Topic 6

Using PMBus for Improved System-Level Power Management

Kurt Hesse

Agenda

- **PMBus overview**
- Some tasks and PMBus solutions
- Brief design example

Evolution

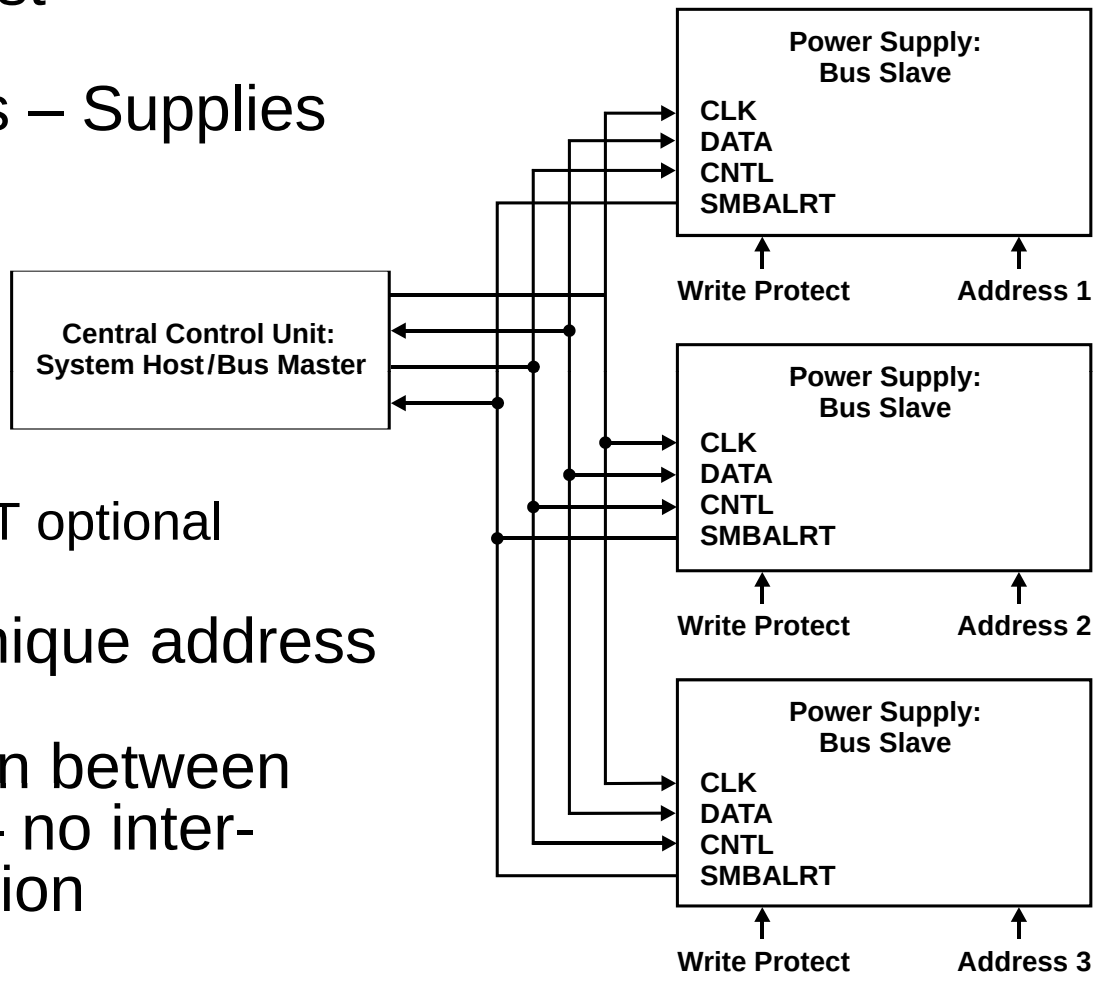
- Newer systems require more advanced capabilities
- Supplies evolving in:
  - Size and integration
  - Functionality
  - Flexibility
  - Monitoring and control
- PMBus is an answer

What is PMBus?

- Open standard
  - Serial interface to controller
    - Electrical specification protocol and command language
  - Anyone can use – royalty free
  - Controlled by Special Interest Group (SIG):
    - www.pmbus.org
- Communication only
  - No form factor
  - No I/O specification
  - No specific functionality requirement
  - No pinout
- Implementation of the physical supply is completely open

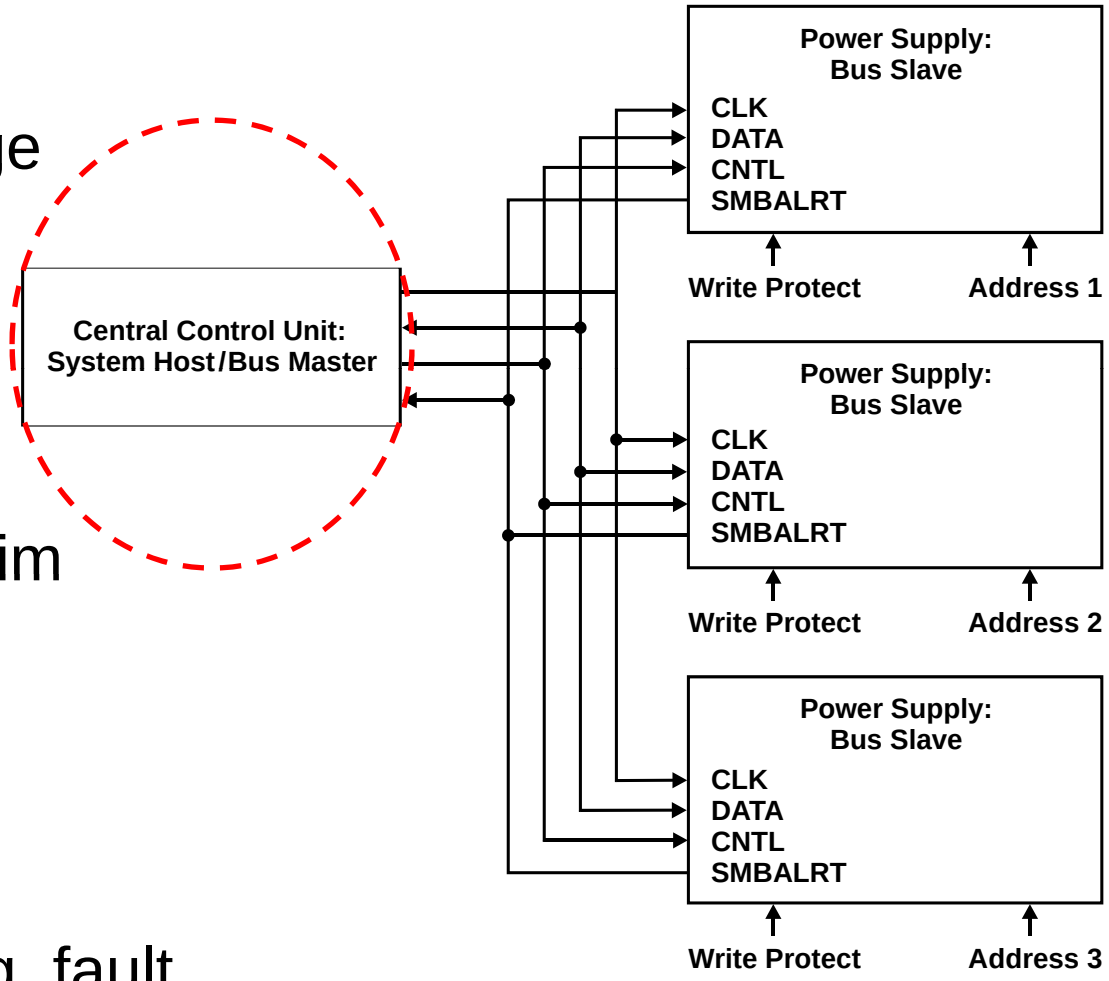
Typical PMBus System

- Single master – Host
- One or more slaves – Supplies
- Bus:
  - CLOCK and DATA required
  - CNTL and SMBALRT optional
- Each device has unique address
- Only communication between host and supplies – no inter-supply communication



