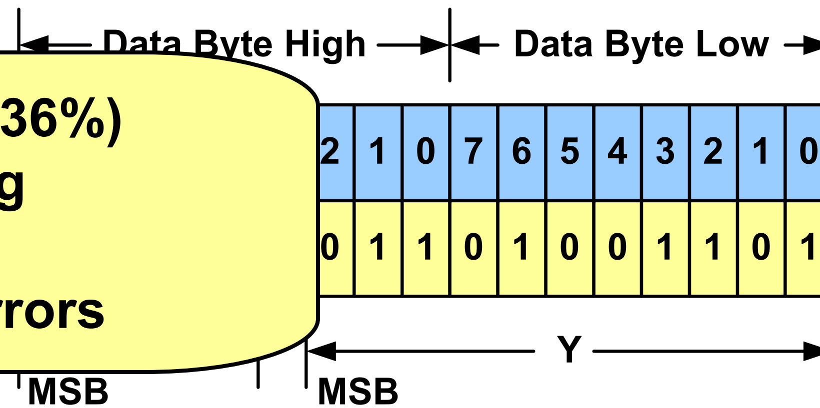
$$0 \bullet \frac{1}{8} + 0 \bullet \frac{1}{16} + 1 \bullet \frac{1}{32} + \cdots$$

$$1 \bullet \frac{1}{64} + 0 \bullet \frac{1}{128} + 1 \bullet \frac{1}{256}$$

- Result = 3.0078

Literal Mode As A Non-Integer Binary Value

- Can Think Of N As
  Te
  Ma
  Mo
  11.0



Error Of 7.8 mV (0.0236%)
Is From Rounding
848.8 To 845
And Quantization Errors

- Can Also Th Of This As A Sum Of owers Of 2

$$1 \bullet 2 + 1 \bullet 1 + 0 \bullet \frac{1}{2} + 1 \bullet \frac{1}{4} + \ldots$$

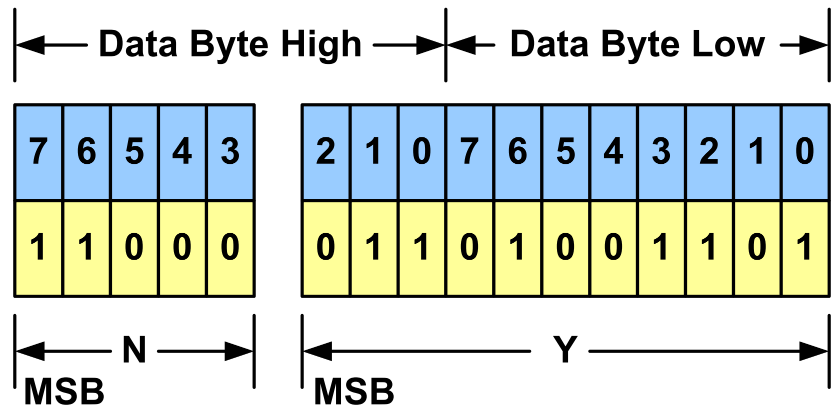
$$0 \bullet \frac{1}{8} + 0 \bullet \frac{1}{16} + 1 \bullet \frac{1}{32} + \cdots$$

$$1 \bullet \frac{1}{64} + 0 \bullet \frac{1}{128} + 1 \bullet \frac{1}{256}$$

- Result = 3.0078

Literal Mode As
LSB Size & Number Of LSBs

- Also Can Think Of This As
  - The LSB Size (2^N)
  - The Number Of LSBs (Y)
- LSB = $2^N = 2^{-8}$
  = 0.00390625
- Y = Number Of LSBs
  = 845
- $X = 845 \times 0.00390625$
  = 3.30078125



Example Decode

- Received Value: 11100011 01100111

High Byte (Received Second)

Low Byte (Received First)

Example Decode

- Received Value: 11100011 01100111

- Separate Into *N* And *Y*

$$11100 \quad 01101100111$$

$$N = 11100b = -4$$

$$Y = 01101100111b = 871$$

Example Decode

- Received Value:

$$11100011 \quad 01100111$$

- Separate Into *N* And *Y*

$$11100 \quad 01101100111$$

$$N = 11100\text{b} = -4$$

$$Y = 01101100111\text{b} = 871$$

- Calculate *X*

$$X = Y \bullet 2^{N} = 871 \times 2^{-4}$$

$$= \frac{871}{16} = 54.438$$

Power Management Bus Implementers Forum

System Management Interface Forum

Example Decode

- Received Value: $11100011 \quad 01100111$
- Separate Into *N* And *Y*

$$11100 \quad 01101100111$$
$$N = 11100\text{b} = -4$$
$$Y = 01101100111\text{b} = 871$$

- Calculate *X*

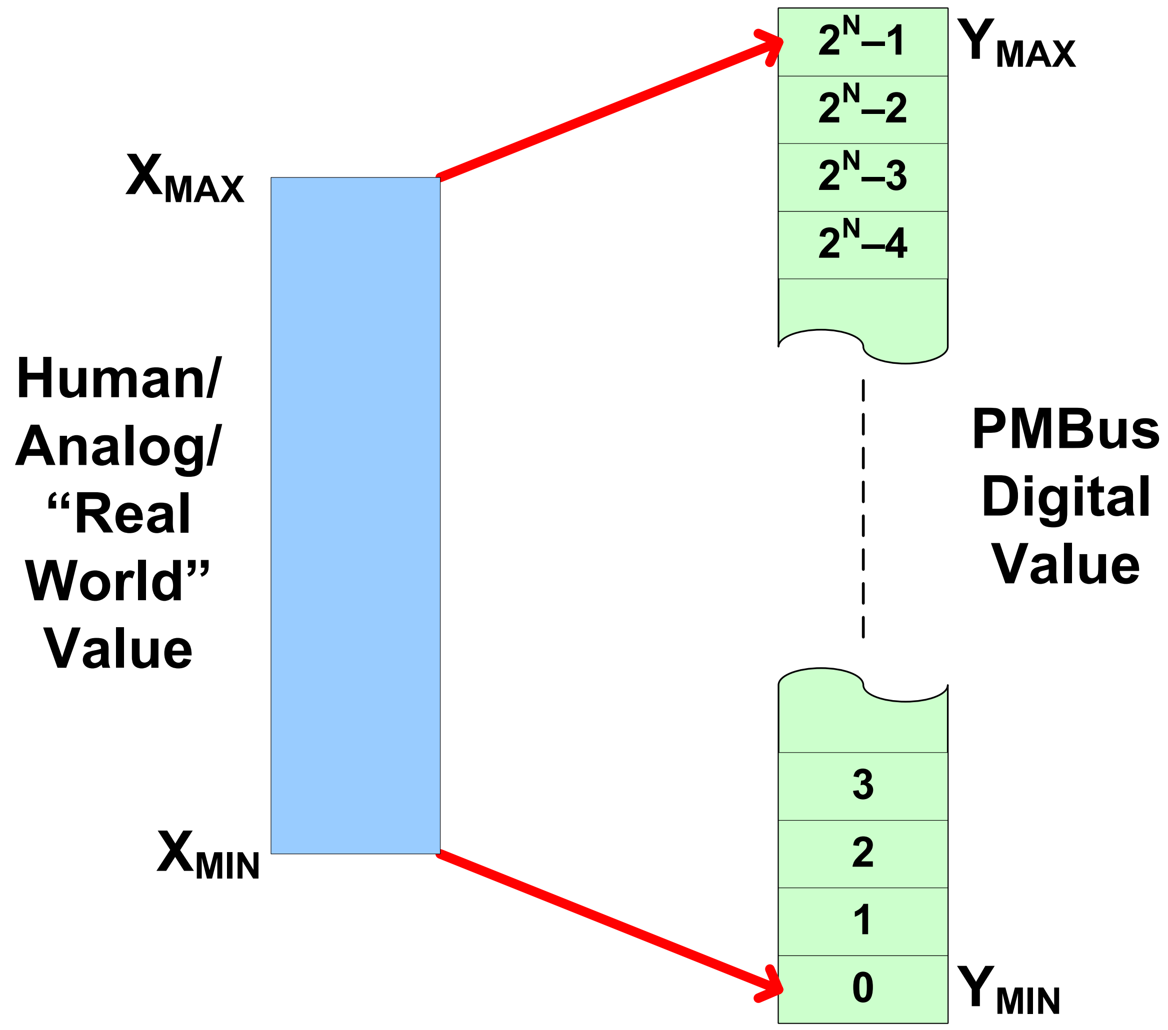
$$X = Y \bullet 2^{N} = 871 \times 2^{-4}$$
$$= \frac{871}{16} = 54.438$$

- Original Value: 54.46
  – Error: 22 mV => 0.040%

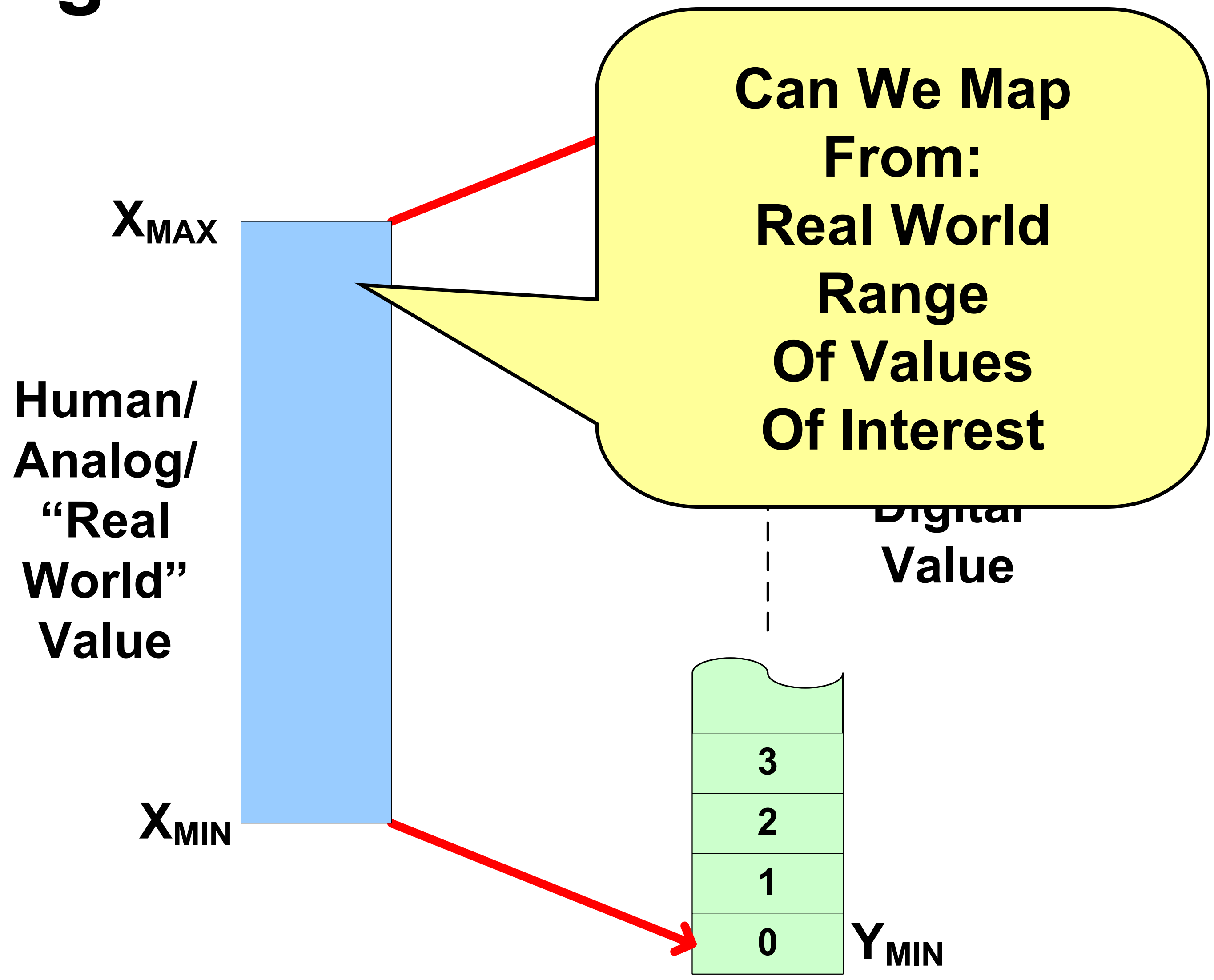
Error: We Got Lucky!

- Suppose Full Scale Was 60 V
- Resolution:
  60 V ÷ 1023 = 58.65 mV/bit
- Some Applications , Such As A Telecomm Rectifier, Need A Much Finer Resolution
  - Typically 10-20 mV/bit
- But Range Of Interest Is Not 0 V to 60 V, More Like 42 V To 58 V

