PMBus™ Basics and TI's Point-of-Load Solutions

George Lakkas
Product Marketing Manager
Texas Instruments

October 8, 2015

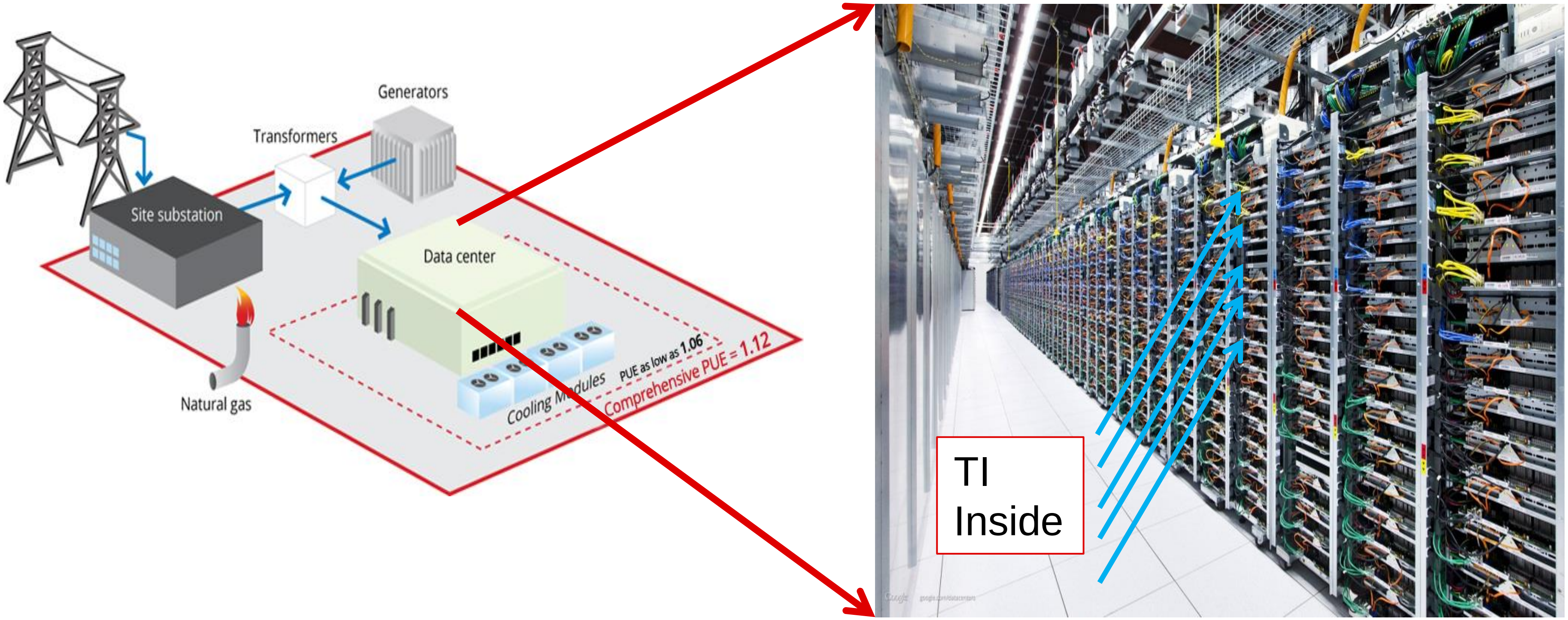
Texas Instruments

Agenda

- Active Power Management: Why?
- Introduction to PMBus™
- Adaptive Voltage Scaling (AVS), Power Monitoring and Sequencing via PMBus
- PMBus Point-of-Load (POL) Solutions

Active Power Management: Why?

Data center power usage effectiveness



In 2013, US data centers consumed 91 Billion kWh of electrical energy, translating to billions of dollars USD. Cloud Infrastructure equipment ODMs, OEMs, and Data Center owners have a huge incentive to understand their power usage, and eliminate waste.

Power usage effectiveness

$$PUE = \frac{Total\ Facility\ Energy}{IT\ Equipment\ Energy}$$

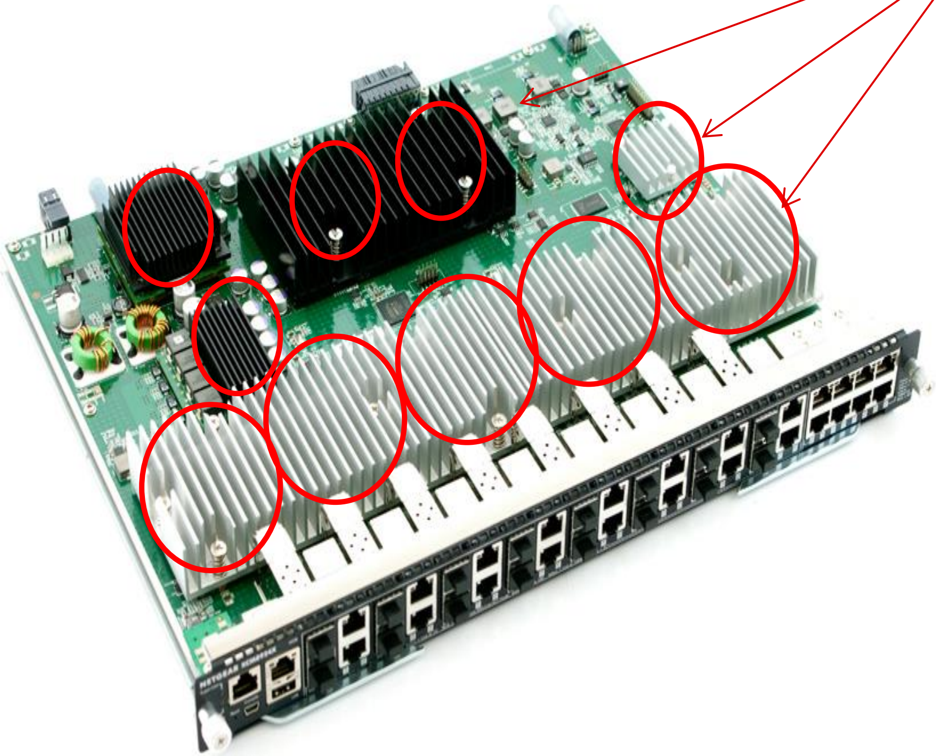
- **Power usage effectiveness** (**PUE**) is a measure of how efficiently a computer data center uses energy; specifically, how much energy is used by the computing equipment (in contrast to cooling and other overhead).
- PUE is the ratio of total amount of energy used by a computer data center facility to the energy delivered to computing equipment.
- Ideal PUE is 1.0.
- Companies are driving down their PUE to save money.
- As PUE decreases, IT infrastructure itself becomes a bigger portion of total datacenter cost. The only way to keep reducing cost is to decrease IT power waste.