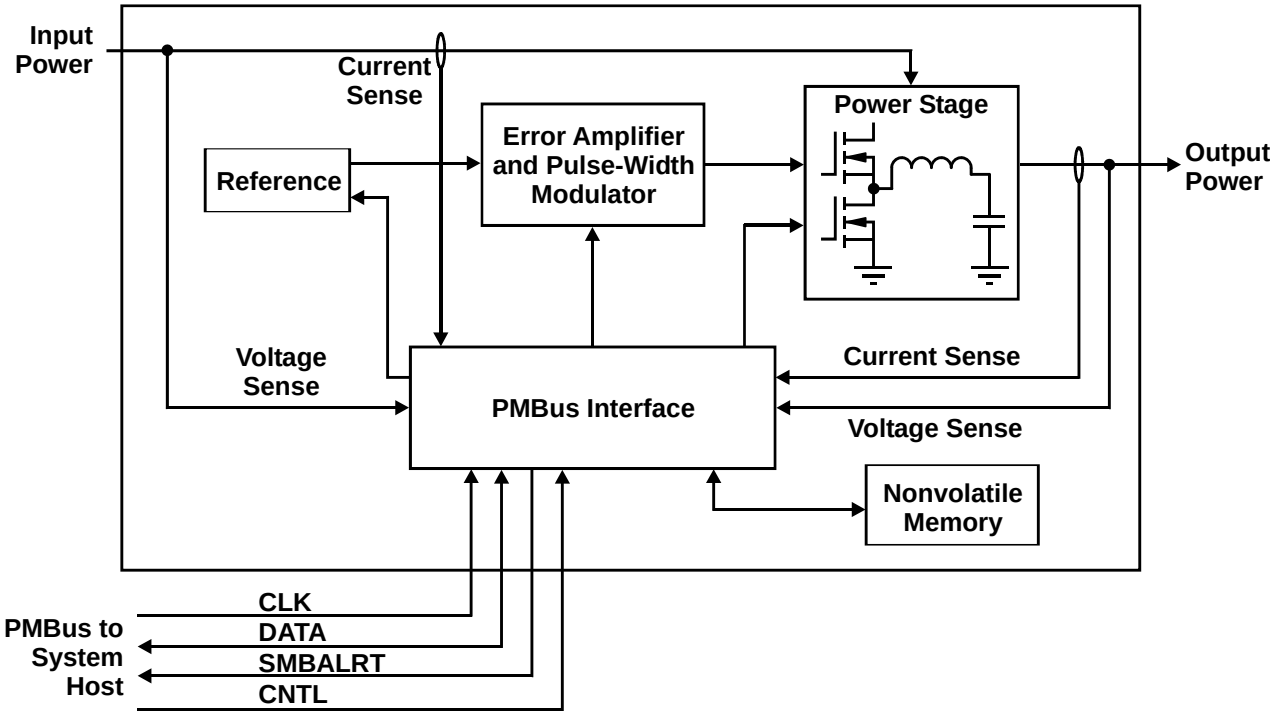
What is the Host?

- PC
  - Development stage
- Automatic Test Equipment
  - Production testing
  - Final parameter trim
- Microcontroller/ Microprocessor
  - In the field
  - Power sequencing, fault management, etc.

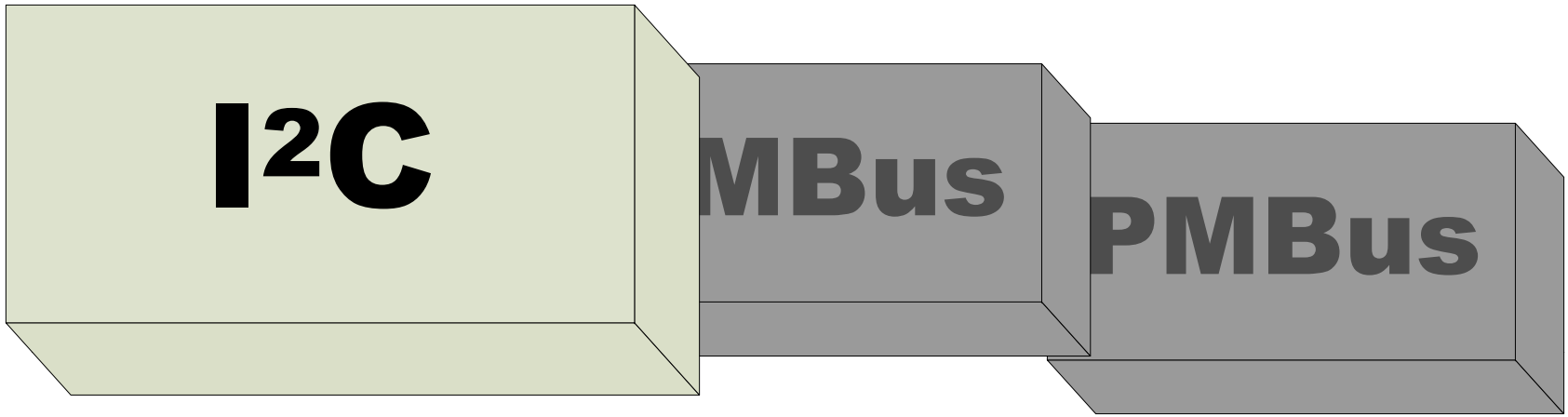


PMBus-Converter Architecture

- Similar to standard controller
- PMBus interface addon is typical
  - Hooks into controller for PMBus functions
- NVRAM
  - Store operating parameters, user data, etc.