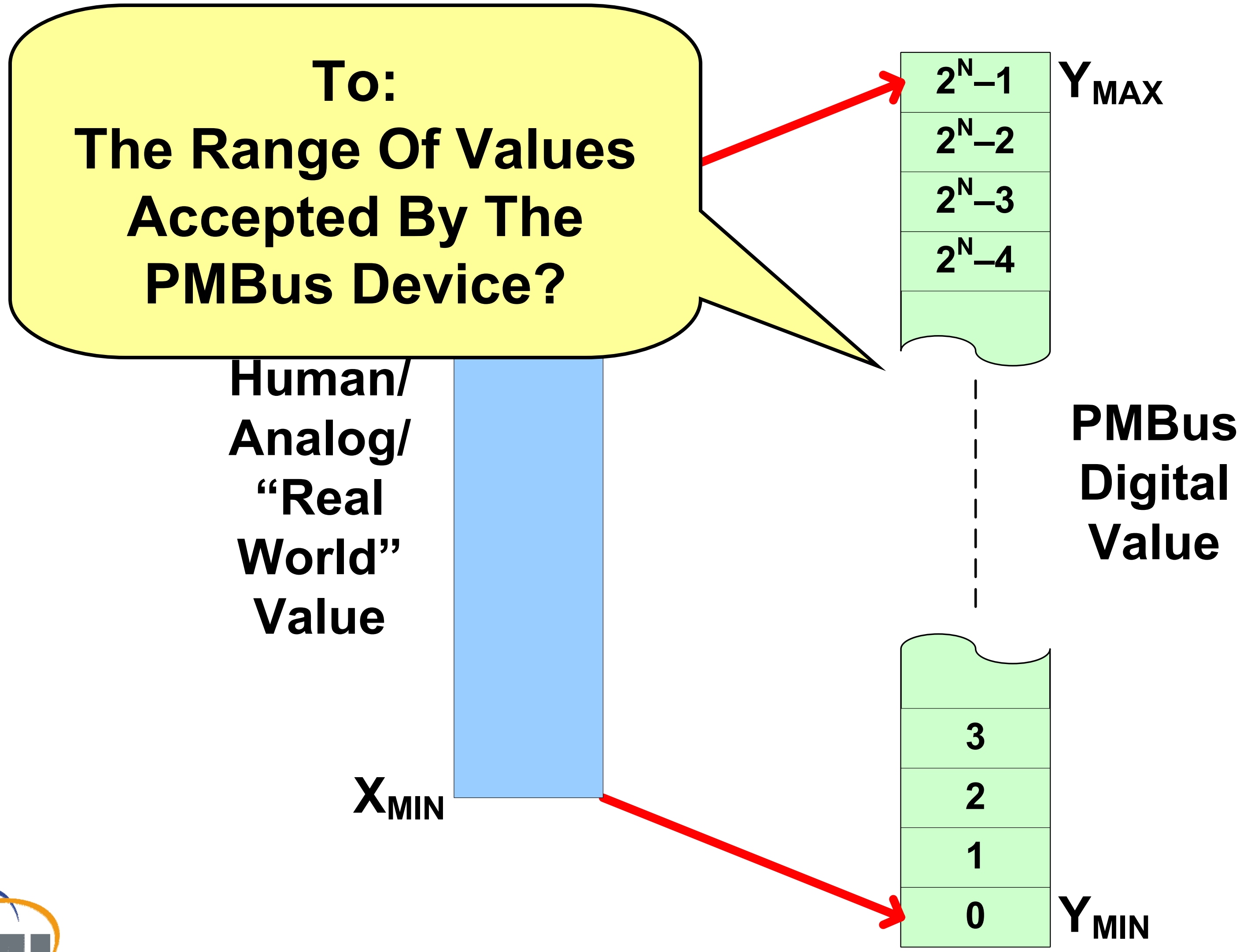
Scaling With Offset



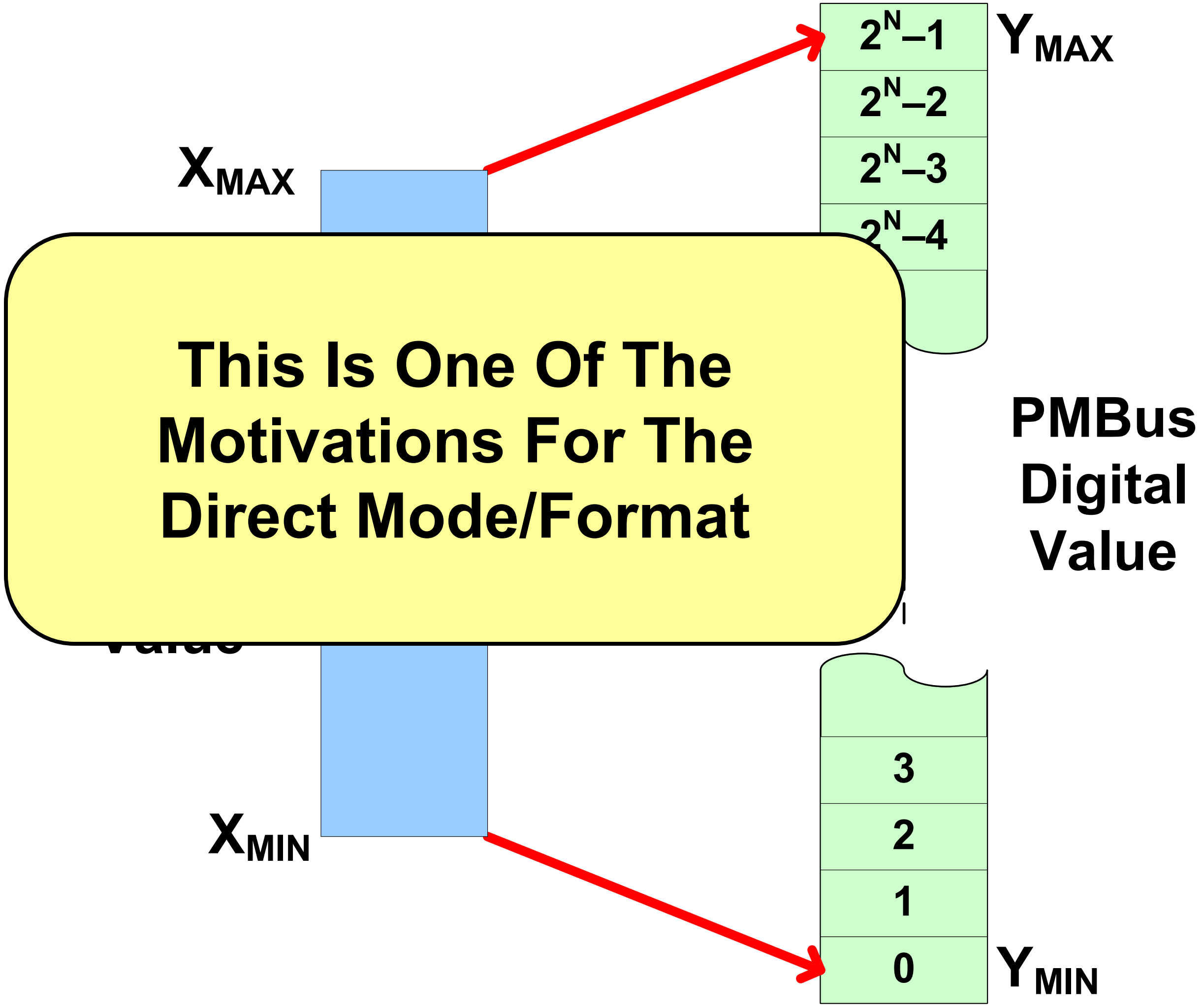
Scaling With Offset



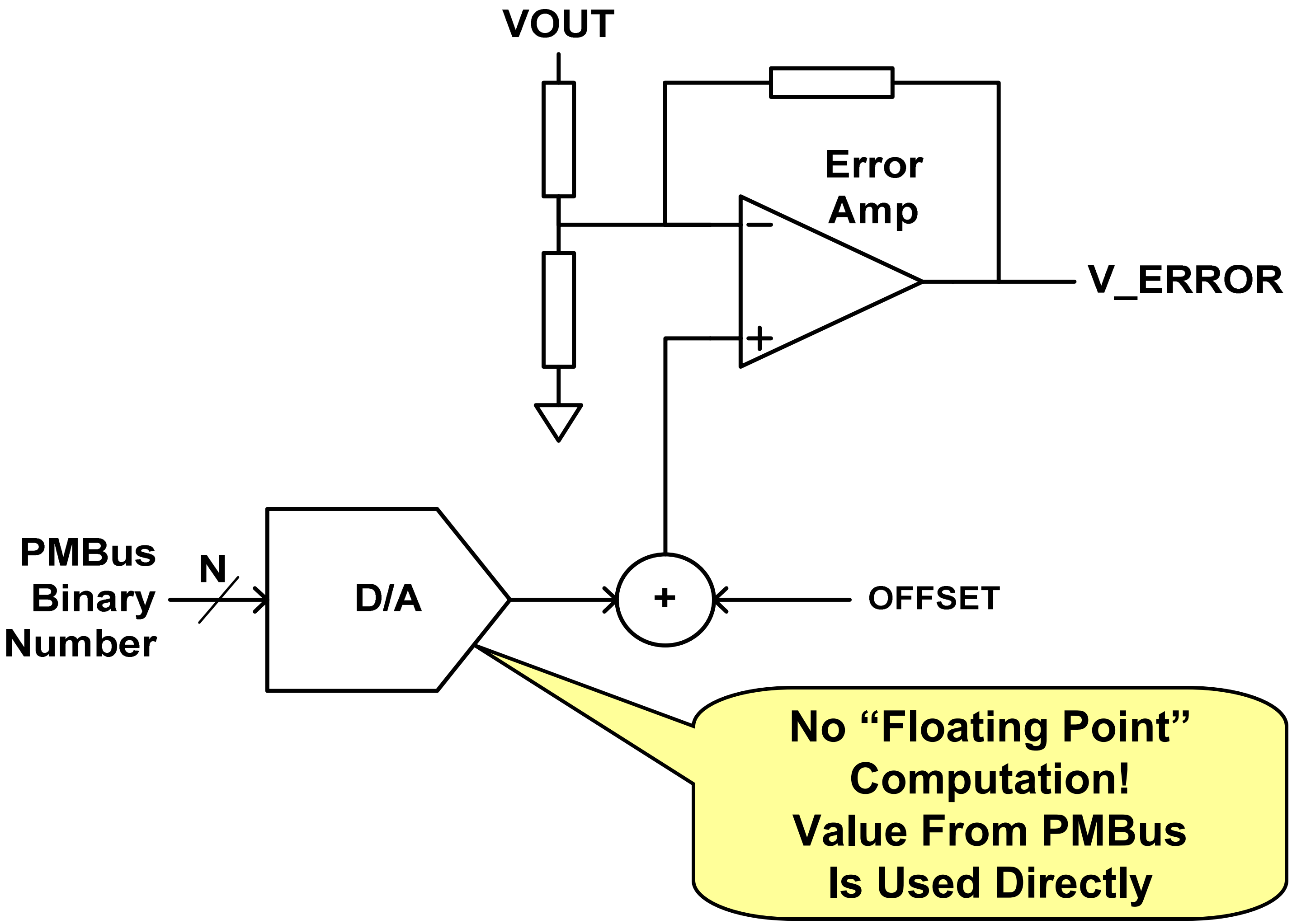
Scaling With Offset



Scaling With Offset



More Direct Mode Motivation



Direct Mode Equation

- The Direct Mode Uses An Equation As Follows:

$$Y = (mX + b) \bullet 10^{R}$$

- Where:
  - *Y* Is The Value Transmitted To Or Received From The PMBus Device (16 Bits, Signed)
  - *X* Is The “Human” Value To Be Encoded
  - *m* Is The Scaling Coefficient (16 Bits, Signed)
  - *b* Is The Offset Coefficient (16 Bits, Signed)
  - *R* Is The Scaling Coefficient (8 Bits, Signed)

Direct Mode Equation

NOTICE!

This Is The Form That Will Appear In The PMBus Specification Revision 1.1

This Is "Backwards" From What Is In Specification 1.0 Section 7

– *b* Is The Offset Coefficient (16 Bits, Signed)
– *R* Is The Scaling Coefficient (8 Bits, Signed)

Direct Mode: *m*, *b* And *R*

- *m*, *b* And *R* Are Known As The Coefficients
- They Are Supplied By The PMBus Device Manufacturer
- Preferred:
  Coefficients Stored In The Device And Retrieved By The Host With The COEFFICIENTS Command
- Alternative:
  Coefficients Are Provided In The Product Literature (Data Sheet, Application Note)

Calculating The Coefficients

- Problem
  - 3 Unknowns (*m*, *b*, *R*)
  - 2 Constraints
- The Two Constraints
  - X_{min} => Y_{min} And X_{max} => Y_{max}
- Solution Procedure
  - Assume *R* Is Known And Fixed
  - Solve For m And b In Terms Of X_{min}, X_{max}, Y_{min}, Y_{max}
  - Use A Tool Like Excel To Solve For *m* And *b* For Several Values Of *R*
  - Choose Largest Possible Values Of *m* And *b*

Calculating The Coefficients

- The Constraints

$$X_{\min} \Rightarrow Y_{\min} = 0$$

$$X_{\max} \Rightarrow Y_{\max} = 2^n - 1$$

- Substituting Into The Direct Mode Equation

$$Y_{\min} = (mX_{\min} + b) \bullet 10^R$$

$$Y_{\max} = (mX_{\max} + b) \bullet 10^R$$

- Solving For *m* And *b*

$$m = \left(\frac{Y_{\max} - Y_{\min}}{X_{\max} - X_{\min}} \right) \bullet 10^{-R}$$

$$b = \left(Y_{\min} - \frac{Y_{\max} - Y_{\min}}{X_{\max} - X_{\min}} X_{\min} \right) \bullet 10^{-R}$$

$$= \left(Y_{\max} - \frac{Y_{\max} - Y_{\min}}{X_{\max} - X_{\min}} X_{\max} \right) \bullet 10^{-R}$$

Calculating The Coefficients Example

- AC-DC Rectifier For Telecom Applications
  - Wide Range Of Output Voltage To Control Battery Charging
  - Resolution In Range Of 10–20 mV
- Number Of Bits For Input: 10
  - Y_{min} = 000h
  - Y_{max} = 1023d = 3FFh = 1111111111b
- Output Voltage Range
  - X_{min} = 44 Vdc
  - X_{max} = 58 Vdc
  - Resolution: 13.69 mV/bit

Calculating The Coefficients Example

- Using Microsoft Excel to Solve For *m* And *B* For Various Values Of *R* Yields:

R	m (calculated)	b (calculated)	m (rounded)	b (rounded)
-4	730714.2857	-32151428.57	730714	-32151429
-3	73071.42857	-3215142.857	73071	-3215143
-2	7307.142857	-321514.2857	7307	-321514
-1	730.7142857	-32151.42857	731	-32151
0	73.07142857	-3215.142857	73	-3215
1	7.307142857	-321.5142857	7	-322
2	0.730714286	-32.15142857	1	-32
3	0.073071429	-3.215142857	0	-3
4	0.007307143	-0.321514286	0	0
5	0.000730714	-0.032151429	0	0

Calculating The Coefficients Example

- Using Microsoft Excel to Solve For *m* And *B* For Various Values Of *R* Yields:

R	m (calculated)	b (calculated)	m (rounded)	b (rounded)
-4	730714.2857	-32151428.57	730714	-32151429
-3	73071.42857	-3215142.857	73071	-3215143
-2	7307.142857	-321514.2857	7307	-321514
-1	730.7142857	-32151.42857	731	-32151
0	73.07142857	-3215.1428		
1	7.307142857	-321.514		
2	0.730714286	-32.1514		
3	0.073071429	-3.21514		
4	0.007307143	-0.32151		
5	0.000730714	-0.03215		

Values In Red Exceed The Range Of Values Available To A 16 Bit Signed Integer (+32,767 To –32,768)

Calculating The Coefficients Example

- Us… For *m* And *B*
  Fo… :

For Best Resolution,
Choose Largest Possible
Values Of *m* And *b*
***m* = 731**
***b* = -32151**
***R* = –1**

R			…unded)	b (rounded)
-4			…730714	-32151429
-3	73071.42857	-3215142.[illegible]	73071	-3215143
-2	7307.142857	-321514.2857	7307	-321514
-1	730.7142857	-32151.42857	731	-32151
0	73.07142857	-3215.142857	73	-3215
1	7.307142857	-321.5142857	7	-322
2	0.730714286	-32.15142857	1	-32
3	0.073071429	-3.215142857	0	-3
4	0.007307143	-0.321514286	0	0
5	0.000730714	-0.032151429	0	0

Calculating The Coefficients Example

- Chosen Solution

  m: 731

  b: –32151

  R: –1

- Double Check Calculation

$$Y_{\min} = (mX_{\min} + b) \bullet 10^{R}$$
$$= (731 \bullet 44 - 32{,}151) \bullet 10^{-1}$$
$$= 1.3 \neq 0$$

$$Y_{\max} = (mX_{\max} + b) \bullet 10^{R}$$
$$= (731 \bullet 58 - 32{,}151) \bullet 10^{-1}$$
$$= 1024.7 \neq 1023$$

More Rounding And Quantization Errors!

Calculating The Coefficients Example

- Example
  - Minimum Voltage (X_{min}): 44 V
  - Maximum Voltage (X_{max}): 58 V
  - PMBus Device Resolution: 16 Bits

R	m (calculated)	b (calculated)	m (rounded)	b (rounded)
-2	468107.1429	-20596714.29	468107	-20596714
-1	46810.71429	-2059671.429	46811	-2059671
0	4681.071429	-205967.1429	4681	-205967
1	468.1071429	-20596.71429	468	-20597
2	46.81071429	-2059.671429	47	-2060
3	4.681071429	-205.9671429	5	-206
4	0.468107143	-20.59671429	0	-21

Calculating The Coefficients Example

- Chosen Solution

  m: 468

  b: –20597

  R: 1

- Double Check Calculation

$$Y_{\min} = (mX_{\min} + b) \cdot 10^{R} = (468 \cdot 44 - 20{,}597) \cdot 10^{+1} = -5 \neq 0$$

$$Y_{\max} = (mX_{\max} + b) \cdot 10^{R} = (468 \cdot 58 - 20{,}597) \cdot 10^{+1} = 65{,}470 \neq 65{,}535$$

Still Have Rounding And Quantization Errors!

What To Do?

- Choices
  - Live With It
  - Adjust The Slope (m)
  - Adjust the Offset (b)
  - Adjust Both
  - Adjust X_{max} And X_{min}
- Optimization Is Left As An Exercise For The Student

What To Do?

- Choices
  - Live With It
  - Adjust The Slope (m)
  -
  -
  -

Lesson:
You Must Pay Attention To Errors Introduced By Discrete Arithmetic!

- Optimization Is Left As An Exercise For The Student

Decoding Direct Mode Example

- Example Of Reading The Output Current Of An Isolated DC-DC Bus Converter
- Using COEFFICIENTS Command Returns Values For *m*, *b* And *R* As:
  - $m = 850$
  - $b = 0$
  - $R = -2$
- Using READ_IOUT Command Returns The Value 0000000001101001b => 105d

Decoding Direct Mode Example

- Use The Inverse Of The Equation Used To Encode

$$Y = (mX + b)\bullet 10^{R}$$

$$X = \frac{1}{m}(Y \bullet 10^{-R} - b)$$

- Substitute Values And Solve

$$X = \frac{1}{850}(105 \bullet 10^{-(-2)} - 0)$$

$$= \frac{10500}{850} = 12.35$$

- Output Current = 12.35 A

Note That These Calculations Are Done In The Host And Not The PMBus Device!