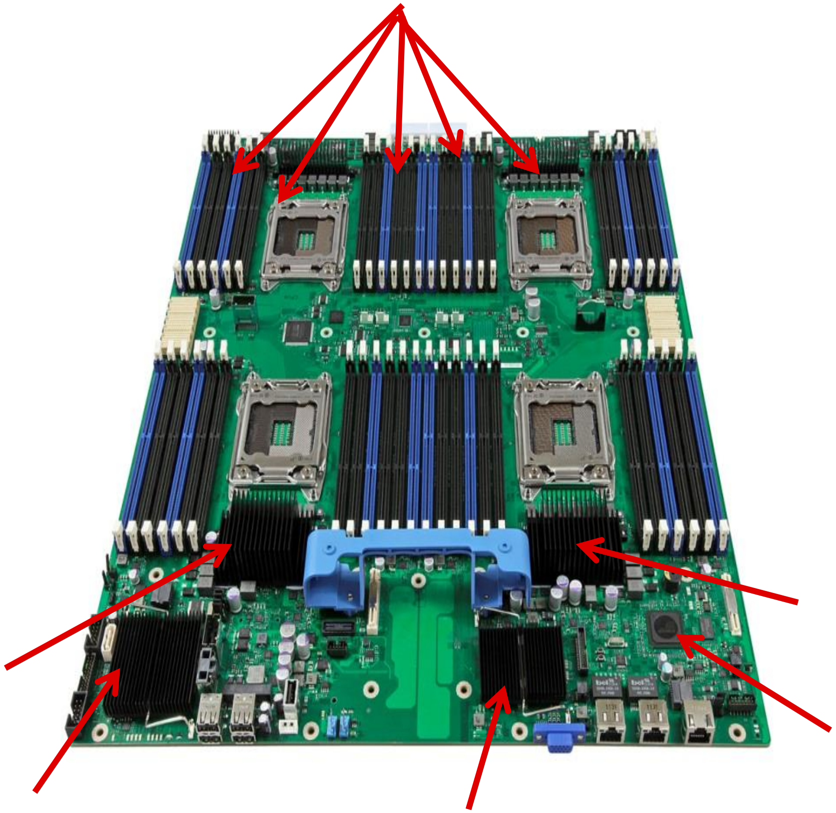
Example: Google boasts PUE of 1.12! → Servers/IT account for ~89% (1/1.12) of the energy use in a modern data center!

IT efficiency starts at CPU, ASIC, FPGA power

Enterprise Ethernet Switch

Mid-high current CPU, ASIC, FPGA devices. All need power!





Source:
http://www.storagereview.com/

Modern CPU, ASIC, FPGA power isn't trivial

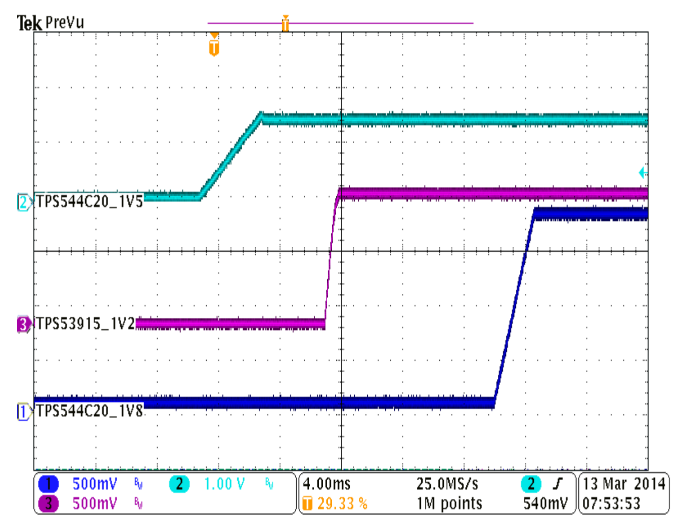
- Higher currents, faster transients, tighter tolerances
  - Example: 1V @ 35A, ±3% total tolerance (including DC, ripple, transient)

- Multiple power domains which require complex sequencing
  - Single board can have 30+ power rails which need to be properly sequenced for safety

Active power management through PMBus™: Less power usage, and better power control

- Active power management is most commonly implemented through PMBus.

- **PMBus can provide:**
  - Adaptive voltage scaling → Reduced power usage
  - Multiple rail control → Supply sequencing, re-configurability
  - Power supply monitoring → Board-level power use information, fault monitoring
  - Temperature information for load balancing and/or enhanced reliability



PMBus benefits in other applications

1. **Data collection.** Collect data in the hopes that later analysis will help them make better decisions in the future (power stage optimization, i.e)

2. **Live Performance Monitoring.** Able to provide a dashboard for real time diagnostics and optimization

3. **System Characterization.** Use telemetry in initial builds and possibly during ICT to help refine unknown parameters (like current levels and board temperatures) during test and qualification, but do not use it in live systems.

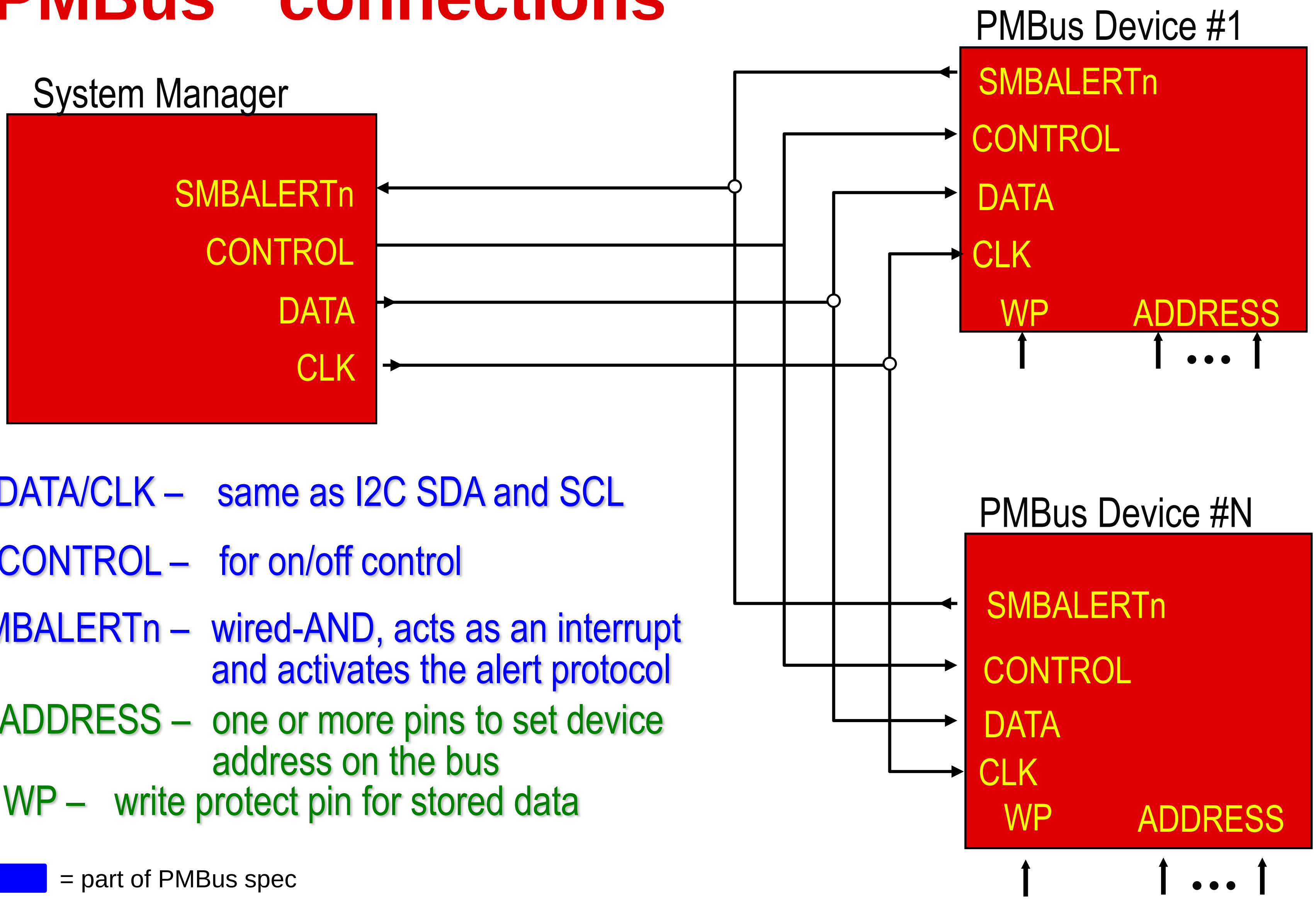
4. **Failure Analysis.** This is typically asking for "Black Box" or "Status Saving" - telemetry data to be available after a failure to analyze the failure to:
   1. Determine Cause
   2. Determine Failure Prediction Methods ("Killer App")

Introduction to PMBus™

What is PMBus™?