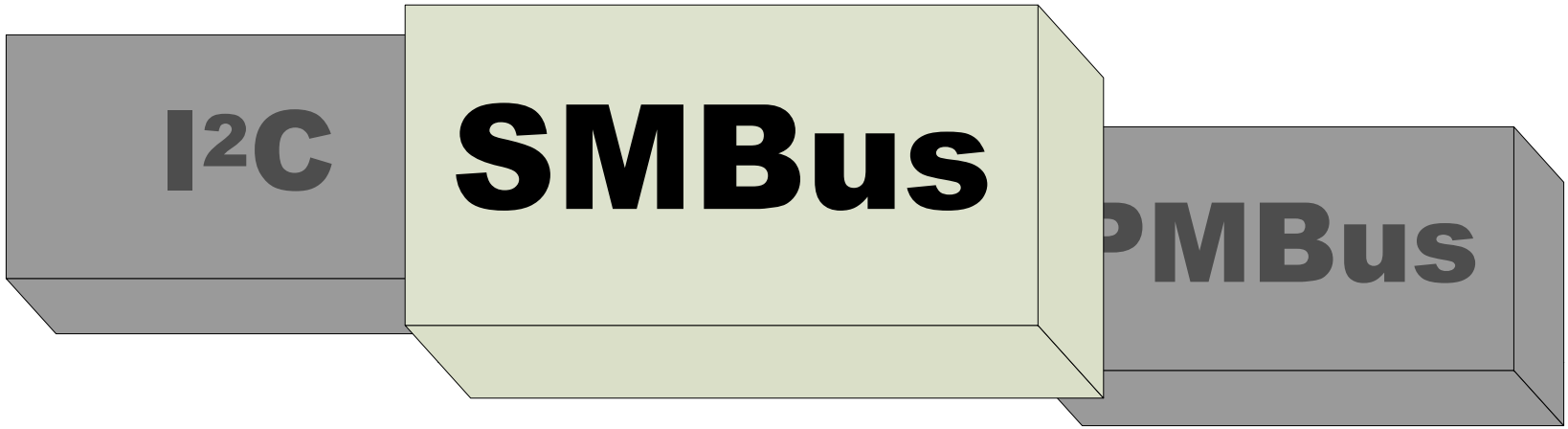


Ancestry of PMBus



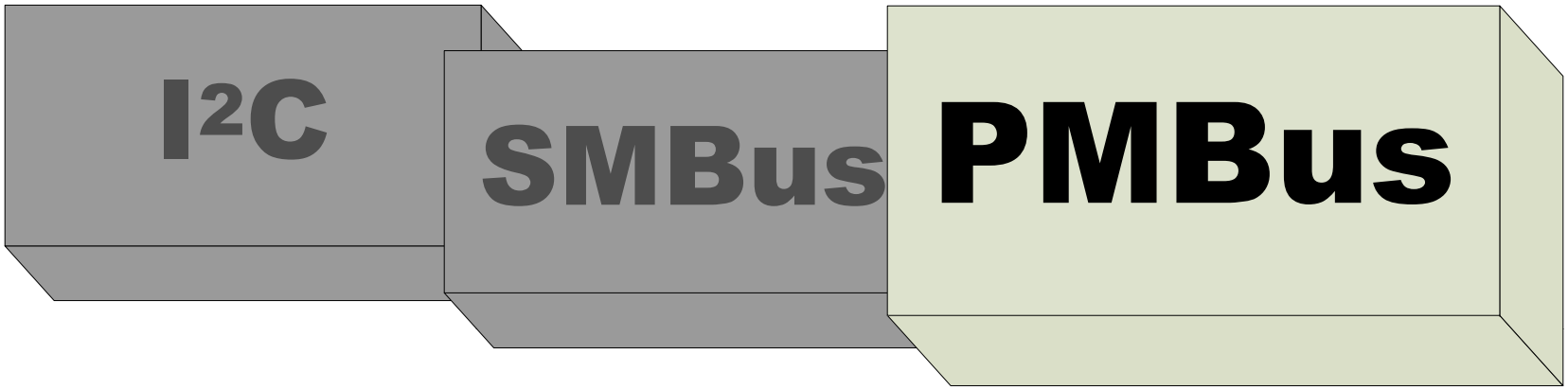
- Inter-Integrated Circuit – Philips development
- Simple serial 2-wire bus
- Vulnerable
  - Excessive clock stretching causing bus lockup
  - No standard error check

Ancestry of PMBus



- System Management Bus – www.smbus.org
- Fixes bus lockup from clock stretching issues
  - Bus times out and resets
- Standard error checking defined
- Host-notify protocol defined
  - Slave device can alert master when needed

Ancestry of PMBus



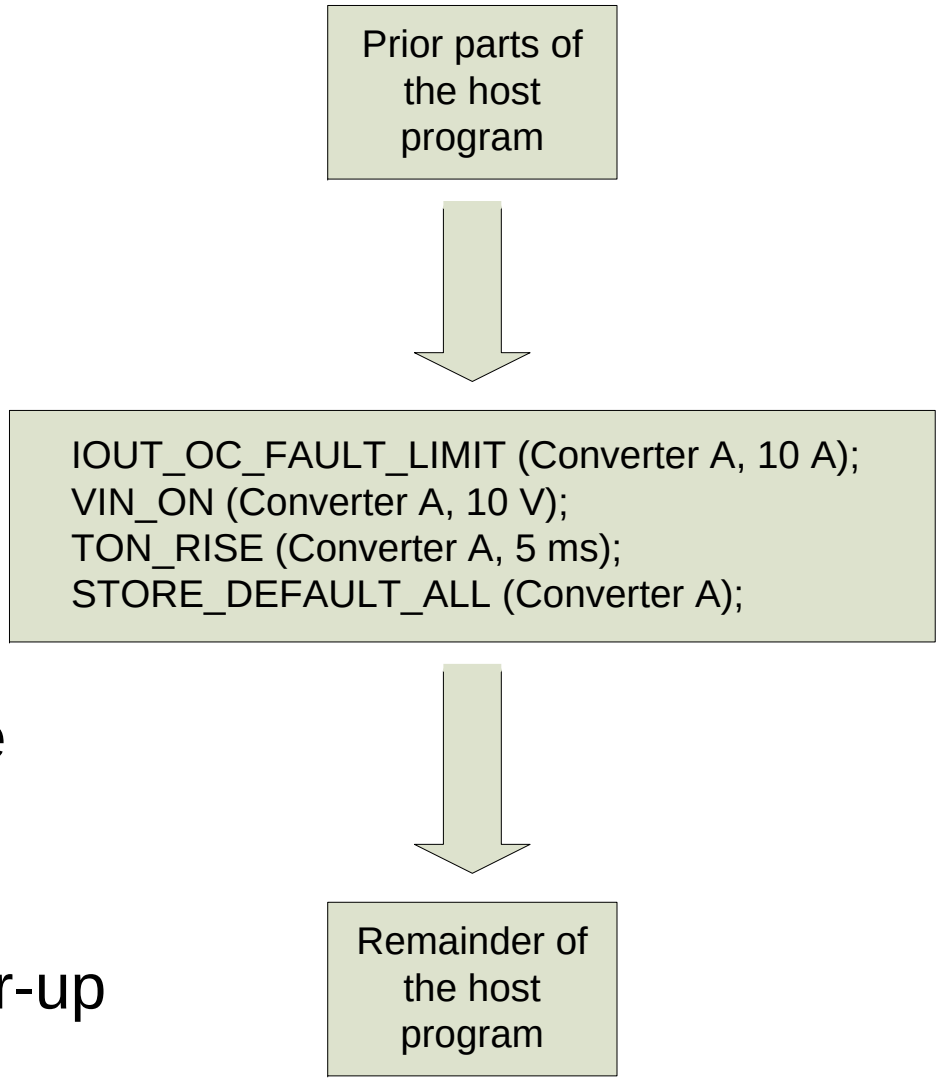
- Power Management Bus – www.pmbus.org
- Outgrowth of SMBus
- Notable differences
  - Host-notify protocol is optional
  - Group-command protocol
    - Multiple devices can execute commands at the same time