Setting The Output Voltage

Step 1
Which Data Format?
(aka Which Mode)

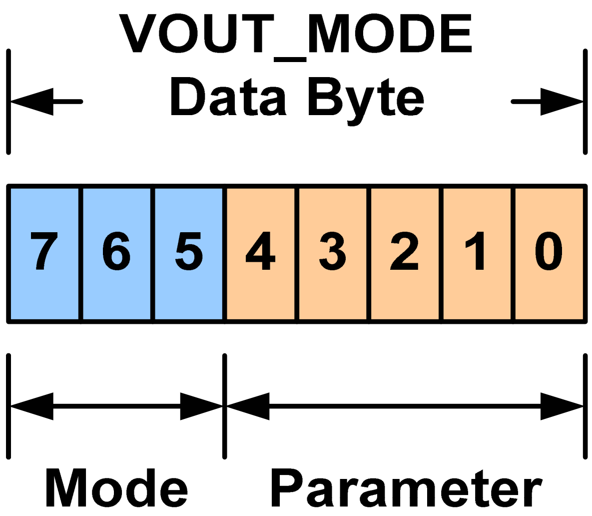
- Linear
- Direct
- VID

VOUT_MODE Command

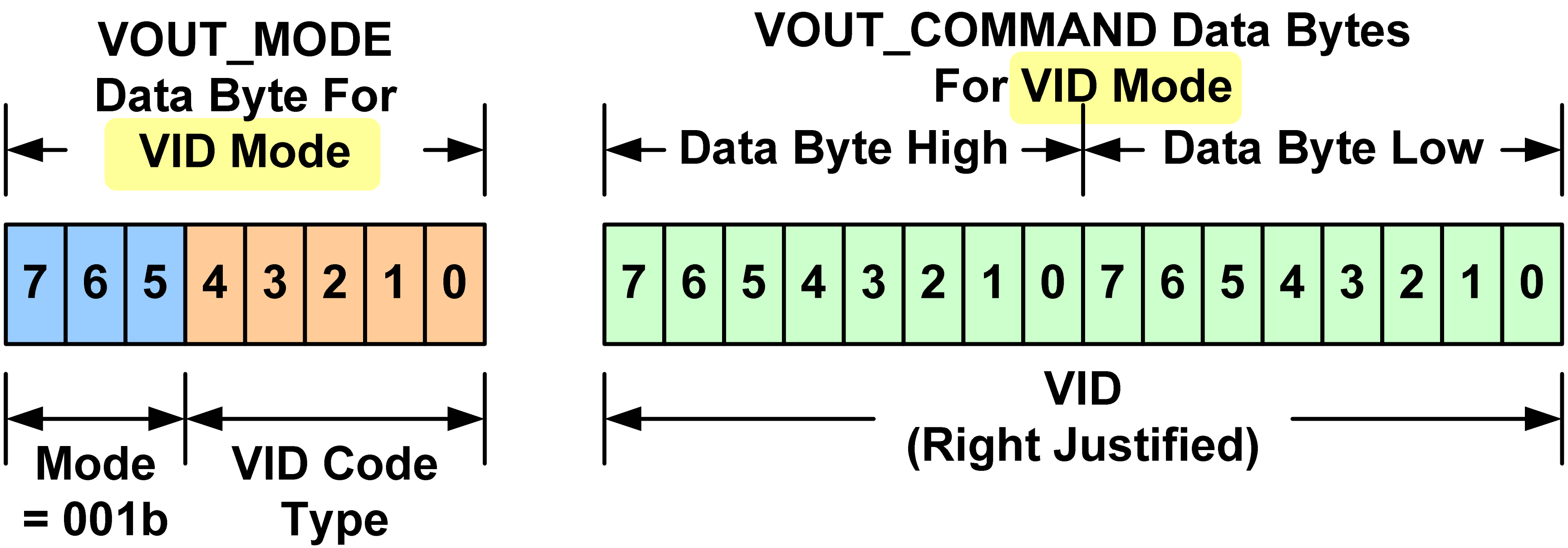
Step 2
Set The Output Voltage Using The VOUT_COMMAND Command

VOUT_MODE Command

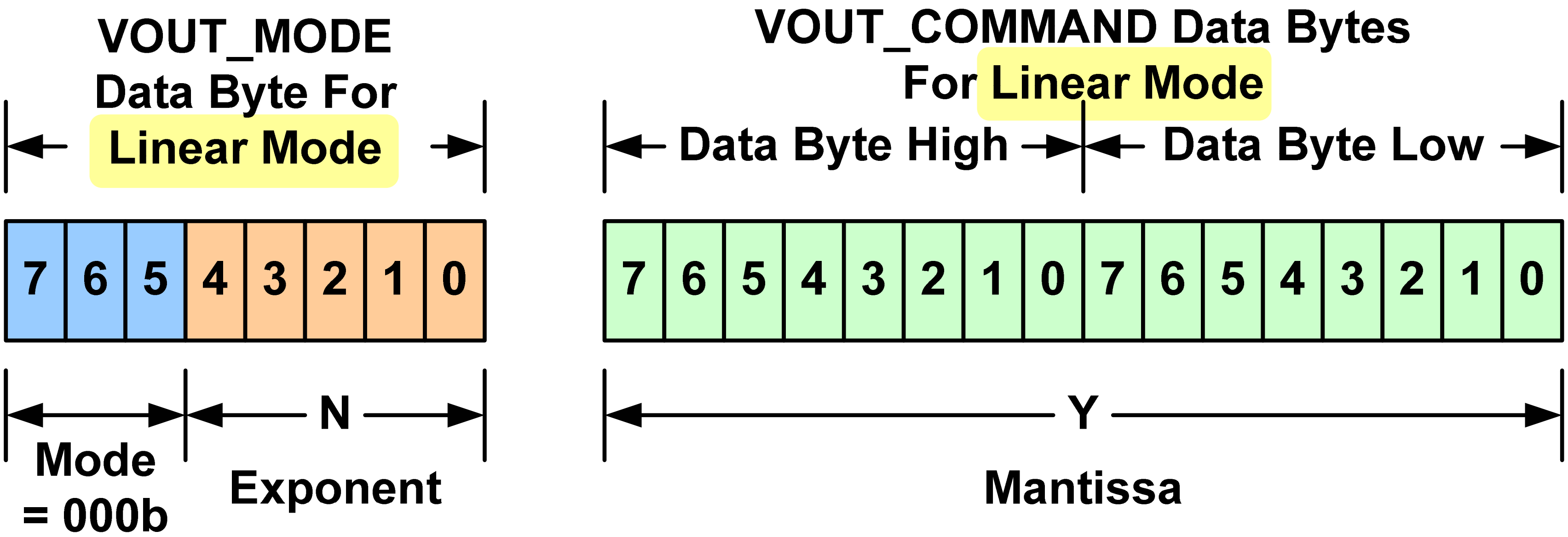
- VOUT_MODE Command Is Sent Separately From Any Other Command, Such As VOUT_COMMAND
- Sent Only When Necessary To Change The Mode
  - Only Once?
- Applies For All Output Voltage Related Commands