- PMBus is an I^2C-based communication standard for power supply management
- Owned and regulated by the System Management Interface Forum (SMIF)
  - SMIF membership is open to all
- Royalty free
- Specifications are freely available
- Works with all types of power management products
  - AC/DC power supplies
  - Isolated DC/DC converter and bus converter modules
  - Non-isolated point-of-load converters
  - Hot swap controllers and sub-system power monitors
  - Supply sequencers and POL voltage programmers
  - Monitors and fan controllers

PMBus™ connections



DATA/CLK – same as I2C SDA and SCL

CONTROL – for on/off control

SMBALERTn – wired-AND, acts as an interrupt and activates the alert protocol

ADDRESS – one or more pins to set device address on the bus

WP – write protect pin for stored data

= part of PMBus spec

= vendor specific

PMBus™ is an open standard

- Owned and regulated by The System Management Interface Forum (SMIF)
  - SMIF membership is open to all
- Royalty free
- Released specifications are freely available
- Works with all types of power management products:
  - AC/DC power supplies
  - Isolated DC/DC converter and bus converter modules
  - Non-isolated point-of-load (POL) converters
  - Hot swap controllers and sub-system power monitors
  - Supply sequencers and POL voltage programmers
  - Monitors and fan controllers

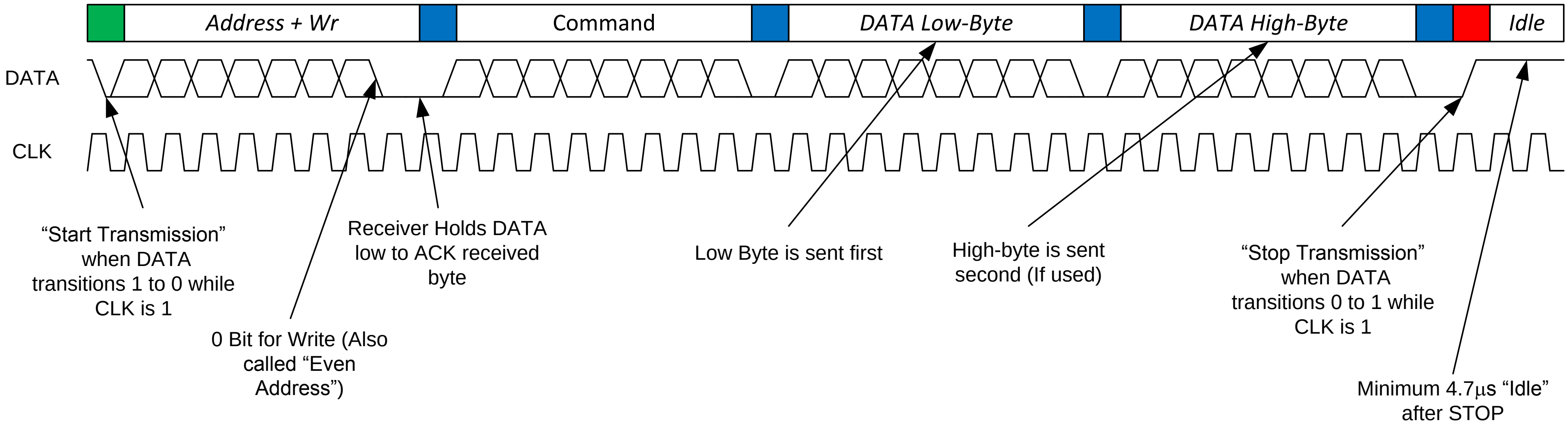
PMBus™ protocol and advantages

PMBUS

- Data Transport based on SMBUS v3.0 (PMBUS v1.3)
- Standard Command Codes tailored to Power Supply Applications
- Allows 400 kHz and 1MHz Clock (SMBUS 100 kHz)
- Added GROUP Command, Extended Command, Zone Read and Write Protocol

- Based on industry-standard SMBus hardware
  - Robust, alert-based monitoring, PEC support
- Standards-based command set
  - Comprehensive support for key functions related to configuration and control, monitoring, and fault management
- Platform convergence
  - Compatible with POL controllers for powering Intel VR12./VR13, Texas Instruments DSPs, and AMD CPUs
  - Compatible with other PMBus controllers for POL, isolated power, and system protection
- Future product compatibility
  - Additional PMBus-based products continue to be released with software compatibility

PMBus™ Write/Read Word



PMBus Write Word (2-byte) Transaction

Like I^{2}C, PMBus is a variable length packet of 8-bit data bytes each with their own receiver acknowledge, wrapped between a Start and Stop bit.

The first byte is always a 7-bit "slave address" followed by a 0 "write bit", sometimes called the "Even Address" that identifies the intended receiver of the packet.

The second byte is an 8-bit "Command" byte, identifying the PMBus command being transmitted using that command's Command Code.

After the command byte, the transmitter either sends data associated with the command to write to the receiver's command register, from lowest byte to highest byte, or sends a new start bit, indicating the desire to read the data associated with the command register from the receiver, after which, the receiver transmits the data following the same lowest-byte first format.

Basic PMBus™ requirements

PMBus devices…

- Must start up safely without bus communication
  - Some devices may need to be pre-programmed before start up
- Can be used with or without a power system manager or controller
- Supports “set and forget” mode
  - Can be programmed once at time of manufacture
  - Operates without bus communication
- Load default settings from either/or:
  - Hard-coded constants
  - Pin programming
  - Non-volatile memory

Adaptive Voltage Scaling (AVS), Power Monitoring and Sequencing via PMBus™

Adaptive voltage scaling – Why?
Reduce total energy use and maintain performance

Information Processing Efficiency

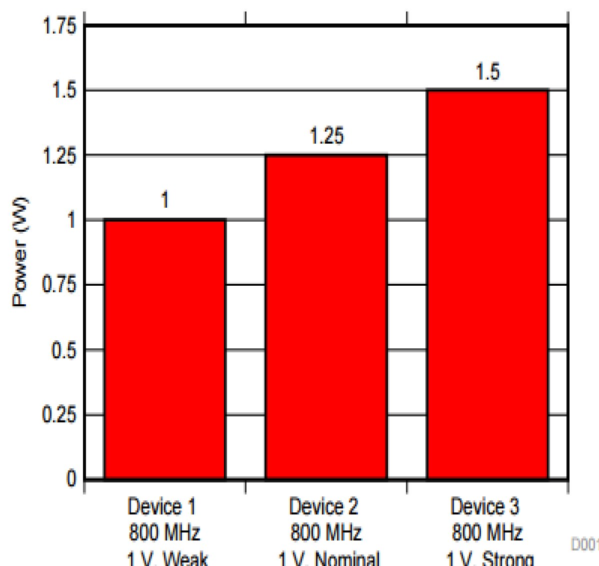
Power supply efficiency is important, but POL power loss is only a small portion of the system losses.

Typical POL efficiency might be 85-90%

E.g. 90% of the power consumption is from the load itself!