PMBus Compliance

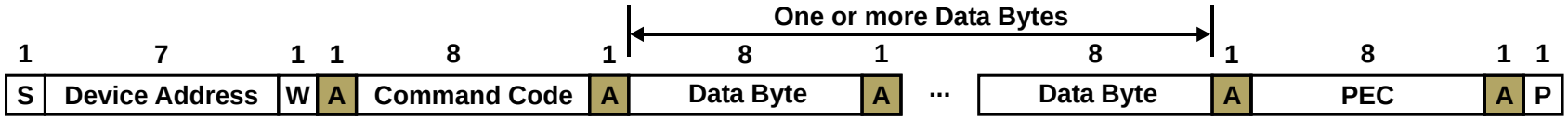
- PMBus specification is in two parts
- Compliant converters must:
  - Meet all Part I specifications
    - Hardware interface (electrical, timing)
  - Implement at least one command
    - Other than manufacturer specific
  - Supported commands must comply with Part II of the specification
    - Command language
  - Non-supported command
    - Respond according to Part II
- Also:
  - Must be able start without PMBus communication

Usage Mechanics

- Not hard!
- Simple sequential commands
- Example sets converter's:
  - Overcurrent limit
  - Input turn-on voltage
  - Soft-start time
  - Parameters to power-up defaults



PMBus Command



- PMBus command is fairly simple
- START and STOP conditions have DATA-change state while CLK is high.
  - Normal transfer: DATA is stable while CLK is high

Types of Commands Available

- **On, off, and margin testing**
- Output voltage related
- Addressing, memory communication, and capability
- Fault handling
- Sequencing
- Status
- Telemetry
- Other

- Deal with turning converter on and off, output margining
- Commands in group:
  - OPERATION
  - ON_OFF_CONFIG
  - VOUT_MARGIN_HIGH
  - VOUT_MARGIN_LOW

Types of Commands Available

- On, off, and margin testing
- **Output voltage related**
- Addressing, memory communication, and capability
- Fault handling
- Sequencing
- Status
- Telemetry
- Other

- Commands affect the output voltage of the converter
- Commands in group:
  - VOUT_COMMAND
  - VOUT_TRIM
  - VOUT_CAL_OFFSET
  - VOUT_SCALE_LOOP

Types of Commands Available

- On, off, and margin testing
- Output voltage related
- **Addressing, memory communication, and capability**
- Fault handling
- Sequencing
- Status
- Telemetry
- Other

- Deal with device addressing, memory, communication, and capability of the converter
- Commands in group:
  - STORE_DEFAULT_ALL
  - STORE_DEFAULT_CODE
  - PHASE, PAGE
  - WRITE_PROTECT

Types of Commands Available

- On, off, and margin testing
- Output voltage related
- Addressing, memory communication, and capability
- **Fault handling**
- Sequencing
- Status
- Telemetry
- Other

- Determine how converter responds to faults
- Defines what fault and warning levels are
- Commands in group:
  - IOUT_OC_WARN_LIMIT
  - IOUT_OC_FAULT_LIMIT
  - IOUT_OC_FAULT_RESPONSE
  - Many others

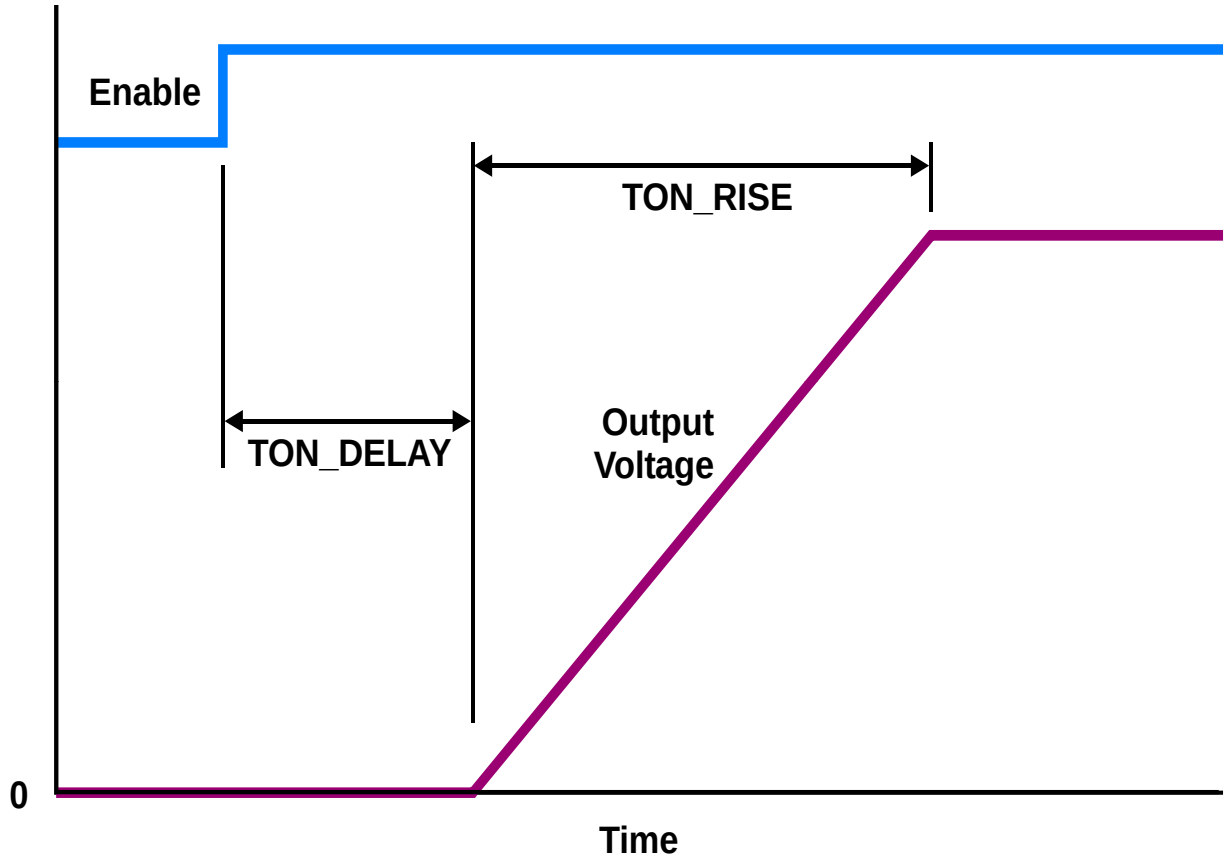
Types of Commands Available

- On, off, and margin testing
- Output voltage related
- Addressing, memory communication, and capability
- Fault handling
- **Sequencing**
- Status
- Telemetry
- Other