VOUT_MODE & VOUT_COMMAND

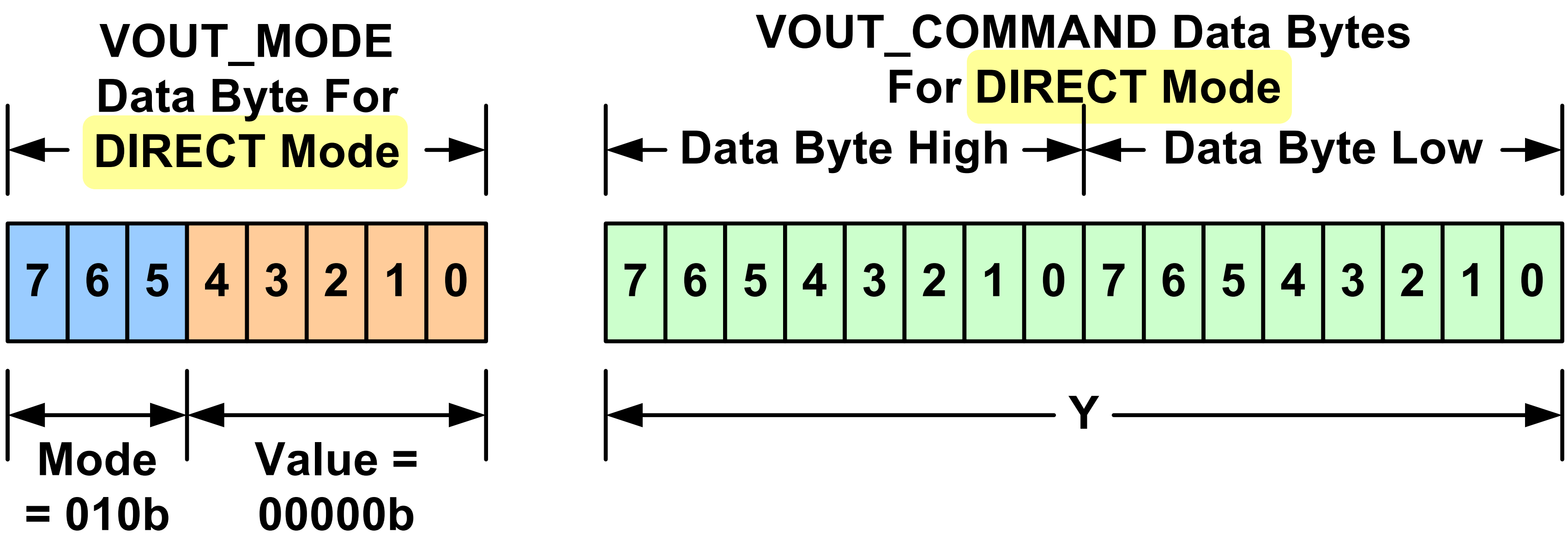


VOUT_MODE & VOUT_COMMAND

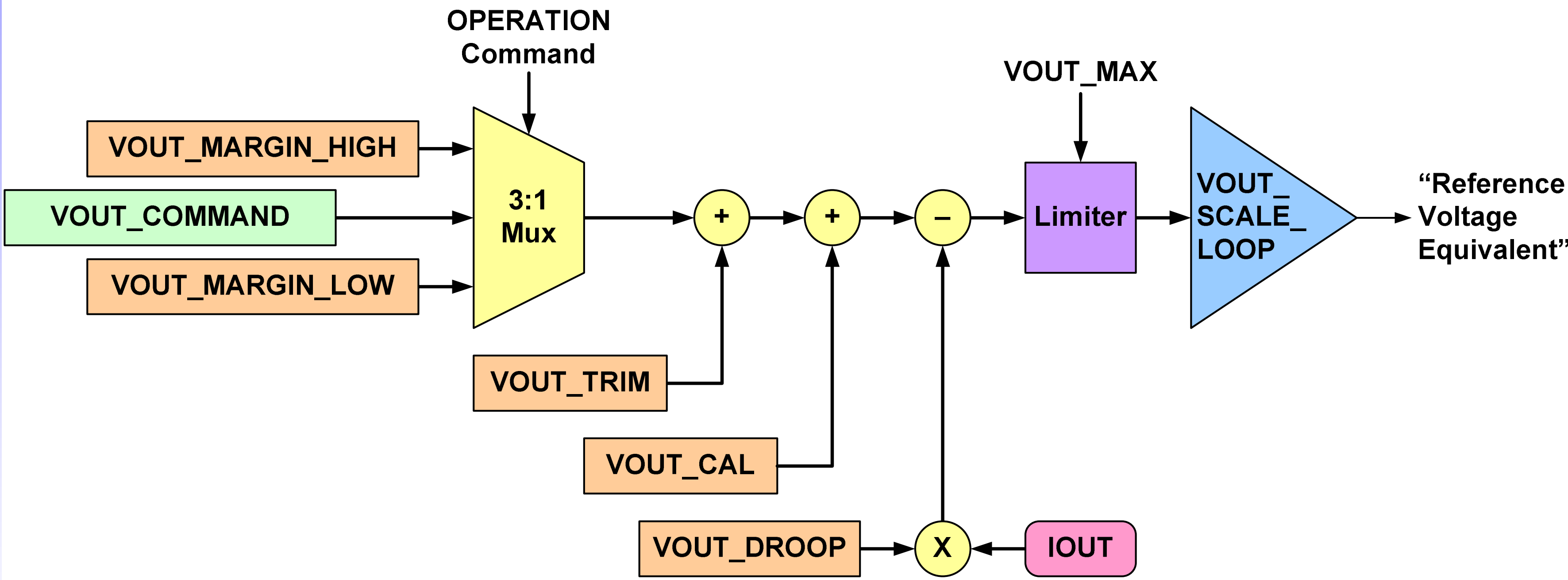


$$\text{Voltage} = Y \cdot 2^{N}$$

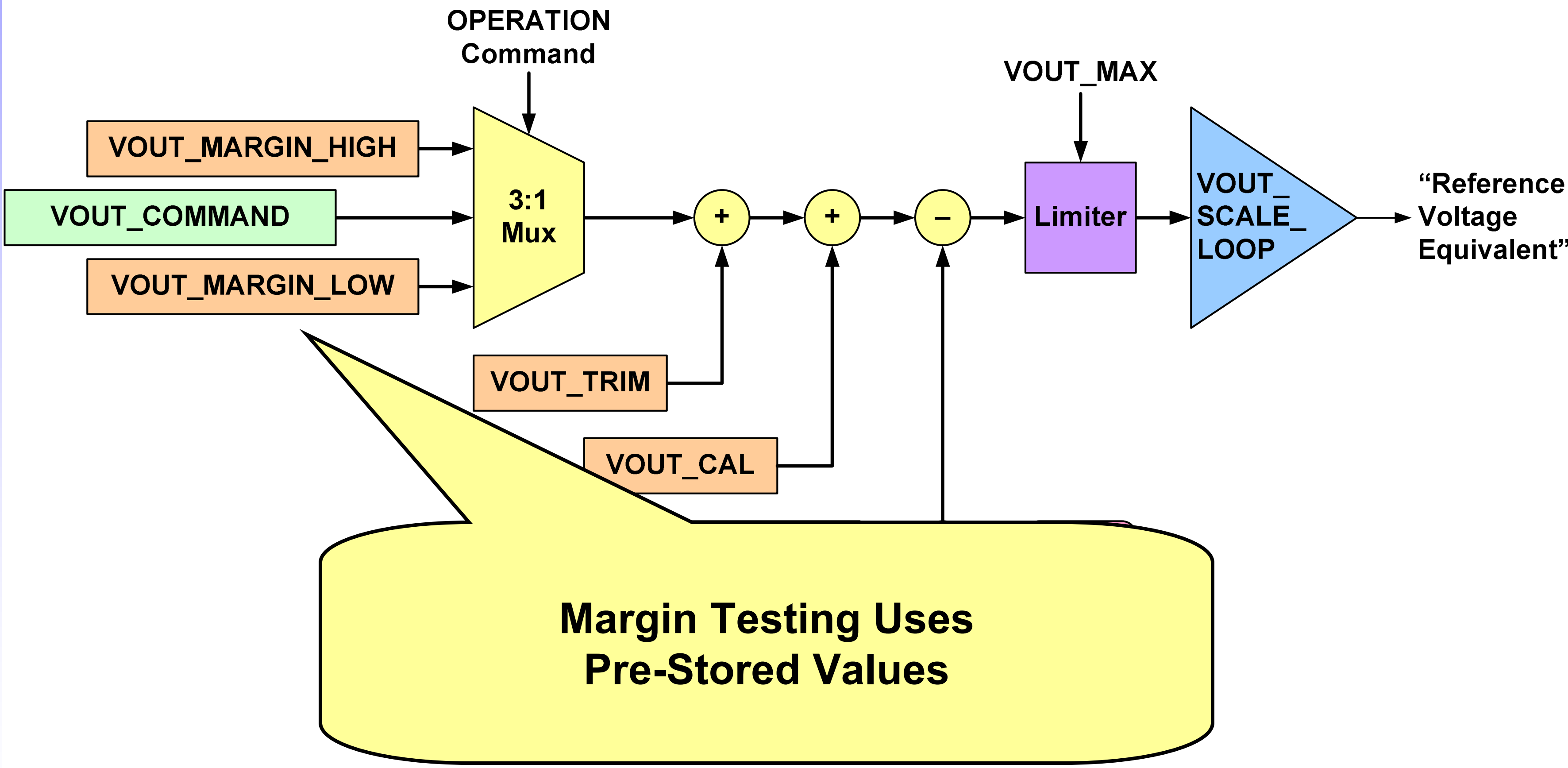
VOUT_MODE & VOUT_COMMAND



Fine Tuning The Output Voltage



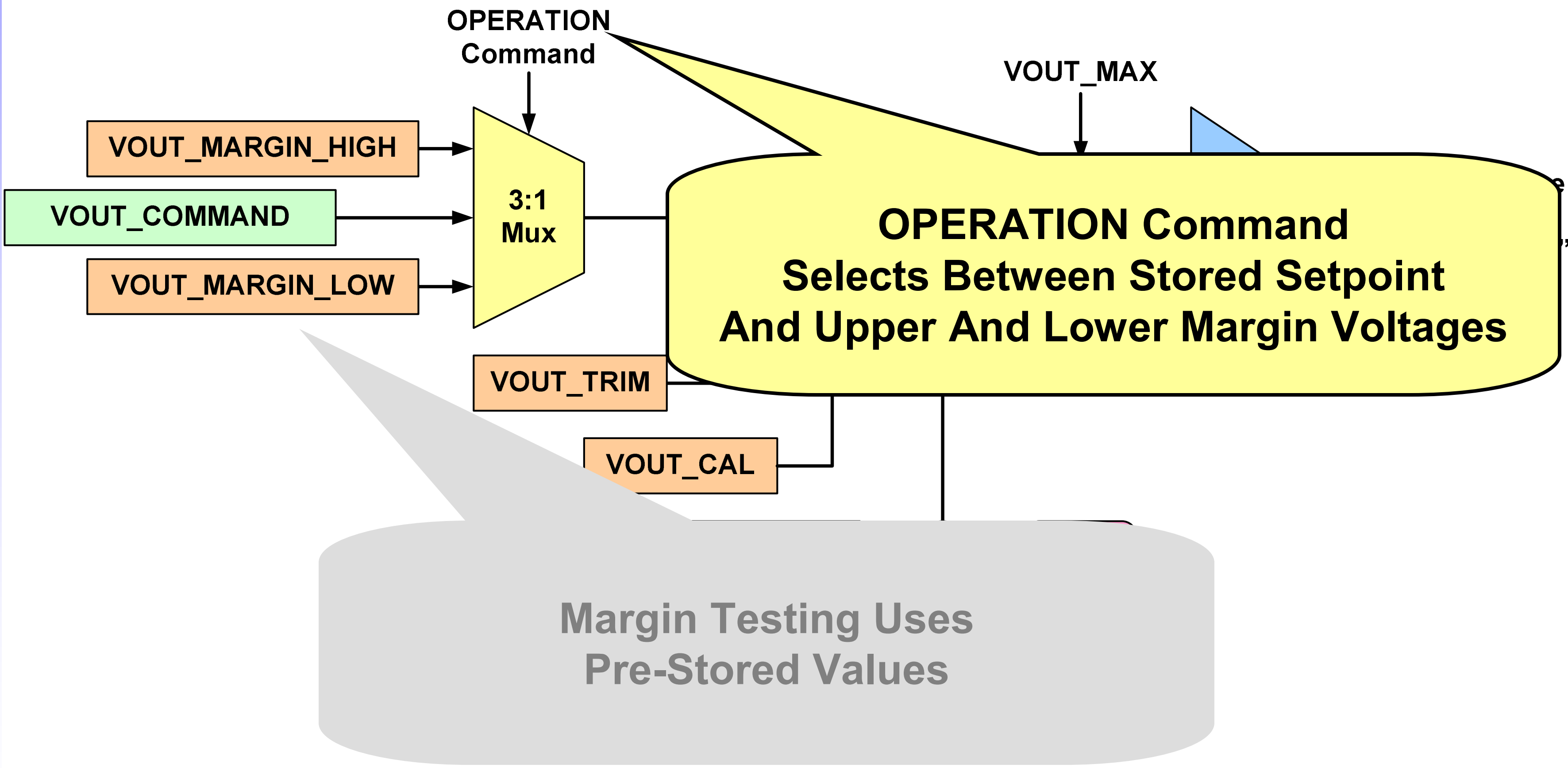
Margin Testing



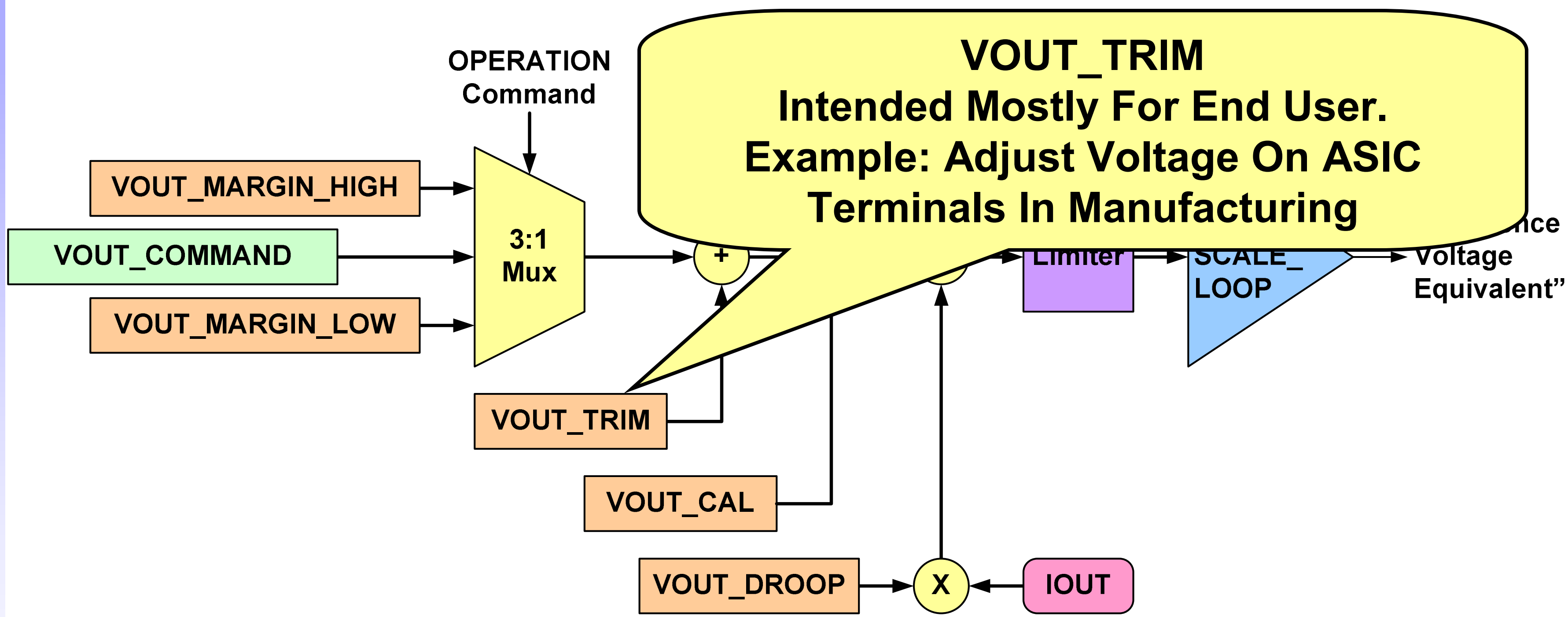
Power Management Bus Implementers Forum

System Management Interface Forum

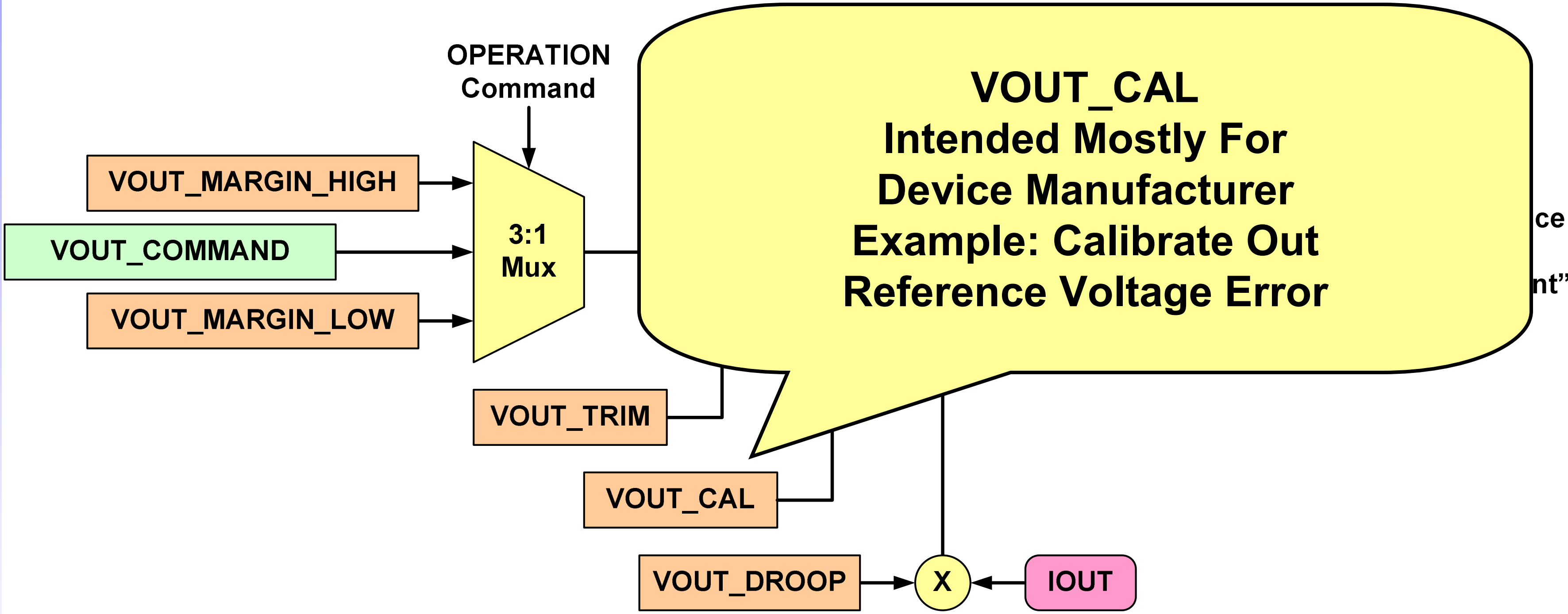
Margin Testing



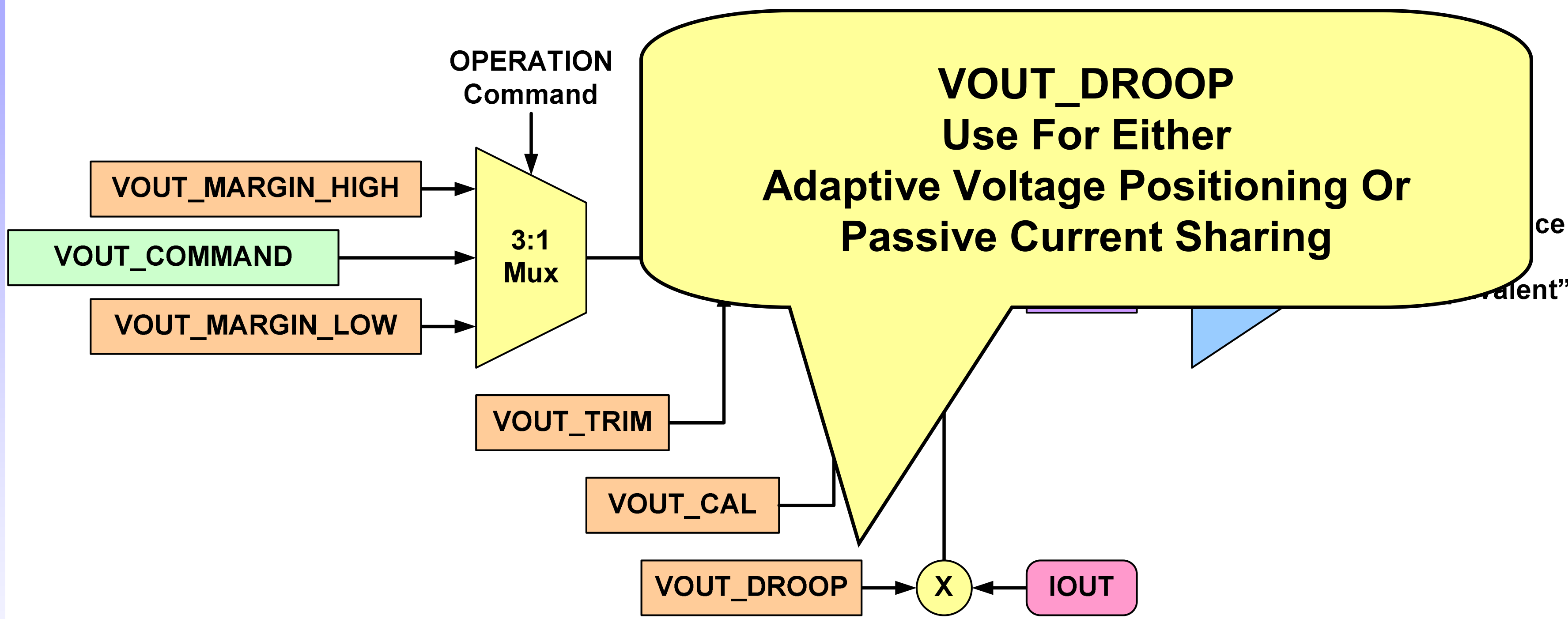
Fine Tuning The Output Voltage



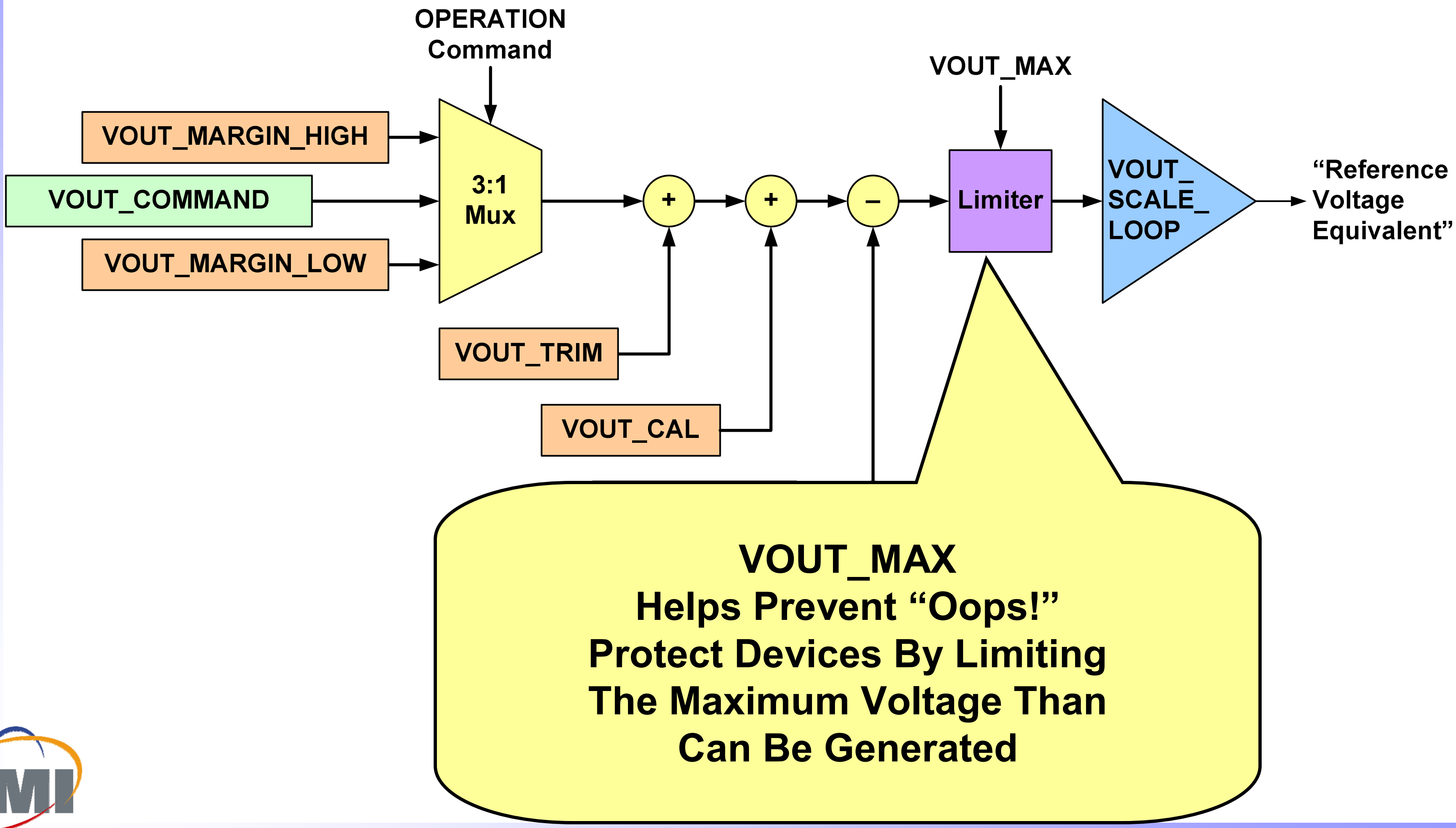
Fine Tuning The Output Voltage



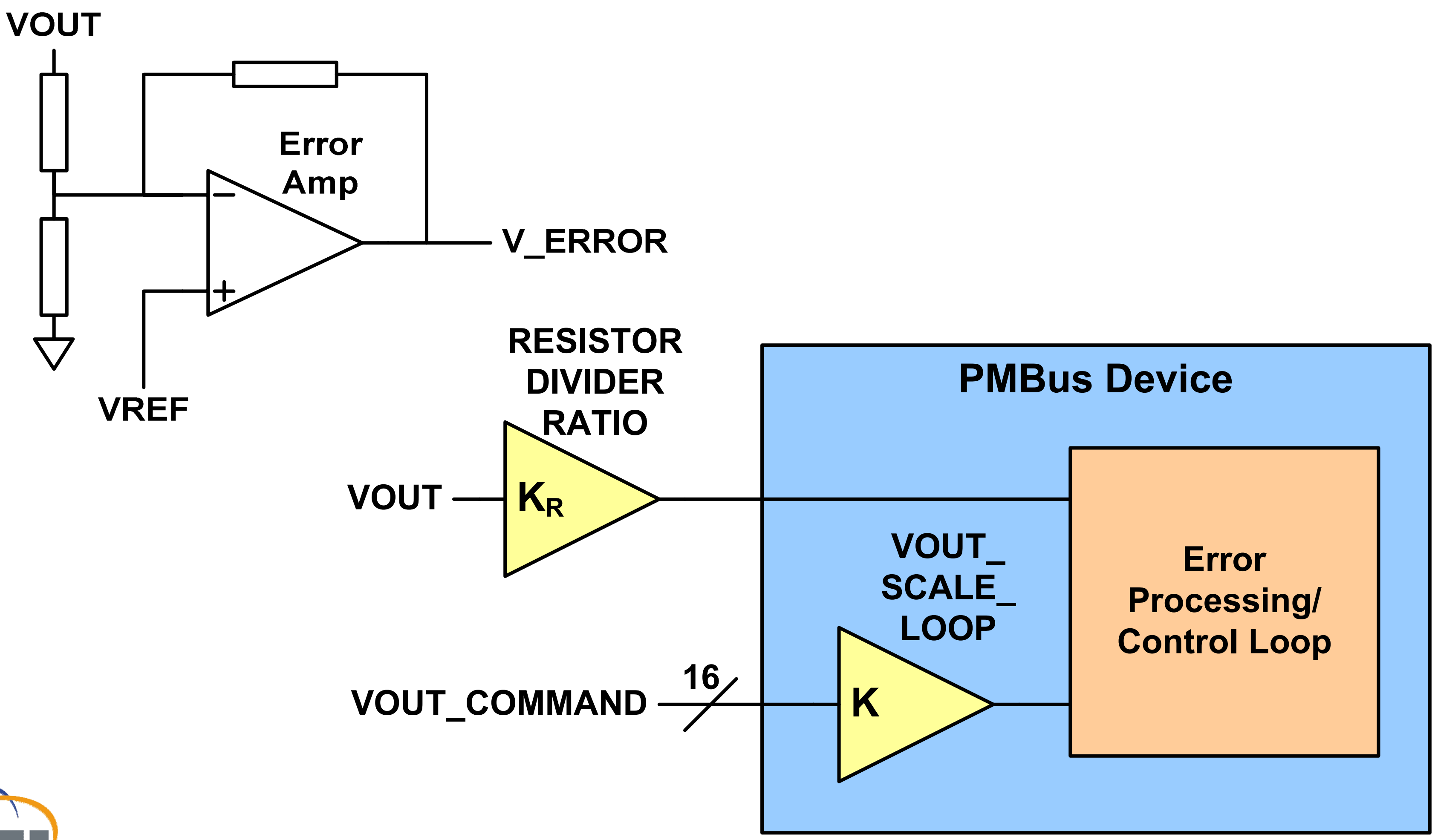
Fine Tuning The Output Voltage



Fine Tuning The Output Voltage



Using And External Divider

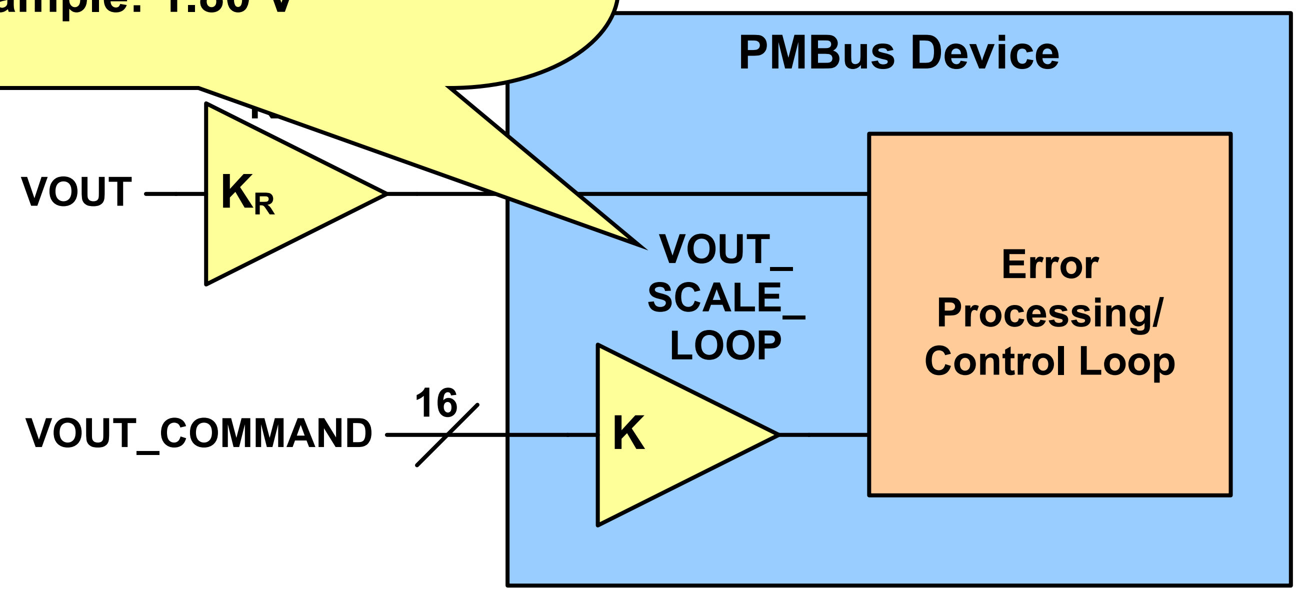


Using And External Divider

Simplifies Life
For The End User

They Do Not Need To Think About The Voltage Divider

Just Send Command Voltage As They Want It
Example: 1.80 V



On/Off Control

- Two Inputs Control Whether A PMBus Device Is Operating Or Not
  - Hardwired CONTROL Pin (Programmable Polarity)
  - OPERATION Command From The Bus
- On/Off Control Totally Programmable
- CONTROL Pin Options
  - Active High Or Active Low
  - Followed Programmed Sequencing Or Shutdown Immediately

ON_OFF_CONFIG

On/Off Control Mode	Device Power	CONTROL Input	Bus Command
Always ON	If Power, Then ON	X	X