Process Variation

Part-to-part variation causes different ASIC, FPGA… devices to have performance for the same input voltage → Strong devices are burning more power than needed!



Dynamic Frequency Scaling

Many modern data processing devices are sufficiently self-aware to “know” when they don’t need full power to operate.

Example: Processor Frequency and input voltage can be reduced to save power when there is little data processing

Types of AVS

- **Static Adaptive Voltage Scaling – Optimizes for Process Variation and component tolerances**
  - **Class 0:** Output voltage is tuned on a per-board basis, at In-Circuit-Test or First-Power-on.
  - **Class 1:** Output voltage is tuned at each board power-on to account for aging effects
  - Static AVS also allows designers to trim out external component tolerances, like feedback divider tolerances
- **Dynamic Adaptive Voltage Scaling** – **Optimizes for traffic, frequency scaling, temperature**
  - **Class 2:** Software/firmware periodically adjusts VOUT on-the-fly based on sensed system conditions
  - **Class 3:** Closed loop hardware control adjusts VOUT on-the-fly

PMBus supports all 4 types of AVS!

AVS through trim and margin

$$V_{OUT} = V_{REF}\left(1 + \frac{R_{TOP}}{R_{BOT}}\right)$$

Parts:

- TPS40422
- TPS40425
- TPS544B20
- TPS544C20