- Deals with timing of startup and shutdown
- Commands in group:
  - TON_DELAY
  - TON_RISE
  - TOFF_DELAY
  - TOFF_FALL

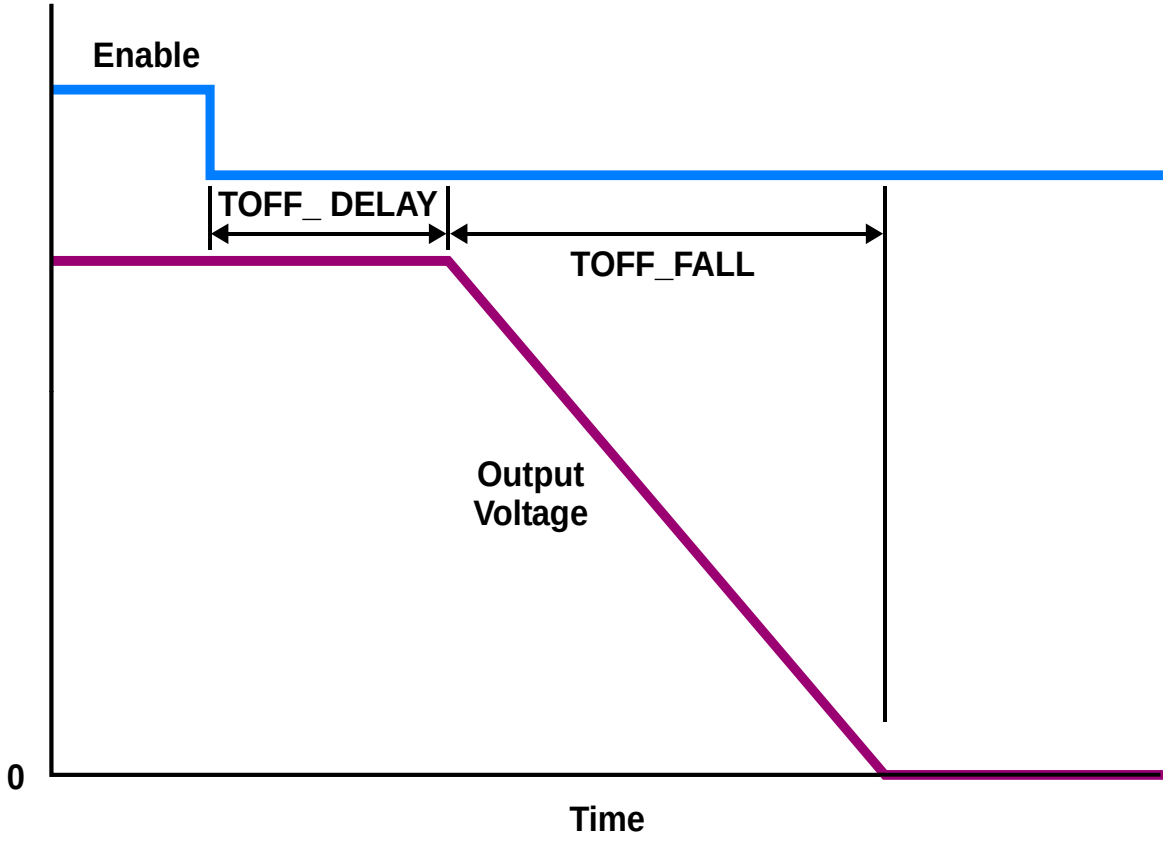
Power-On Sequencing

- Enable
  - CNTL
  - VIN_ON
  - OPERATION



Power-Off Sequencing

- Enable
  - CNTL
  - VIN_OFF
  - OPERATION



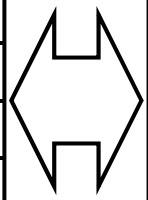
Types of Commands Available

- On, off, and margin testing
- Output voltage related
- Addressing, memory communication, and capability
- Fault handling
- Sequencing
- **Status**
- Telemetry
- Other

- Information about status of converter
- Commands in group:
  - STATUS_BYTE
  - STATUS_WORD
  - STATUS_IOUT
  - STATUS_CML

STATUS_BYTE and STATUS_WORD

STATUS_WORD
High Byte
7– VOUT
6 – IOUT/POUT
5 – INPUT
4 – MFR
3 – POWER_GOOD
2 – FANS
1 – OTHER
0 – UNKNOWN
Low Byte
7– BUSY
6 – OFF
5 – VOUT_OV
4 – IOUT_OC
3 – VIN_UV
2 – TEMPERATURE
1 – CML
0 – None of the above



STATUS_BYTE
7– BUSY
6 – OFF
5 – VOUT_OV
4 – IOUT_OC
3 – VIN_UV
2 – TEMPERATURE
1 – CML
0 – None of the above

- STATUS_BYTE part of STATUS_WORD
- Most flags have more detailed status commands

Types of Commands Available

- On, off, and margin testing
- Output voltage related
- Addressing, memory communication, and capability
- Fault handling
- Sequencing
- Status
- **Telemetry**
- Other

- Report back converter operating conditions
- Commands in group:
  - READ_VIN
  - READ_VOUT
  - READ_IIN
  - READ_IOUT
  - READ_DUTY_CYCLE
  - READ_FREQUENCY
  - Several others

Types of Commands Available

- On, off, and margin testing
- Output voltage related
- Addressing, memory communication, and capability
- Fault handling
- Sequencing
- Status
- Telemetry
- **Other**

- Commands that don't really fit elsewhere:
  - FREQUENCY_SWITCH
  - VIN_ON
  - VIN_OFF
  - IOUT_CAL_GAIN
  - IOUT_CAL_OFFSET
- Other classes not mentioned:
  - Manufacturer data
  - User data