Respond To CONTROL Only	If Power, Respond To CONTROL And Bus Commands As Programmed	Active High	Ignore Bus Commands
		Active Low	
Respond To Bus Only		Ignore CONTROL	Respond To Bus Commands
Respond To Both CONTROL And Bus		Active High	
		Active Low	

OPERATION Command

Bits [7:6]	Bits [5:4]	Bits [3:2]	Bits [1:0]	Unit On Or Off	Margin State
00	XX	XX	XX	IMMEDIATE OFF (No Sequencing)	N/A
01	XX	XX	XX	OFF (With Sequencing)	N/A
10	00	XX	XX	ON	OFF
10	01	01	XX	ON	MARGIN LOW (Ignore Fault)
10	01	10	XX	ON	MARGIN LOW (Act On Fault)
10	10	01	XX	ON	MARGIN HIGH (Ignore Fault)
10	10	10	XX	ON	MARGIN HIGH (Act On Fault)

OPERATION Command

Ignore Fault

Prevents Sending An Alarm
Or Responding To An
Output Undervoltage Condition
That Was Deliberately Caused
By Margin Testing

This Allows System Testing To
Proceed Without Special Modifications
To The Power Supply/DC-DC Converter

...ff	Margin State
OFF ...ing)	N/A
...cing)	N/A
	OFF
	MARGIN LOW (Ignore Fault)
	MARGIN LOW (Act On Fault)
	MARGIN HIGH (Ignore Fault)
	MARGIN HIGH (Act On Fault)

OPERATION Command

Bit…				Unit … Off	Margin State
				…OFF …ing)	N/A
				…cing)	N/A
					OFF
					MARGIN LOW (Ignore Fault)
					MARGIN LOW (Act On Fault)
					MARGIN HIGH (Ignore Fault)
					MARGIN HIGH (Act On Fault)

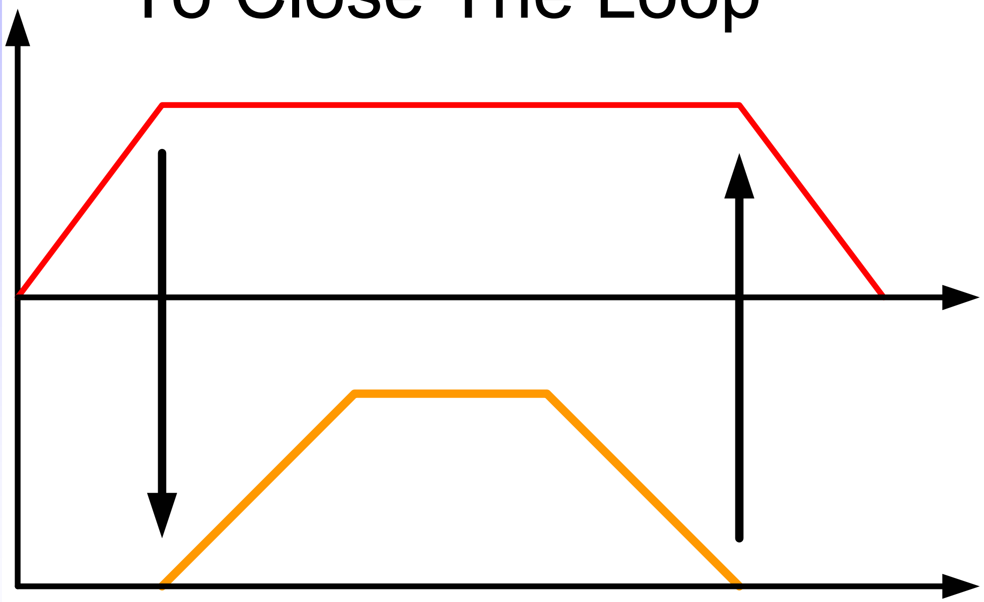
Act On Fault

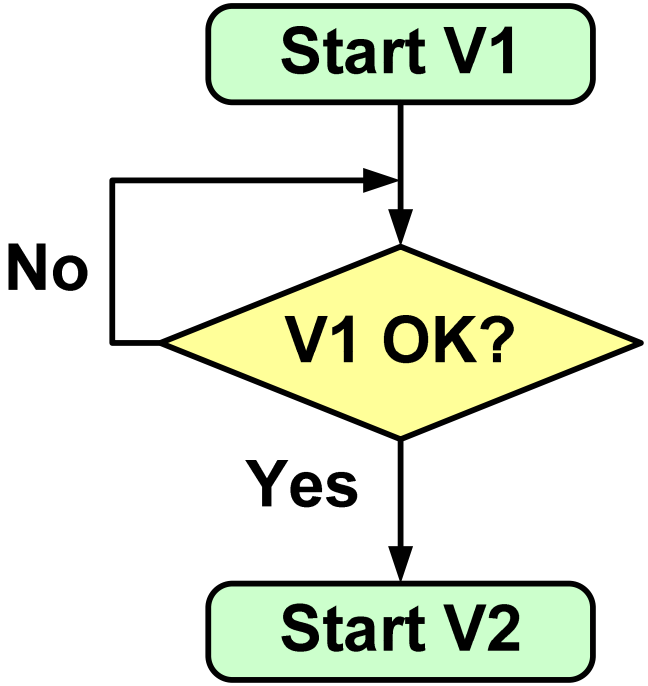
The PMBus Device Will Send An Alarm Or Respond To An Output Undervoltage Condition That Was Deliberately Caused By Margin Testing