- TPS53915

Margin None

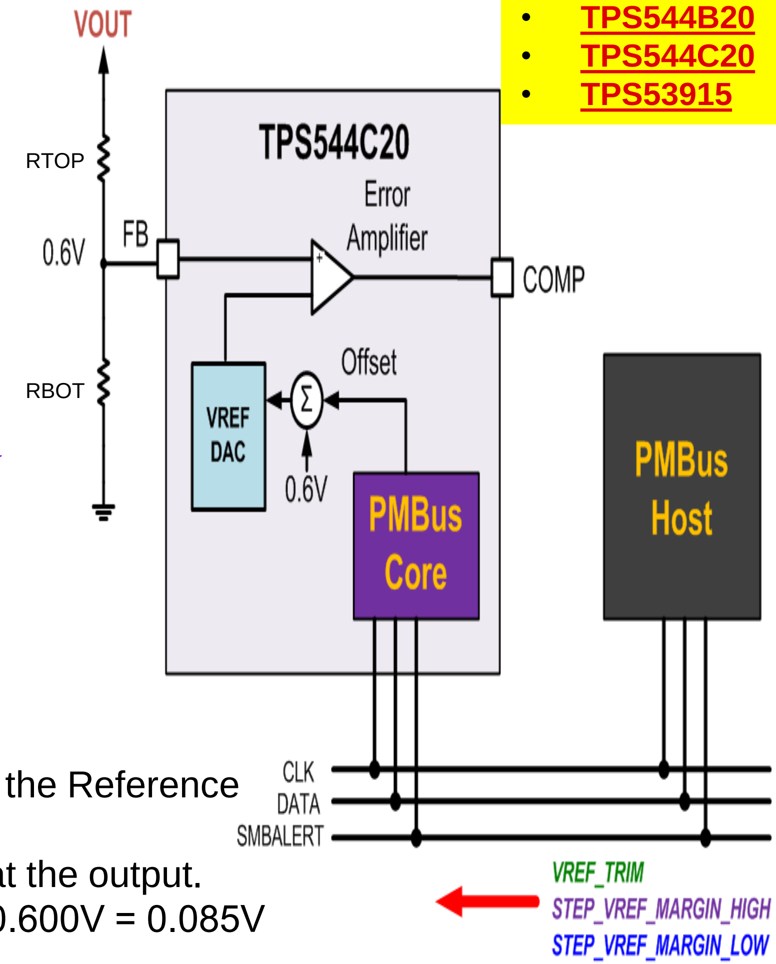
$$V_{REF} = 0.6V + VREF\_TRIM$$

Margin High

$$V_{REF} = 0.6V + VREF\_TRIM + STEP\_VREF\_MARGIN\_HIGH$$

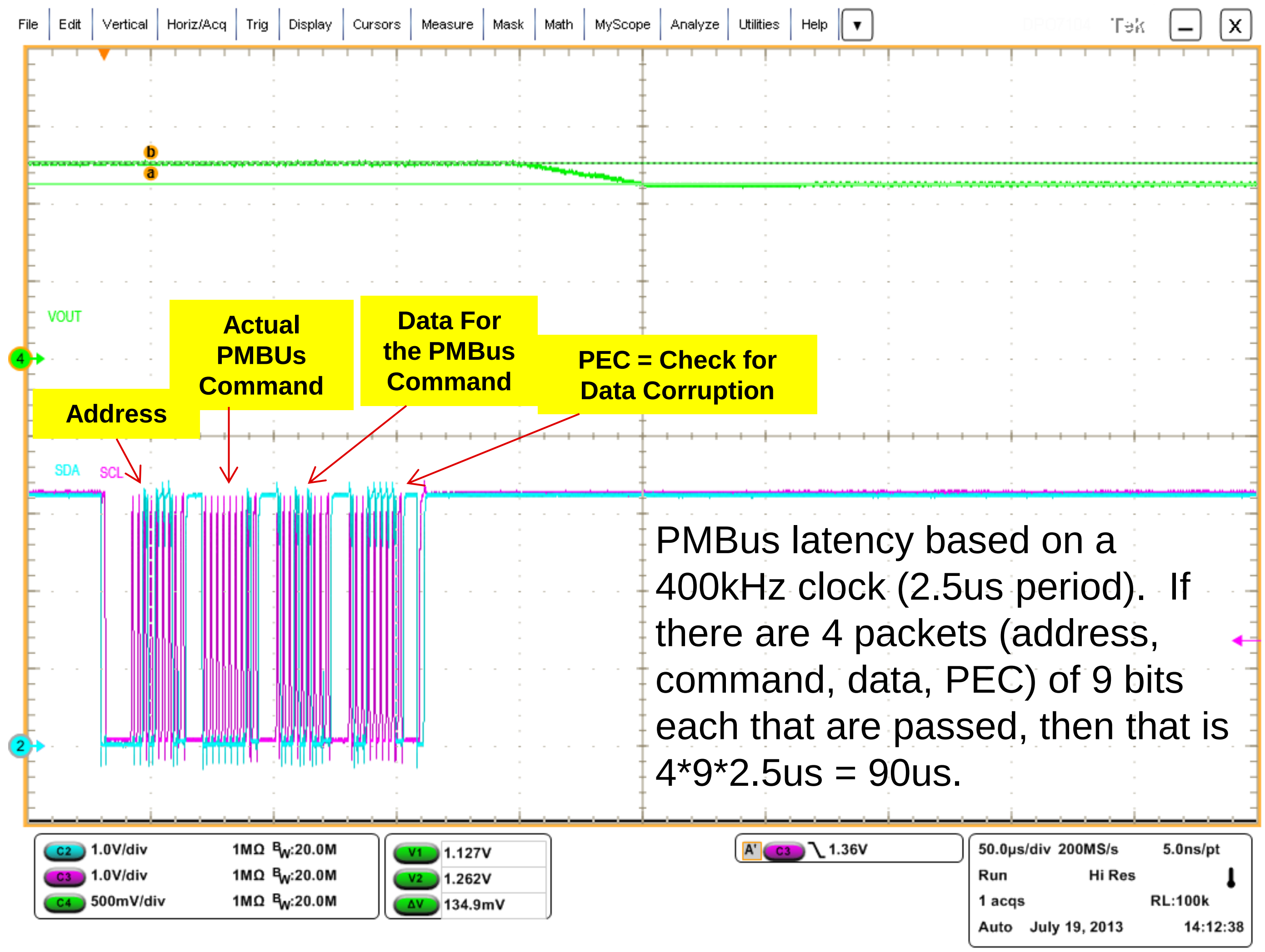
Margin Low

$$V_{REF} = 0.6V + VREF\_TRIM + STEP\_VREF\_MARGIN\_LOW$$

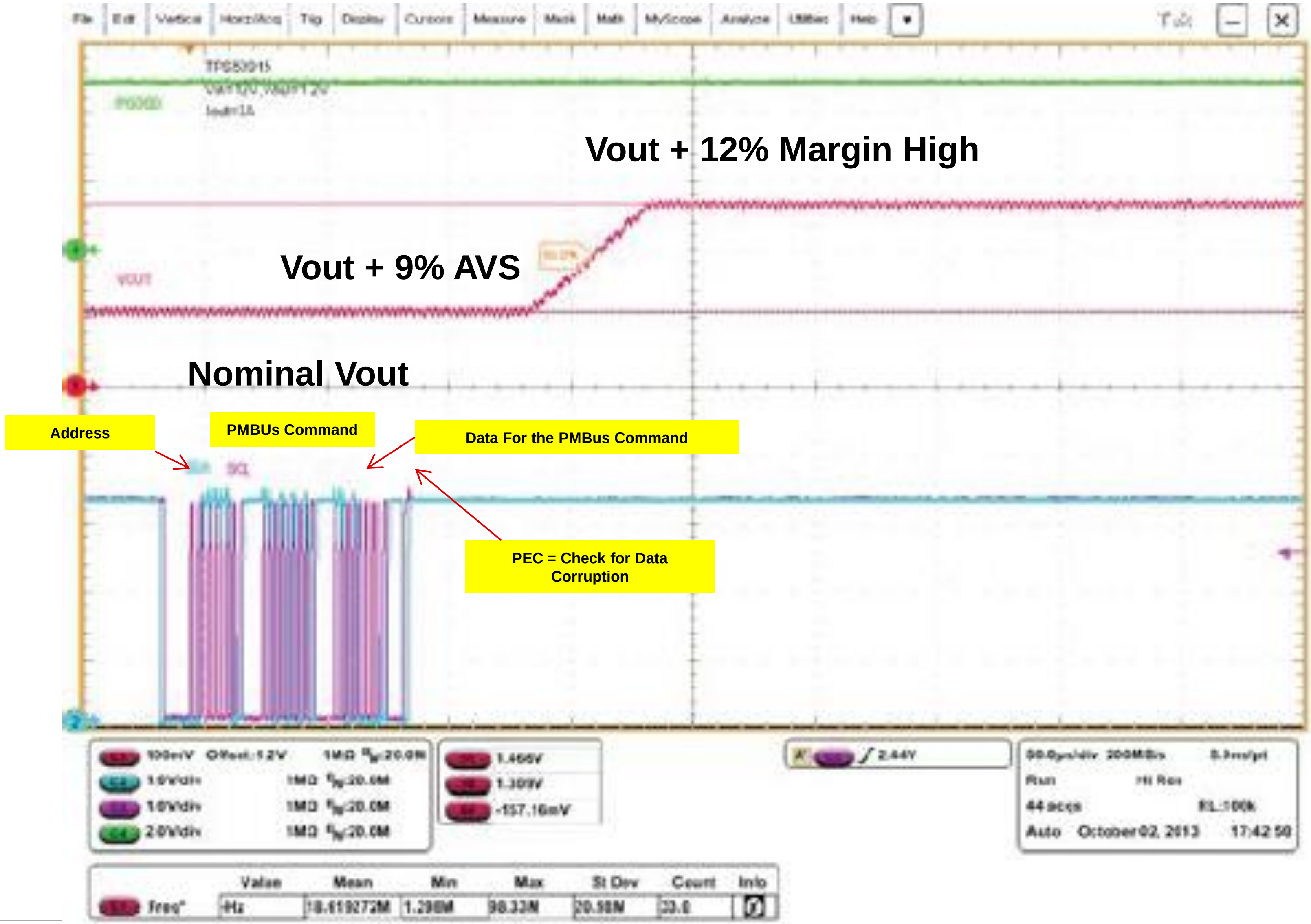


Both the VREF_TRIM and MARGIN_STEP functions are relative to the Reference Voltage.
You need to scale the values by the resistor divider to reflect them at the output.
For a 1V output a 0.051V “margin” should result in a 1V * 0.051V / 0.600V = 0.085V change in the output voltage.
If VREF_TRIM is set to +0.020V the output voltage should be 1.033V (1V + 0.033V from VREF_TRIM).
For a programmed 0.8V output voltage, this would give you a range of 0.640 – 0.880V, though this range can be moved by changing the resistor divider programmed voltage.

TI PMBus™ … VOUT margining through "VOUT_MARGIN" PMBus command



TPS53915 Vout adjustment via PMBus™ through "VOUT_ADJUSTMENT -> VREF TRIM" command



TEXAS INSTRUMENTS

AVS through VOUT_COMMAND

$$VOSL = VOUT\_SCALE\_LOOP$$

$$V_{OUT} = VOUT\_COMMAND = \frac{V_{REF}}{VOSL}$$

$$VOSL = \frac{R_{BOT}}{R_{BOT} + R_{TOP}}$$

PMBus core generates Vref:

$$V_{REF} = VOUT\_COMMAND \times VOSL$$

$$0.5V \leq V_{REF} \leq 1.5V$$ (DAC range)