Agenda

- PMBus overview
- **Some tasks and PMBus solutions**
- Brief design example

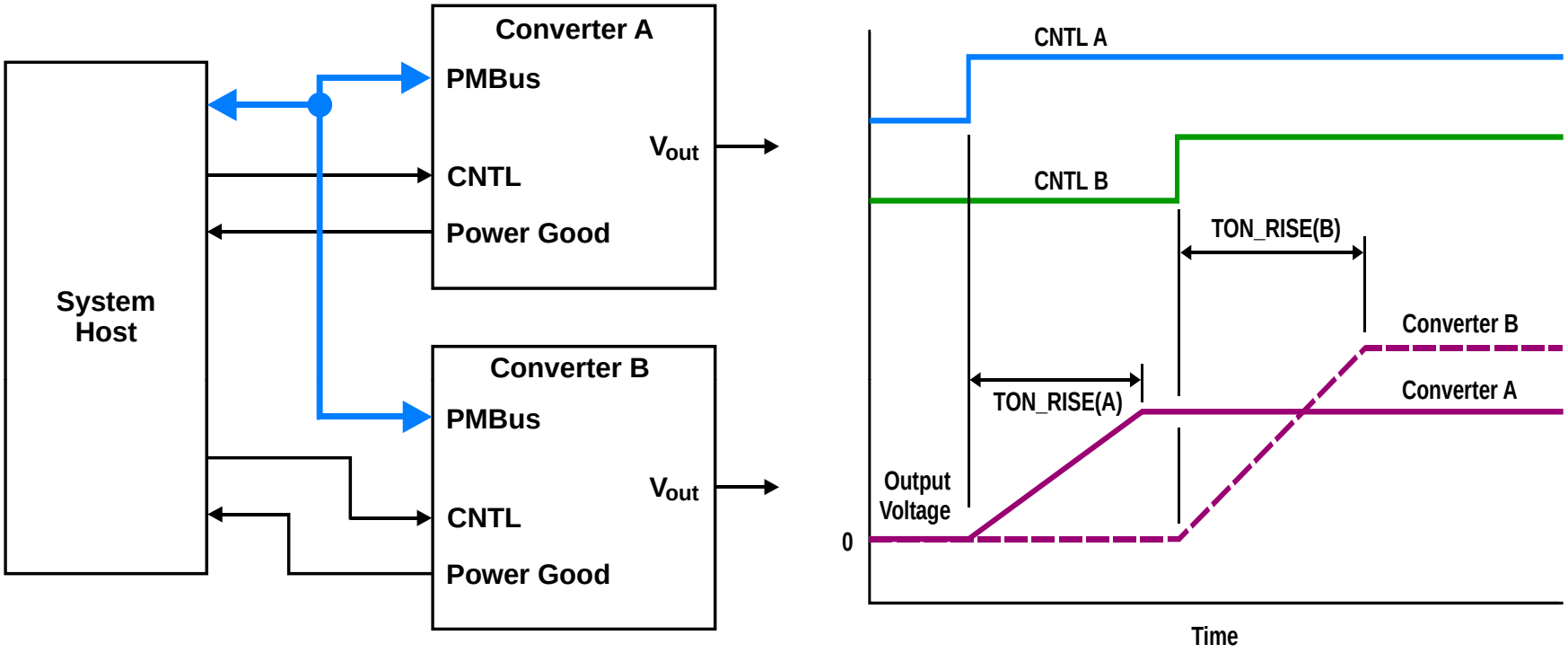
PMBus in the System

- Converter may not support all PMBus commands
- Consider tasks needed
  - Match capability to task requirements
- For example
  - Sequencing
  - Margining
  - Monitoring, etc.

Sequencing

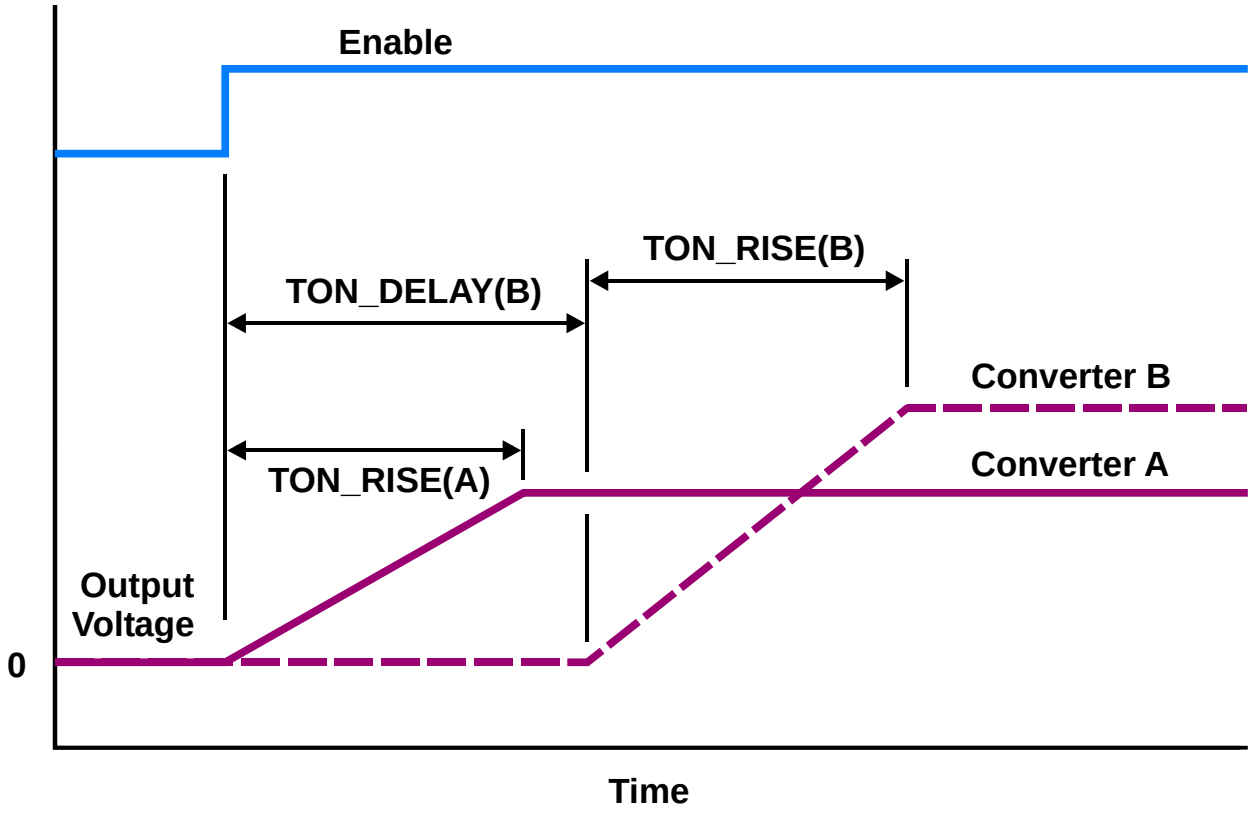
- Several ways to accomplish with PMBus facilities:
  - Remote enable
    - CNTRL pin
    - OPERATION command
  - Time delay
  - Open-loop tracking