This May Be Desired To Protect The System From Extreme Output Voltages

Sequencing: Event Driven

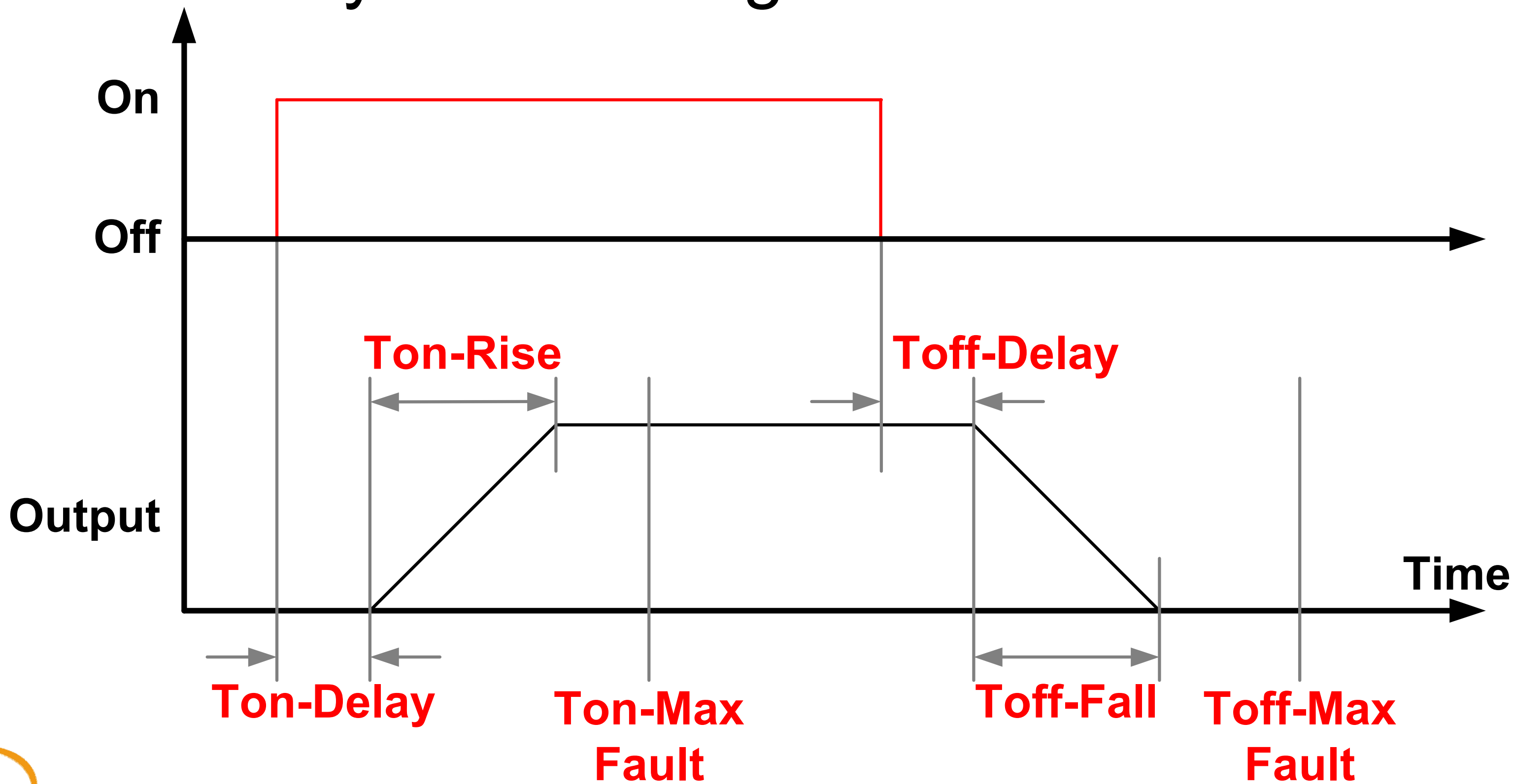
- Event Driven Sequencing Is Closed Loop
- Requires Power System Manager To Close The Loop



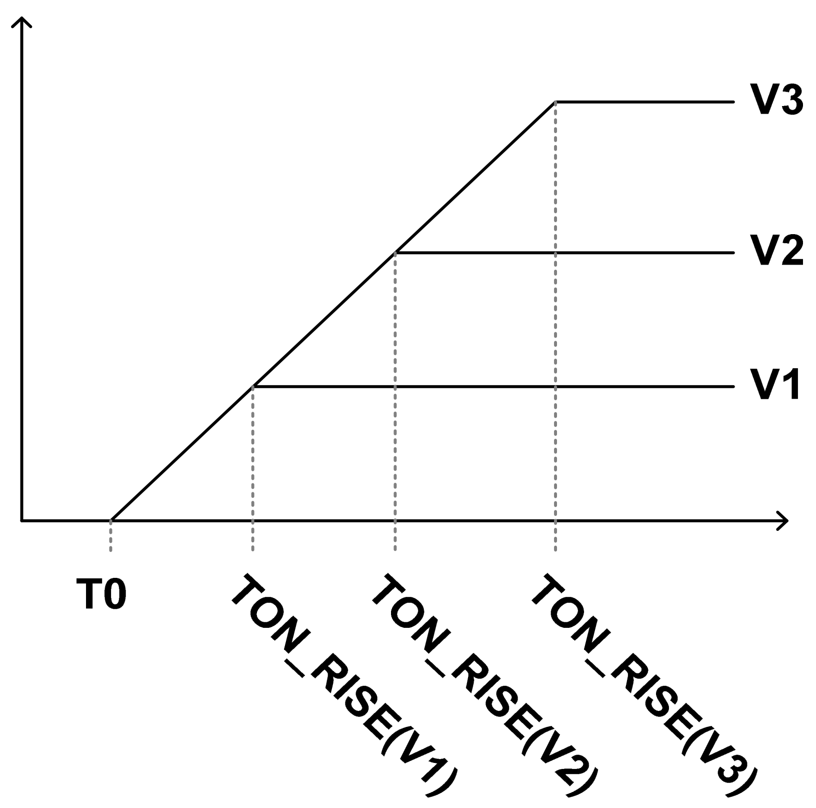


Sequencing: Time Driven Commands

- Open Loop: Does Not Require Power System Manager



Open Loop Tracking



- To Implement An Open Loop Tracking Turn On, Need To Know:
  - Each Output Voltage
  - Desired Rise Time (TON_RISE) For Just One Output Voltage
- Calculate TON_RISE Of All Other Outputs As Follows:

$$TON\_RISE(V2) = TON\_RISE(V1) \cdot \frac{V2}{V1}$$

$$TON\_RISE(V3) = TON\_RISE(V1) \cdot \frac{V3}{V1}$$

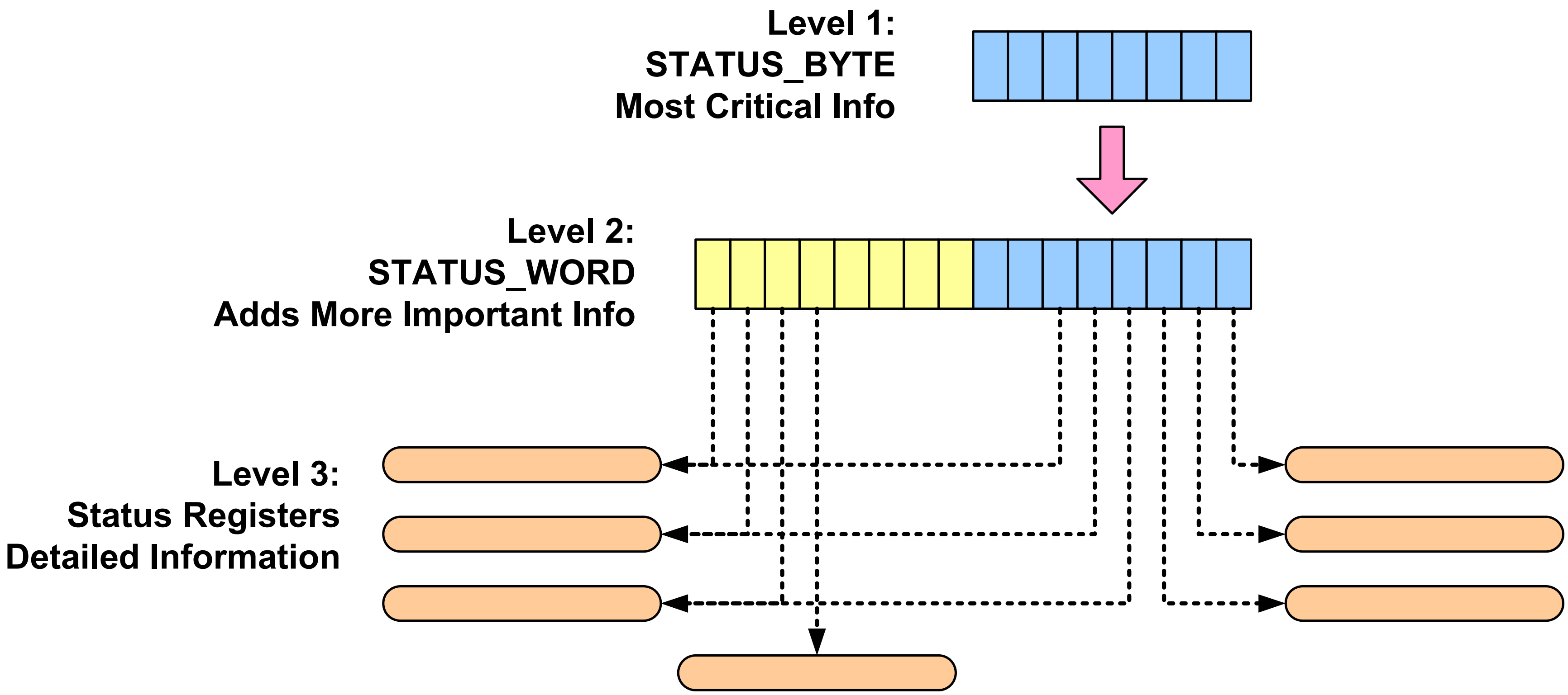
Status Reporting And Fault Management

- The PMBus Protocol Supports Two Alarm Levels
  – Warnings (Minor Alarms)
  – Faults (Major Alarms)
- Warnings Only Result In Host Being Notified That Attention Is Needed
- Faults Cause The PMBus Device To Respond And Take Action Internally As Programmed
- Parametric Information (e.g. Voltage) Can Also Be Read From PMBus Devices

Notifying The Host Of A Fault

- Host Can Continuously Poll PMBus Devices
- PMBus Device Can Send An Interrupt
  - SMBALERT# Signal Is Optional
  - See The SMBus Specification For Details
- PMBus Device Can Become A Bus Master And Transmit Notice To System Host
  - Optional
  - Requires A More Sophisticated Host And More Sophisticated PMBus Devices

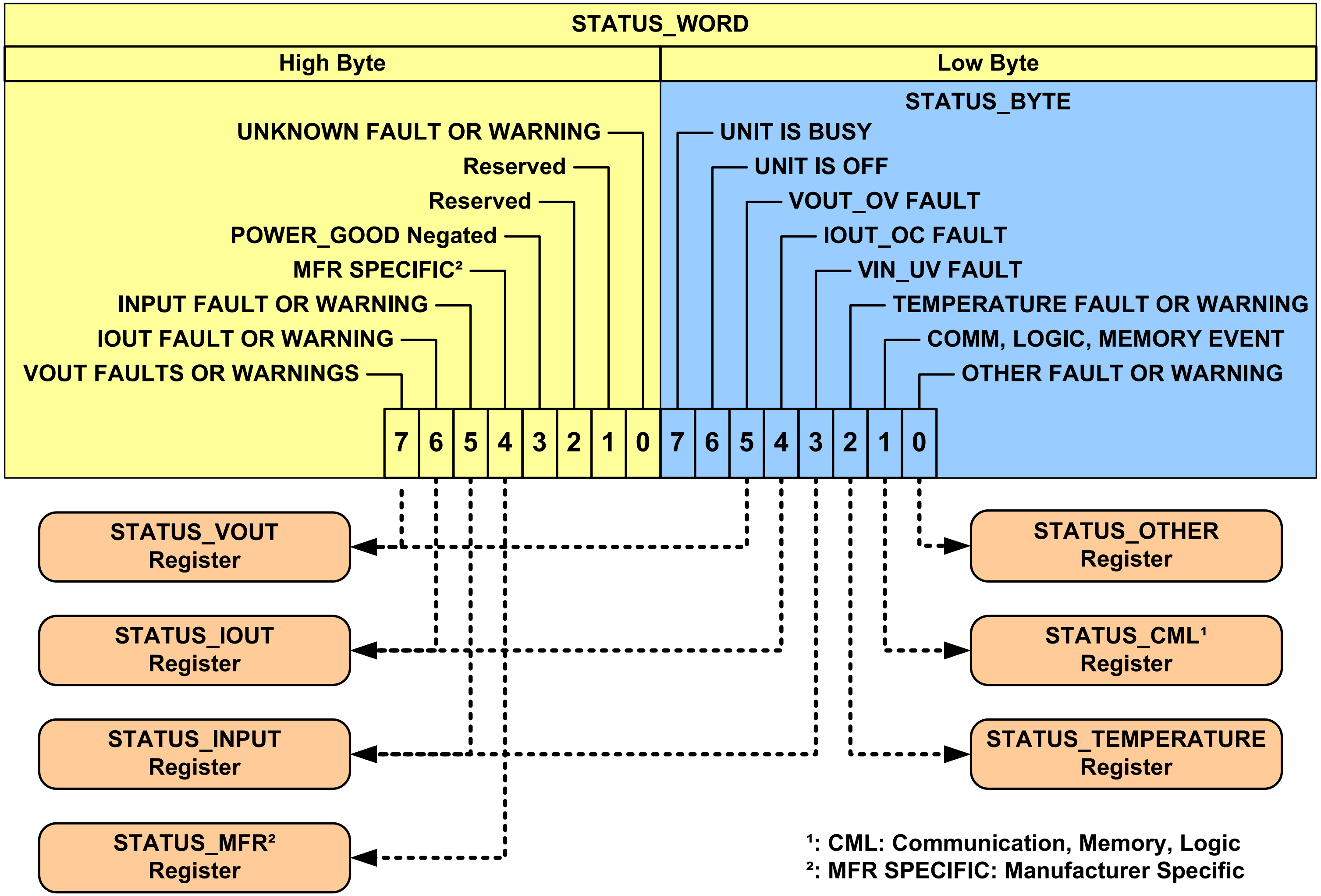
Status Reporting: 3 Levels Of Detail



STATUS_BYTE & STATUS_WORD

STATUS_WORD				
High Byte		Low Byte (STATUS_BYTE)		
Bit	Meaning	Bit	Meaning
7	VOUT FAULTS OR WARNINGS	7	UNIT IS BUSY
6	IOUT FAULT OR WARNING	6	UNIT IS OFF
5	INPUT FAULT OR WARNING	5	VOUT_OV FAULT
4	MFR SPECIFIC[2]	4	IOUT_OC FAULT
3	POWER_GOOD Negated	3	VIN_UV FAULT
2	Reserved	2	TEMPERATURE FAULT OR WARNING
1	Reserved	1	COMM, LOGIC, MEMORY EVENT
0	UNKNOWN FAULT OR WARNING	0	OTHER FAULT OR WARNING