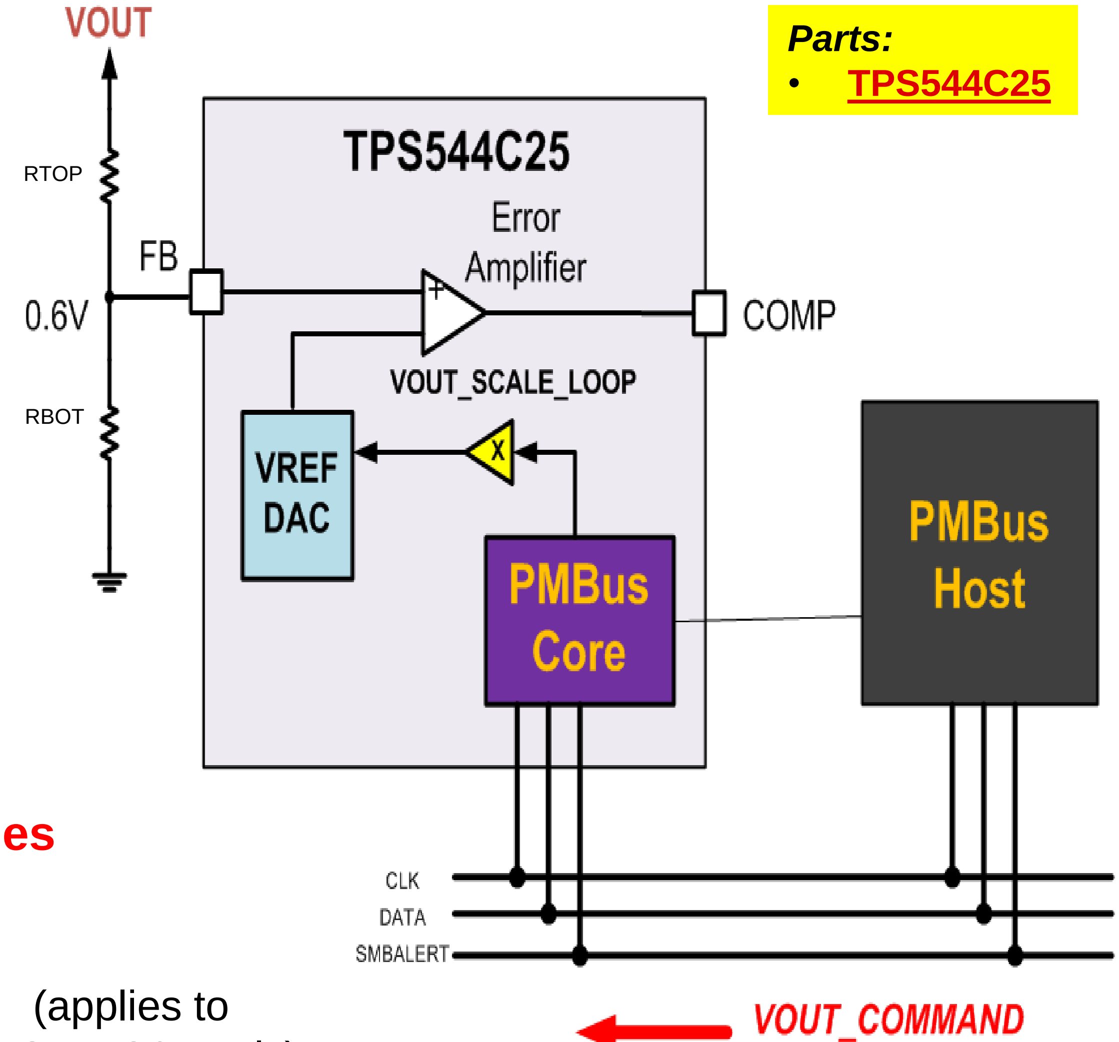
3 supported VOSL ➔ 3 V_{OUT} ranges

$$V_{OUT} = 0.5V - 1.5V\ (VOSL = 1)$$
$$V_{OUT} = 1V - 3V\ (VOSL = 0.5)$$
$$V_{OUT} = 2V - 6V\ (VOSL = 0.25)$$

(applies to TPS544C25 only)

Parts:

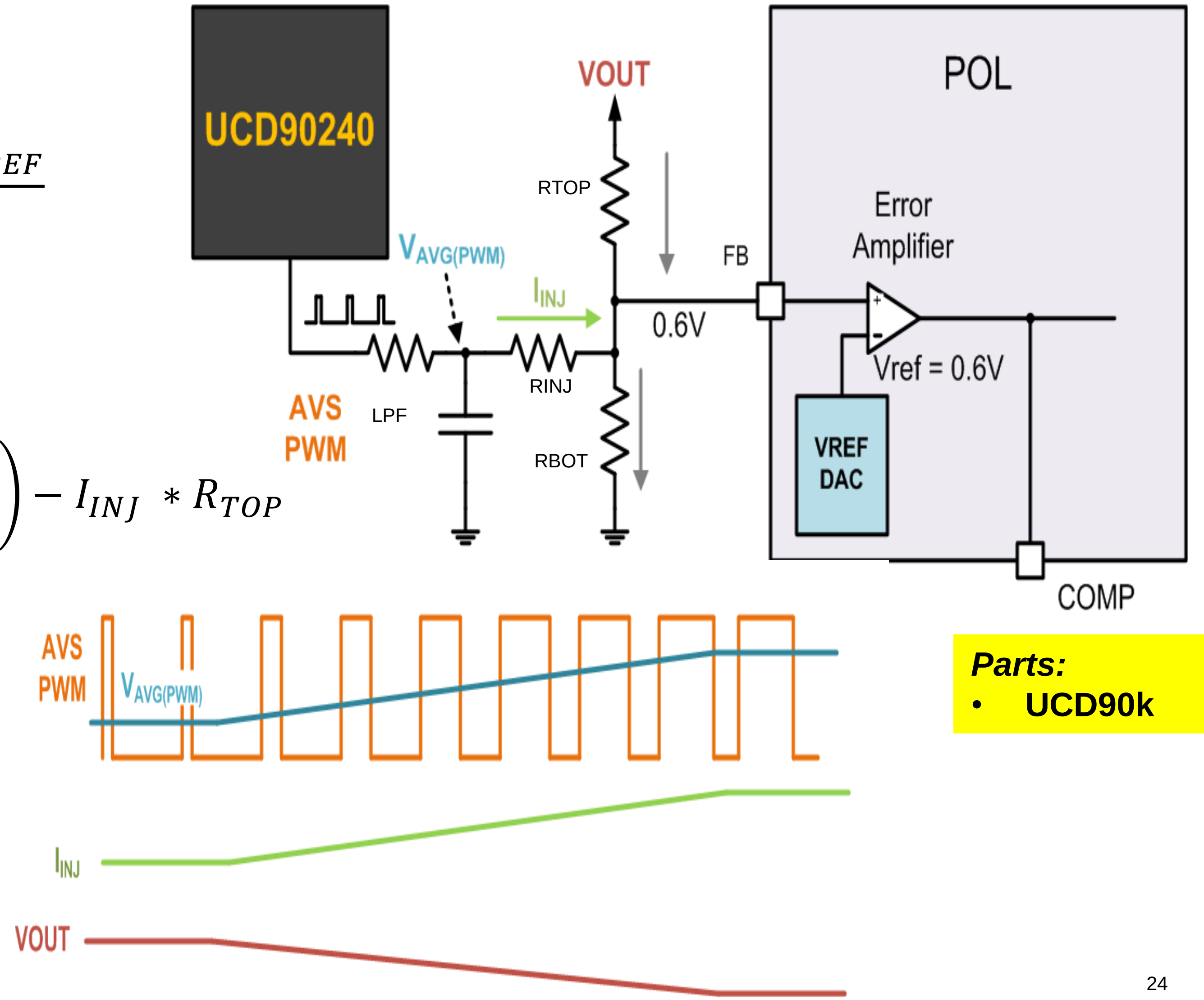
- TPS544C25



AVS through PWM output

$$I_{INJ} = \frac{V_{AVG(PWM)} - V_{REF}}{R_{INJ}}$$

$$V_{OUT} = V_{REF}\left(1 + \frac{R_{TOP}}{R_{BOT}}\right) - I_{INJ} * R_{TOP}$$