Remote Enable



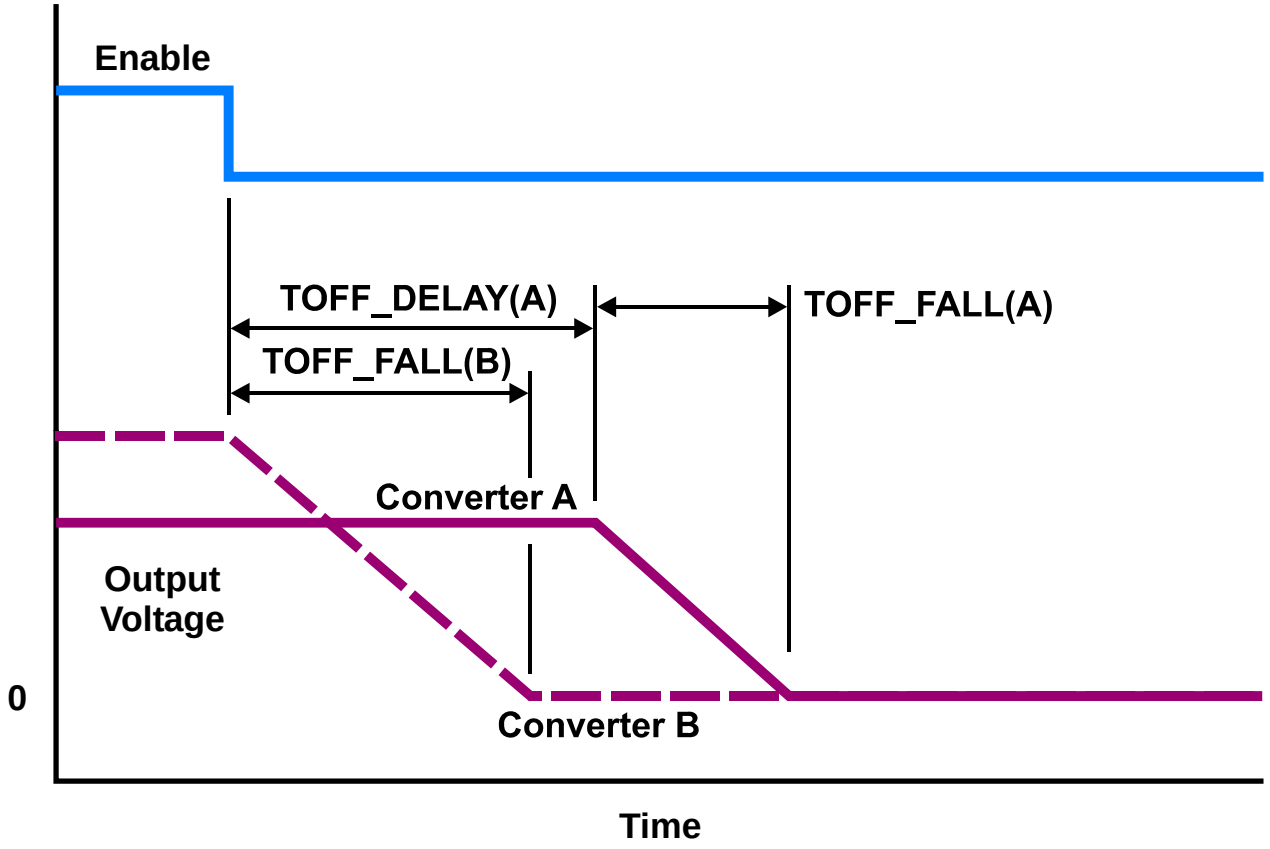
- Simple – not really PMBus solution
- CNTL is like ENABLE pin
- Controlled individually – conventional
- Similar operation possible using OPERATION commands over PMBus

Time-Delay Sequencing



- Converters programmed with startup delay (TON_RISE)
- Enabled simultaneously
- Eases burden on system host