Status Registers

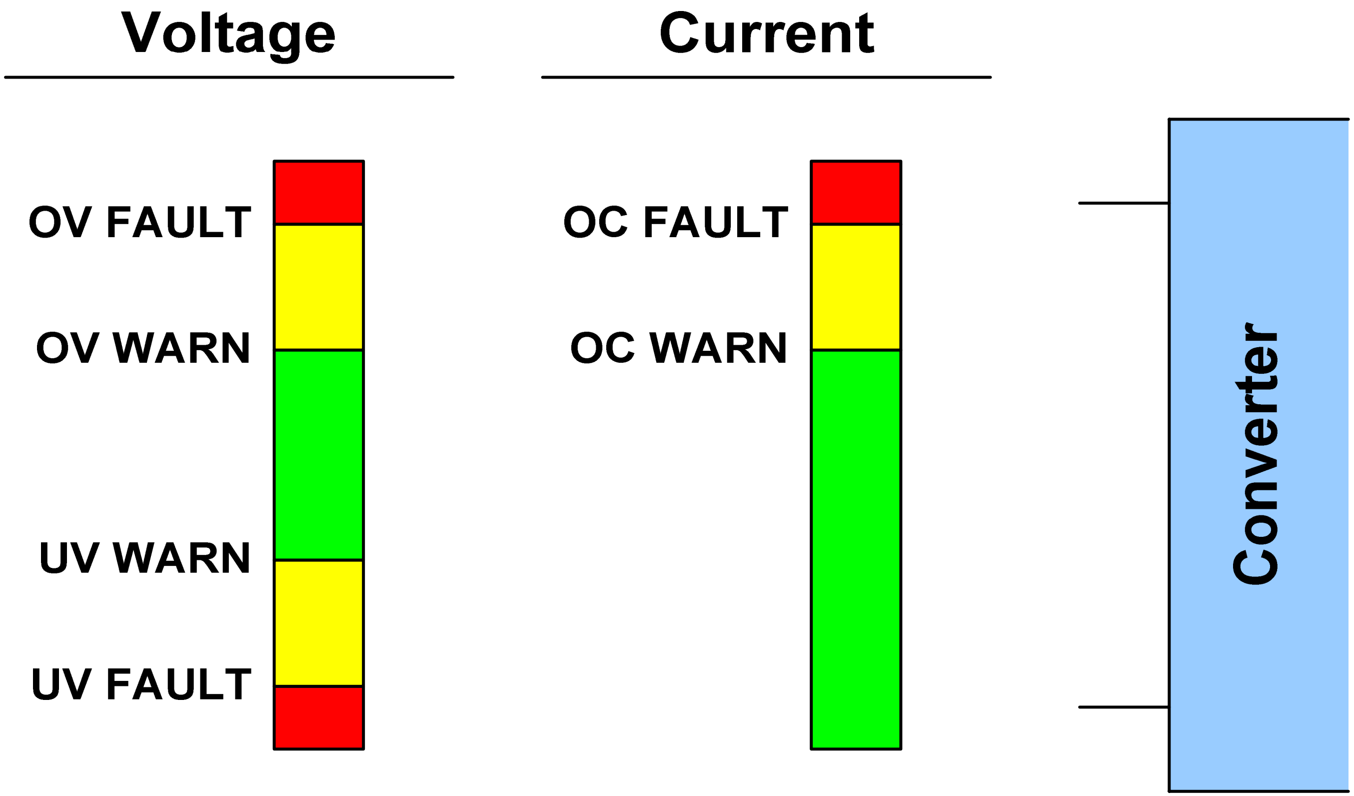


[1]: CML: Communication, Memory, Logic
[2]: MFR SPECIFIC: Manufacturer Specific

Clearing Status Bits

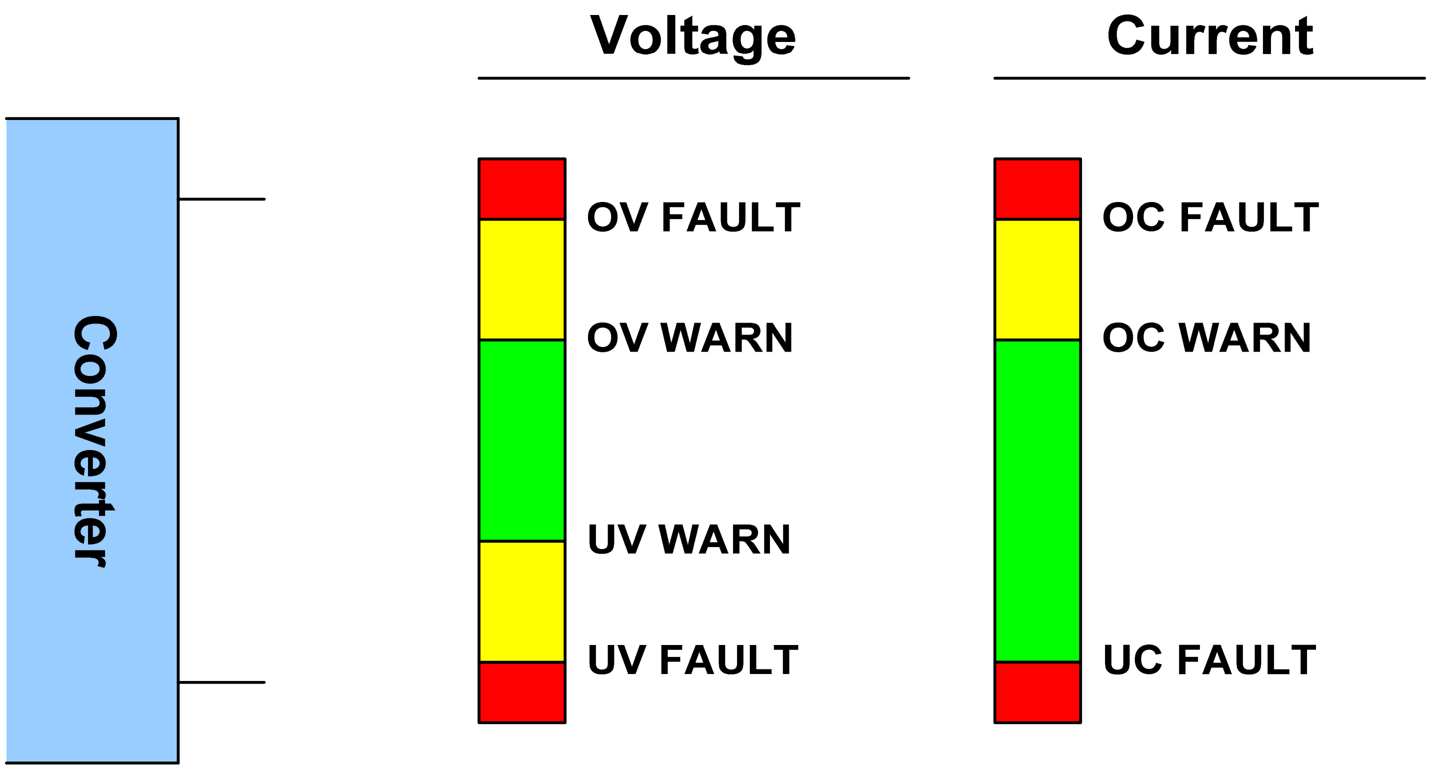
- Any warning or fault bits set in the status registers remain set, even if the fault or warning condition is removed or corrected, until:
  - The device receives a CLEAR_FAULTS command,
  - A RESET signal (if one exists) is asserted,
  - The output is commanded through the CONTROL pin, the OPERATION command, or the combined action of the CONTROL pin and OPERATION command, to turn off and then to turn back on
  - Bias power is removed from the PMBus device.
- If the warning or fault condition is present when the bit is cleared, the bit is immediately set again. The device shall respond as described in Section 10.2.1 or Section 10.2.2 as appropriate.

Fault Management: Input



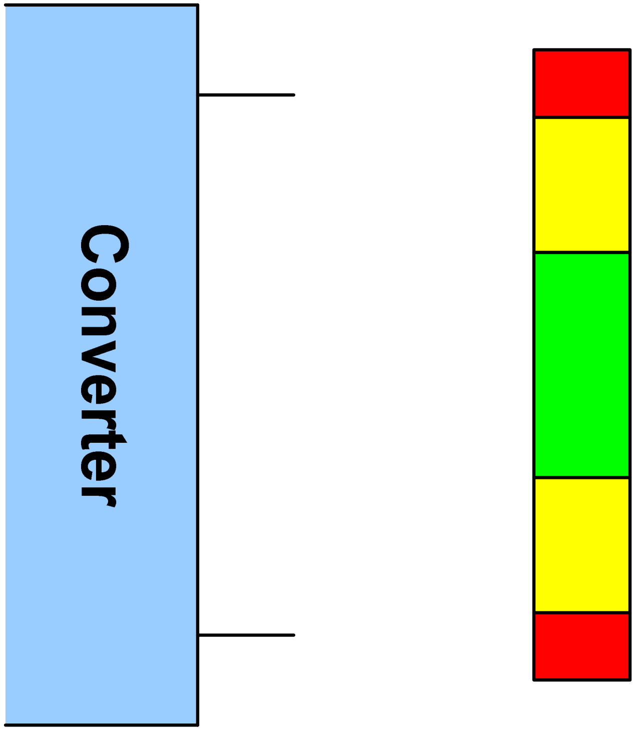
Related Commands: VIN_ON, VIN_OFF

Fault Management: Output



Related Commands:
POWER_GOOD_ON, POWER_GOOD_OFF

Other Fault Management



Temperature

OT FAULT

OT WARN

UT WARN

UT FAULT

Other Faults

Fan Fault 1

Fan Fault 2

Current Share

Power Limiting

Communication

And more...

Voltage Or Temperature Fault Response Programming Byte

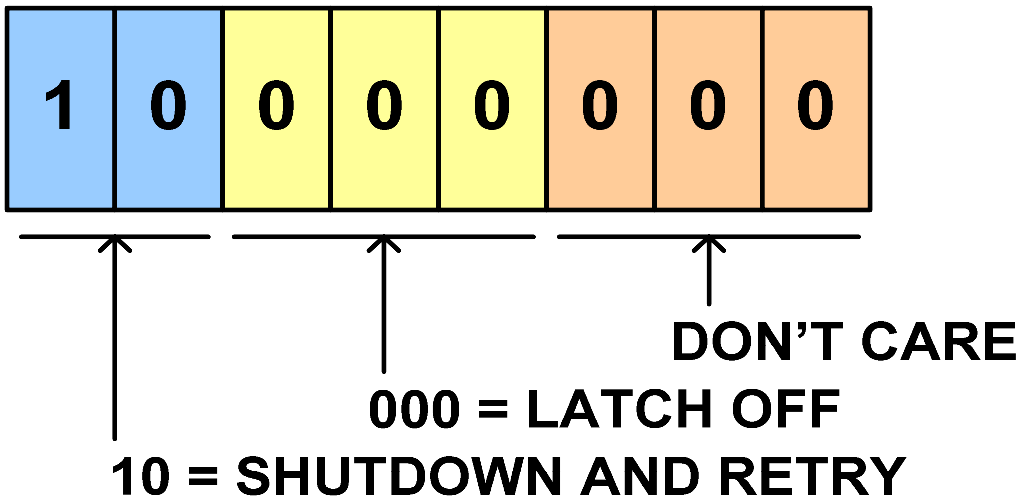
RESPONSE
- 00 - CONTINUE
- 01 - DELAYED OFF
- 10 - SHUTDOWN & RETRY
- 11 - INHIBIT

DELAY TIME
- XXX - NUMBER OF DELAY TIME UNITS

7	6	5	4	3	2	1	0
RESPONSE	RESPONSE	RETRY	RETRY	RETRY	DELAY TIME	DELAY TIME	DELAY TIME

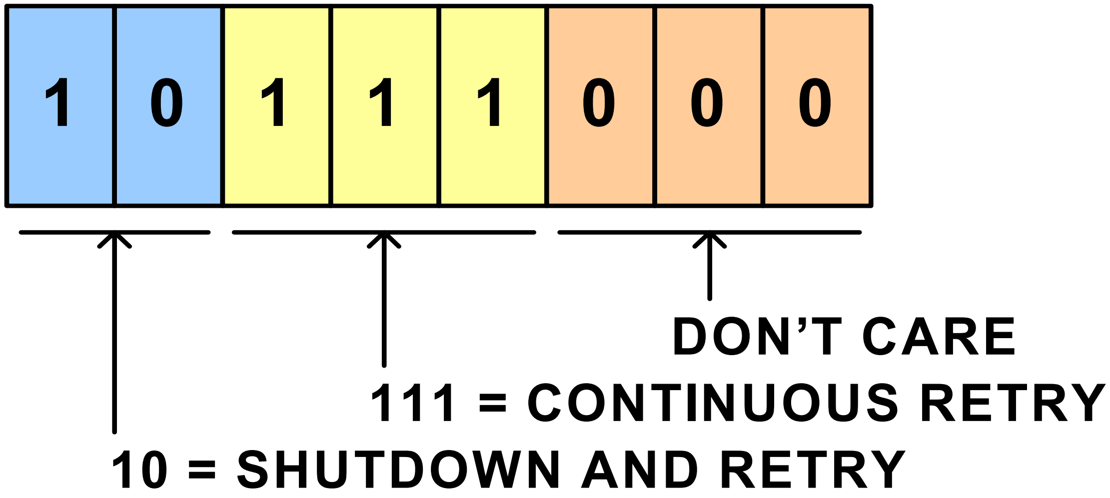
RETRY
- 000 - LATCH OFF
- 001 - 110: RETRY COUNT
- 111 - CONTINUOUS

Fault Response Examples



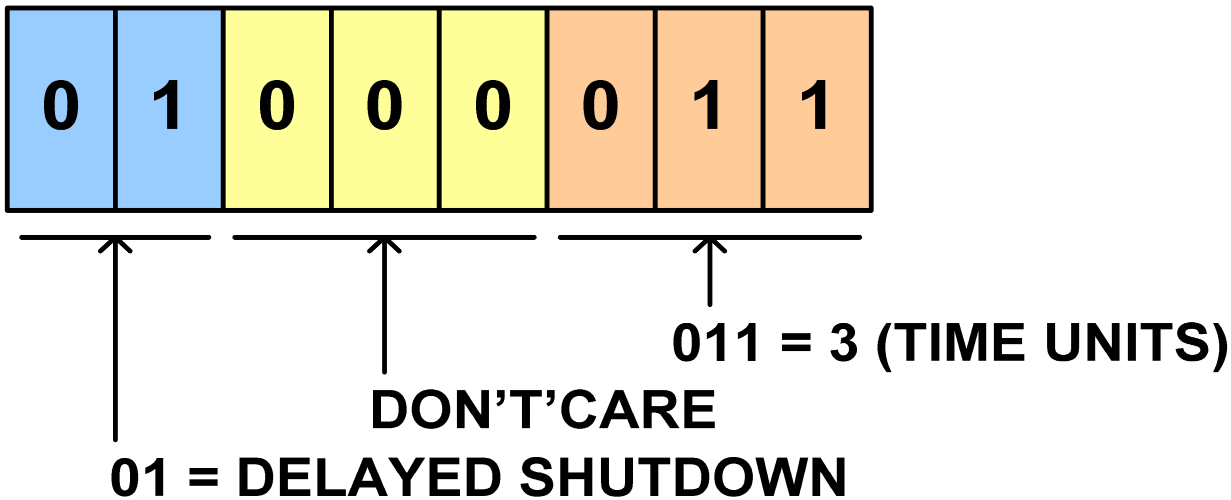
Shut Down And Latch Off

Fault Response Examples



Continuous Hiccup Mode

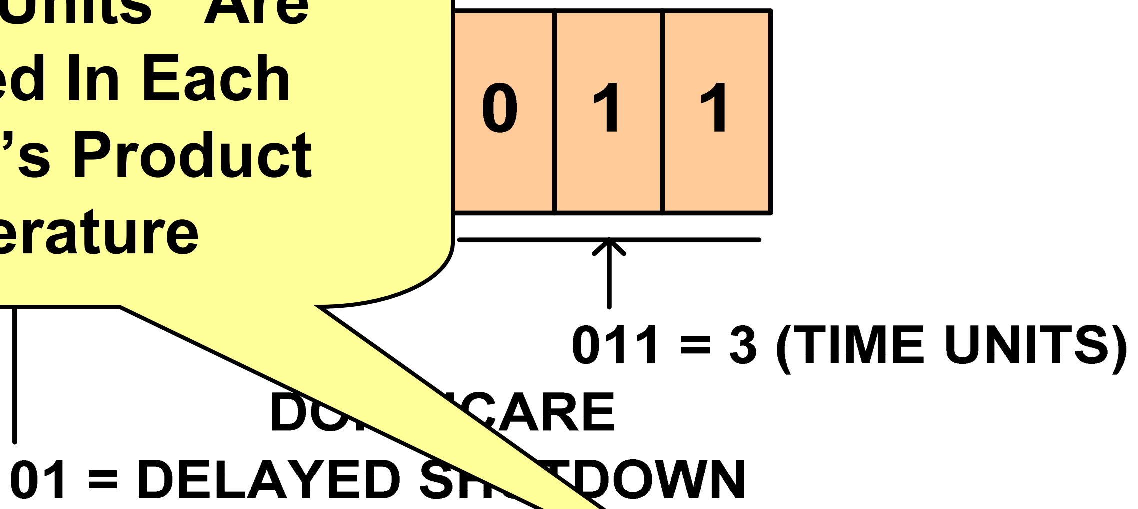
Fault Response Examples



Keep Operating For 3 Time Units.
If Fault Still Exists At That Time,
Shut Down And Latch Off

Fault Response Examples

"Time Units" Are Defined In Each Device's Product Literature



Keep Operating For 3 Time Units.
If Fault Still Exists At That Time,
Shut Down And Latch Off