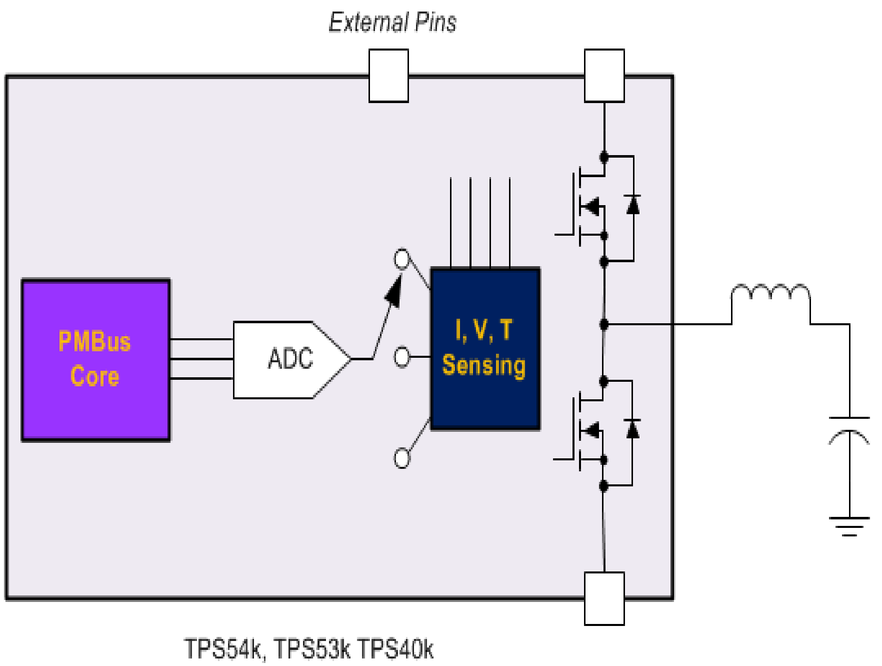


PWM Duty Ratio changes VOUT

Parts:
- **UCD90k**

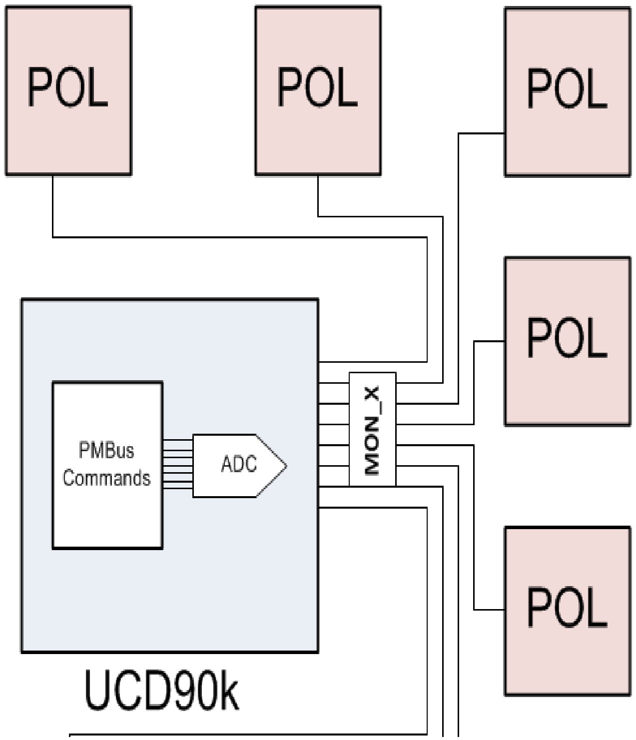
PMBus™ telemetry

TPS54k, TPS53k, TPS40k



ADC integrated into POL

UCD90k Series



ADC muxed to multiple POL

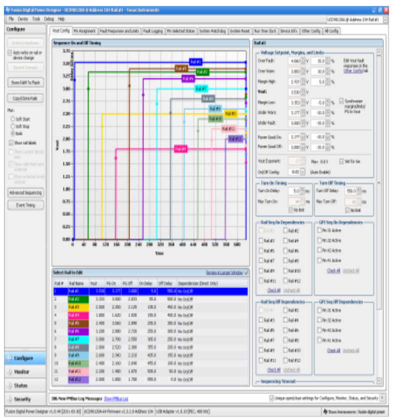
Monitoring: I, V, T telemetry through PMBus™

Fusion Digital Power Designer Software

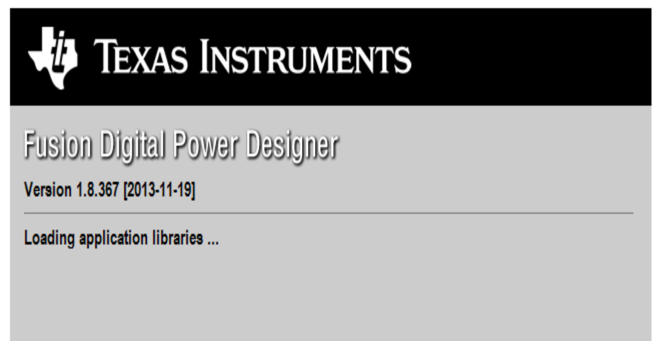
- READ_VOUT – Read the output voltage
- READ_IOUT – Read the DC output current
- READ_TEMPERATURE - Read the temperature via external sensor OR internally

Fusion Digital Power Designer Manufacturing GUI

1. Use Fusion configuration tool to select settings for rails on a board



2. Export settings to a file



3. Load configuration file into manufacturing tool

PMBus Programmer Script

- Device-neutral way to specify the programming steps required to write and verify NVM
- Equipment vendors only need to do programming/algorithm development one time to cover all TPS devices that will use this format
- Script flow:
    1. Verify correct part present via DEVICE_CODE PMBus command read
    2. Write configuration to volatile memory via PMBus
    3. Execute PMBus STORE_XXX_ALL command to save config to NVM
    4. Reset power to device
    5. Read back config via PMBus and verify
- Takes into account device timing requirements
- Text file in “comma separated value” (CSV) format

PMBus Programmer Script Example

TPS40422 dual-rail config (truncated)

Comment	**Verify correct device is present**	
ReadWord	0xFC	0x0007
Comment	**Write configuration**	
Comment	Write MFR_00 0000000000000000b	
WriteWord	0xD0	0x0000
Comment	Write MFR_21 (OPTIONS) EN_ADC_CNTL:1, CH1_DTC:0, CH1_DTC:0	
WriteWord	0xE5	0x0400
...		
Comment	Write ON_OFF_CONFIG [Rail #1] Mode: Always Converting	
WriteByte	0x00	0x00
ReadByte	0x00	0x00
WriteByte	0x02	0x02
Comment	Write IOUT_CAL_GAIN [Rail #1] 1.0681 mohm	
WriteWord	0x38	0x2388
...		
Comment	**Store configuration to data flash**	
Comment	Execute STORE_USER_ALL	
SendByte	0x15	
Pause	200	Pausing 200.00 ms for STORE_USER_ALL hold time