Current Fault Options

RESPONSE

00 - CONTINUE
01 - CONTINUE WITH
LOW VOLTAGE SHUTDOWN
10 - DELAYED OFF
11 - SHUTDOWN & RETRY

DELAY TIME

XXX - NUMBER OF DELAY TIME UNITS

7	6	5	4	3	2	1	0

RETRY

000 - LATCH OFF
001 - 110: RETRY COUNT
111 - CONTINUOUS

Parametric Information

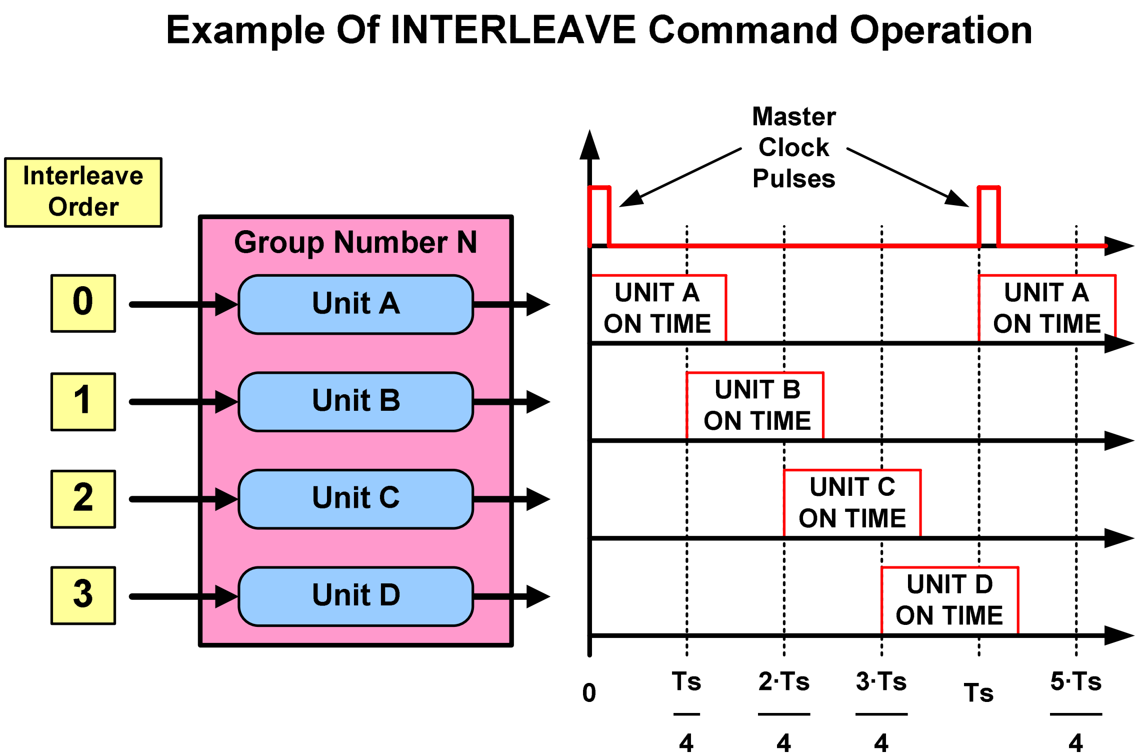
- Input Voltage (READ_VIN)
- Input Current (READ_IIN)
- Output Voltage (READ_VOUT)
- Output Current (READ_IOUT)
- Hold Up Capacitor Voltage (READ_VCAP)
- Temperature (READ_TEMPERTURE_1, _2, _3)
  - Up To 3 Sensors
- Fan Speed (READ_VFAN_1,_2)
  - Up To 2 Fans
- Duty Cycle (READ_DUTY_CYCLE)
- Switching Frequency (READ_FREQUENCY)

Group Commands/Operation

- Used When Multiple Units Need To Execute A Command Simultaneously
- One SMBus Transaction Used To Send Commands To Multiple Addresses
  - Sent In One Large Packet Using Repeated STARTs
- Can Be Same Or Different Commands
  - Example: Command One Unit To Margin Low And All Others To Margin High
- Commands Are Executed When SMBus STOP Condition Received

Interleaving

- INTERLEAVE Command Sets
  – Group Number
  – Number Of Units In The Group
  – Switching Order Within The Group

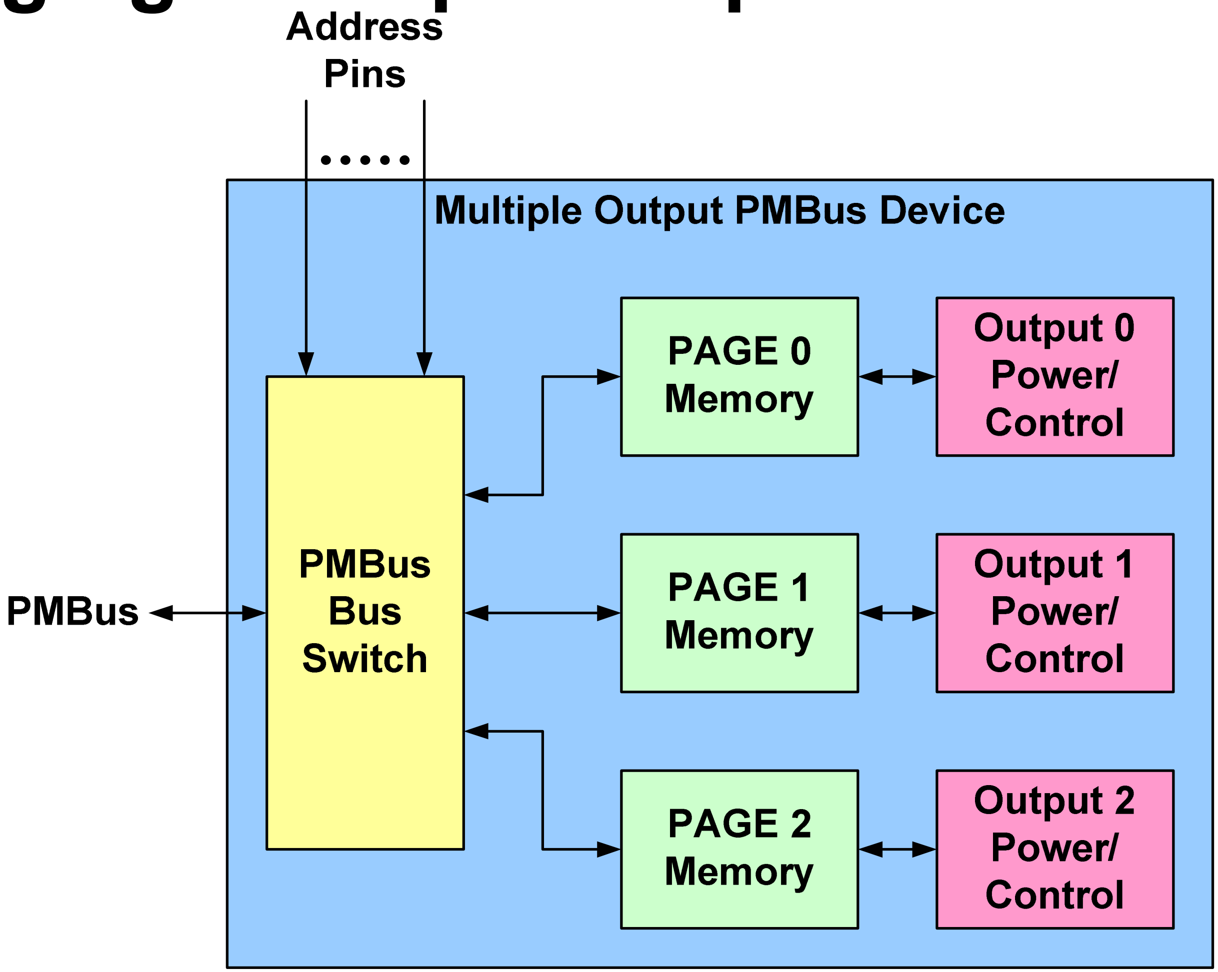


$$Tdelay(Unit\ X) = \frac{Interleave\ Order\ Of\ Unit\ X}{Number\ In\ Group} \bullet T_S$$

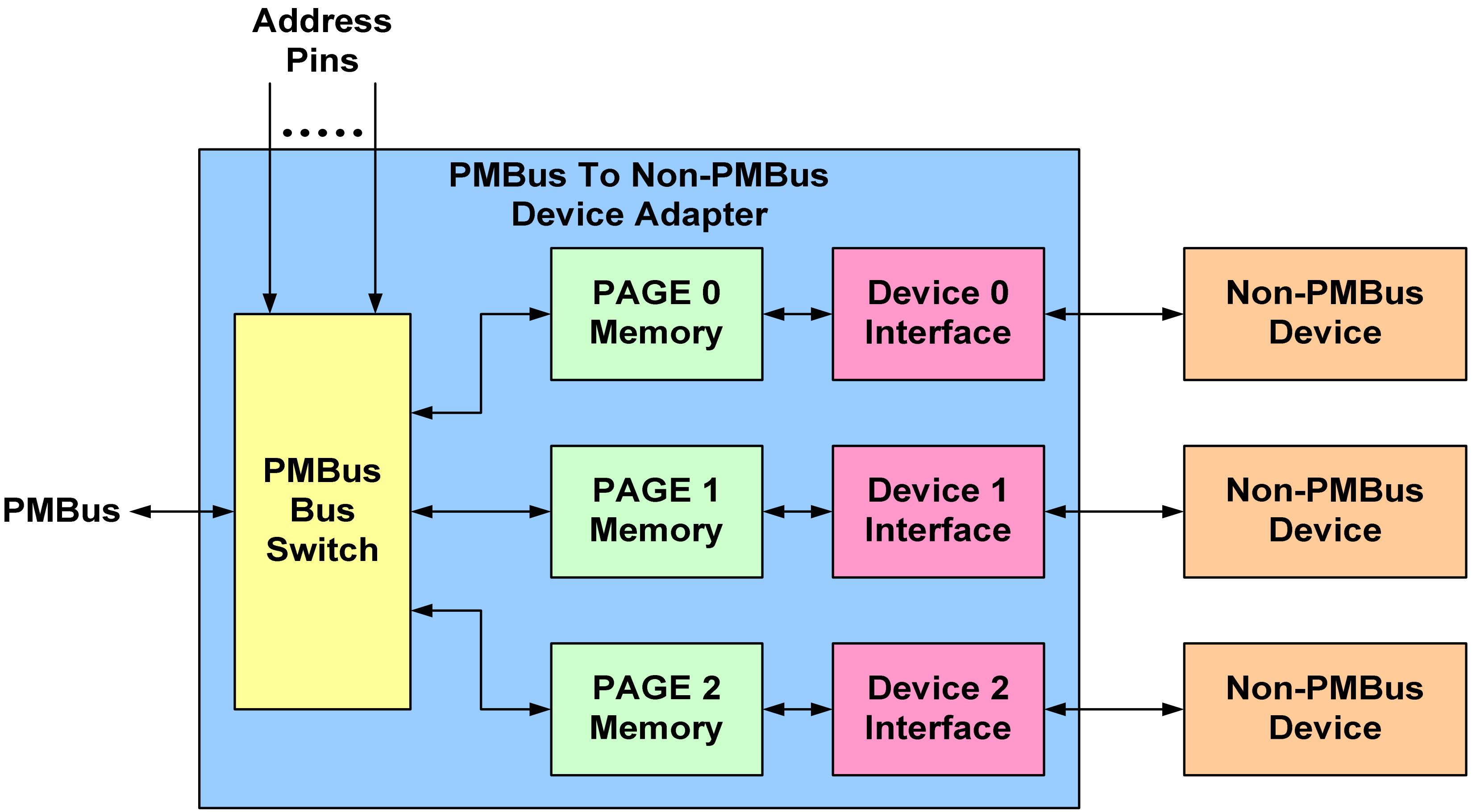
Multiple Output Units And Paging

- Paging Allows One Physical Address To Be Used To Control Multiple Outputs
  - One Address Per Physical Unit
  - One Page Per Output
  - Pages Contain All The Settings Of Each Output
- Paging Process
  - Set Page For Output Of Interest
  - Send Commands
    - Configure, Control, Read Status

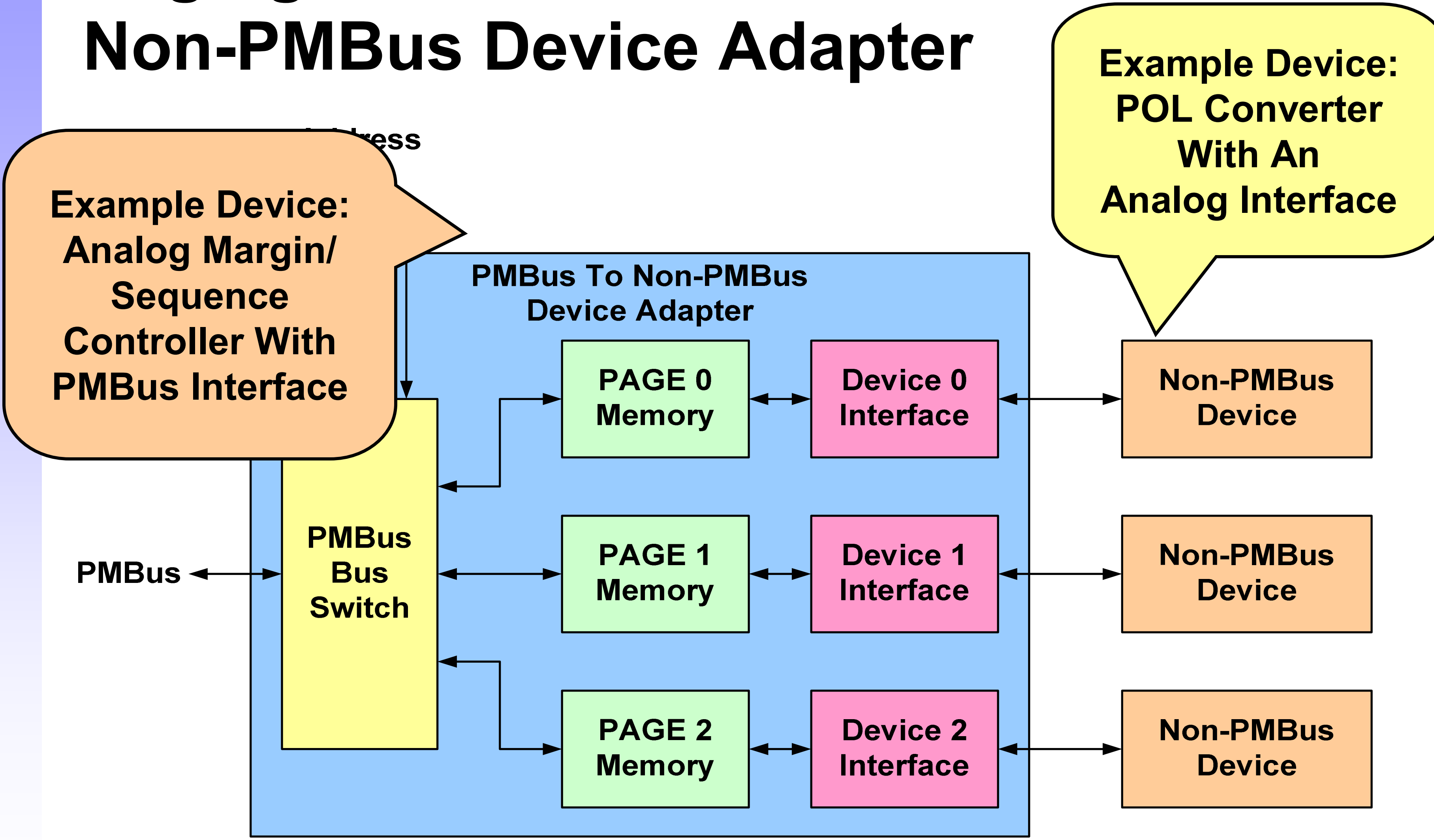
Paging: Multiple Output Units



Paging:
Non-PMBus Device Adapter



Paging: Non-PMBus Device Adapter



Data Integrity And Security

- Protecting Against Corrupted Transmissions
  - Packet Error Checking Can Be Used
- Unwanted Or Unintentional Data Changes
  - Write Protect Pin
  - WRITE_PROTECT Command

Manufacturer And User Data

- Manufacturer's Information
  - Inventory Information (Model Number, Etc.)
  - Ratings Information (Input Voltage Range, Etc.)
- User Data
  - 32 Command Codes For PMBus Device Makers To Support User Inventory And Configuration Data
  - Example: Digital Control Loop Coefficients
- Manufacturer Specific Commands
  - 45 Command Codes Reserved For PMBus Device Makers To Implement Manufacturer Specific Commands

Many Other Configuration Commands

- Maximum Output Voltage
- Maximum Output Power
- Voltage Scale For External Divider Network
- Maximum Duty Cycle
- Switching Frequency
- Turn On/Off Levels For Input Voltage
- Current Scale For Current Sense Resistance
- Current Measurement Calibration

For More Information

www.PMBus.org

info@PMBus.org

Thank You For Your Time And Attention!