User scratch pad

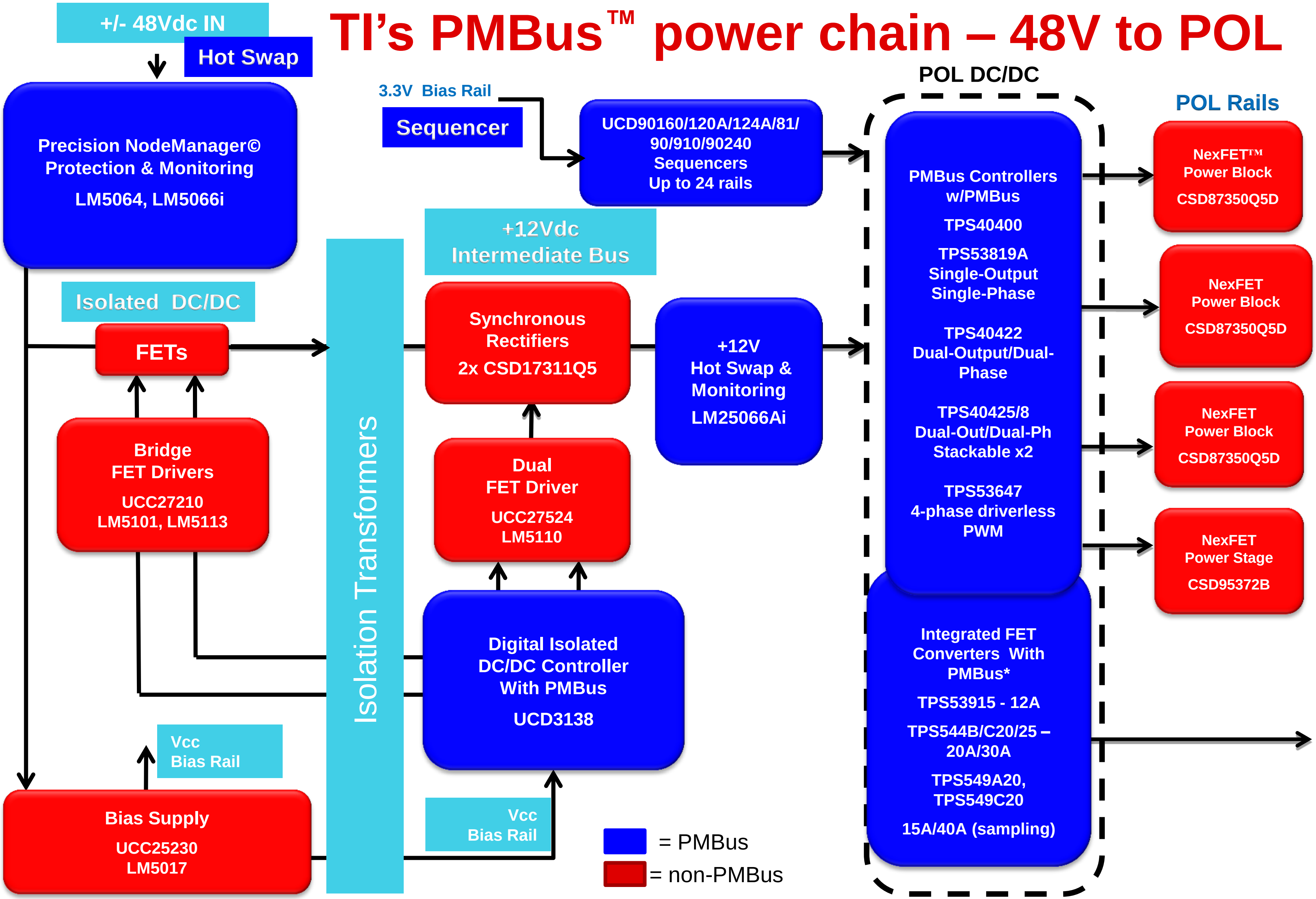
Enables ADC for Vout/Iout/Temp monitoring and increases dead time for Ch1/2 gate drivers

Configures the combination of CNTLx pins input and serial bus commands for on/off

Comment	**Reset power to device**	
Reset		
Comment	**Validate configuration**	
Comment	Validate MFR_00 0000000000000000b	
ReadWord	0xD0	0x0000
Comment	Validate MFR_21 (OPTIONS) EN_ADC_CNTL:1, CH1_DTC:0, CH1_DTC:0	
ReadWord	0xE5	0x0400
...		
Comment	Validate ON_OFF_CONFIG [Rail #1] Mode: Always Converting	
WriteByte	0x00	0x00
ReadByte	0x00	0x00
ReadByte	0x02	0x02
Comment	Validate IOUT_CAL_GAIN [Rail #1] 1.0681 mohm	
ReadWord	0x38	0x2388
...		
Comment	**Script end**	

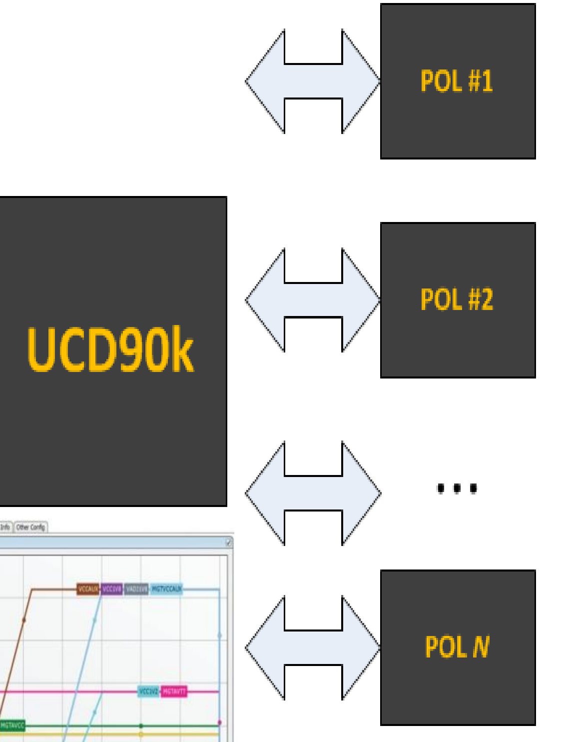
PMBus™ Point-of-Load (POL) Solutions

TI's PMBus™ power chain – 48V to POL



PMBus™ sequencers + system health monitors

Part #	Channels	Additional Features
UCD9080	8	
UCD9081	8	NVM error logging
UCD9090	10	
UCD90910	10	Integrated fan controller
UCD90124A	12	
UCD90120	12	ACPI support
UCD90240	24	NVM error logging

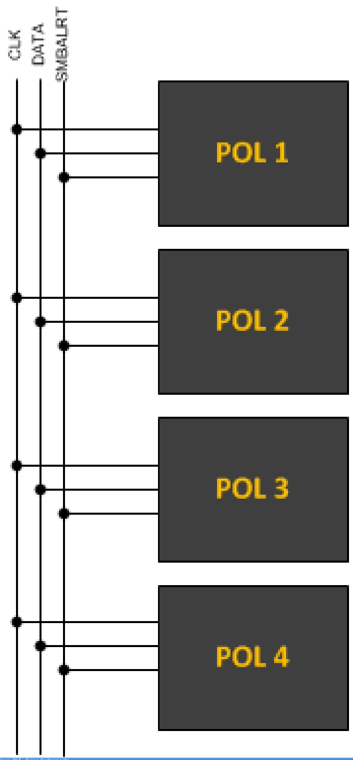


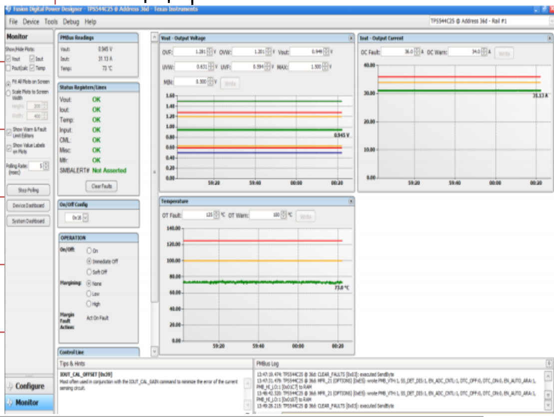
NEW!

Sequencers and system health monitors are fully compatible with analog POL's!

TI's PMBus™ POL solutions

Part Number	Type	IOUT/Phases	Control loop	Full/Lite PMBus
TPS544x20	iFET	20, 30A	DCAP2	Full
TPS544x25	iFET	20, 30A	Voltage mode	Full
TPS40422	Controller	2 Phases	Voltage Mode	Full
TPS40425/8	Controller	2 x 2 Phases (stack)	Voltage Mode	Full
TPS40400	Controller	1 Phase	Voltage Mode	Full
TPS53647	Controller	4 phases	DCAP+	Full
TPS53631/41/61	Controller	3/4/6 phases	DCAP+	Full (Intel VR12.5)
TPS53915	iFET	12A	DCAP3	Lite
TPS53819	Controller	1 Phase	DCAP2	Lite
UCD9222	Controller	2 Output / 2 Phase	Digital	Full
UCD9924	Controller	2 Output / 4 Phase	Digital	Full
UCD9244	Controller	4 Output / 4 Phase	Digital	Full
UCD9246	Controller	4 Output / 6 Phase	Digital	Full
UCD9248	Controller	4 Output / 8 Phase	Digital	Full





Thank you!

www.ti.com/pmbus