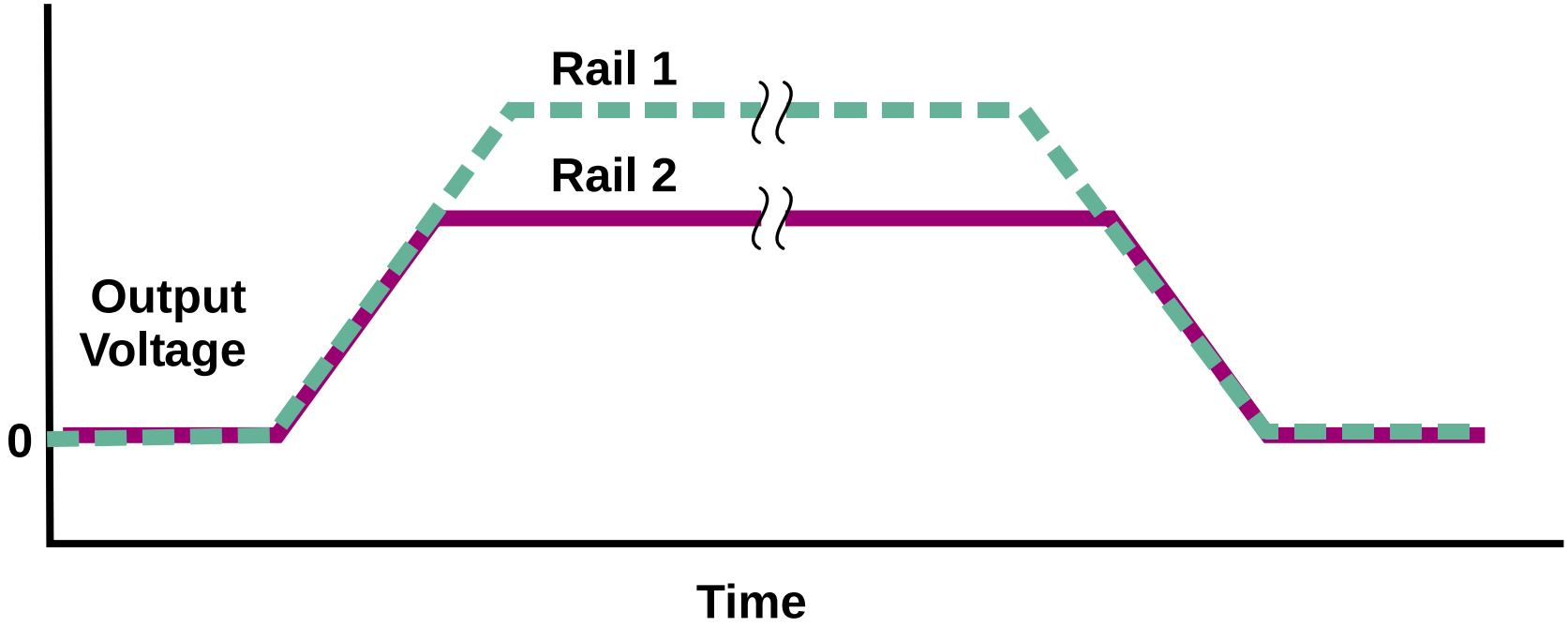
Time-Delay Off



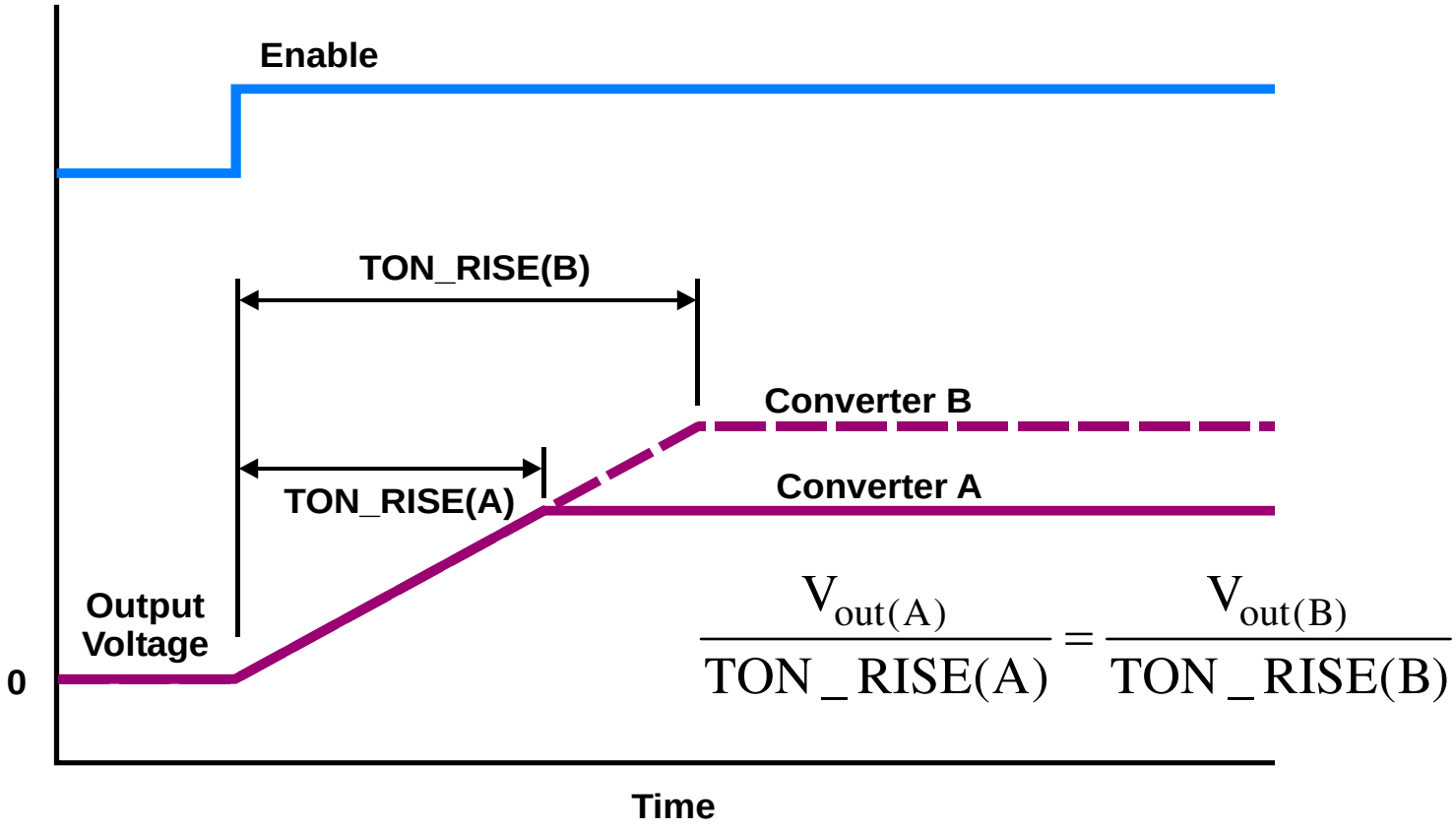
- Similar to previous – using TOFF_DELAY
- Watch fall times if TOFF_FALL not supported
- Both converters disabled simultaneously

Tracking

- Startup methodology keeping multiple-voltage rails close
  - Startup
  - Possibly shutdown
  - Startup and shutdown only here - Not for DDR

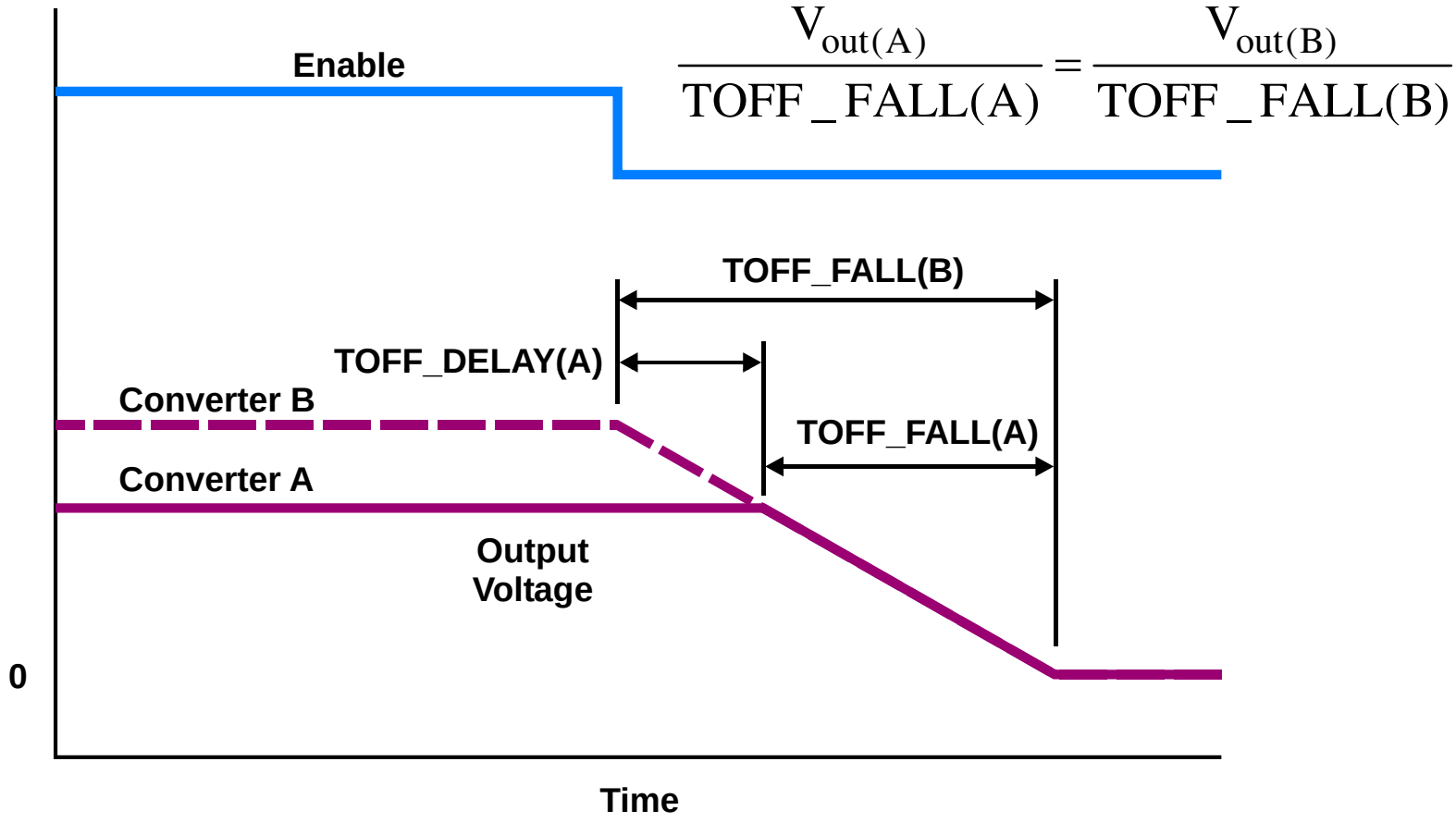


PMBus-Tracking Startup



- Set slopes of rise times equal
- Requires very fine control of TON_RISE

PMBus-Tracking Shutdown

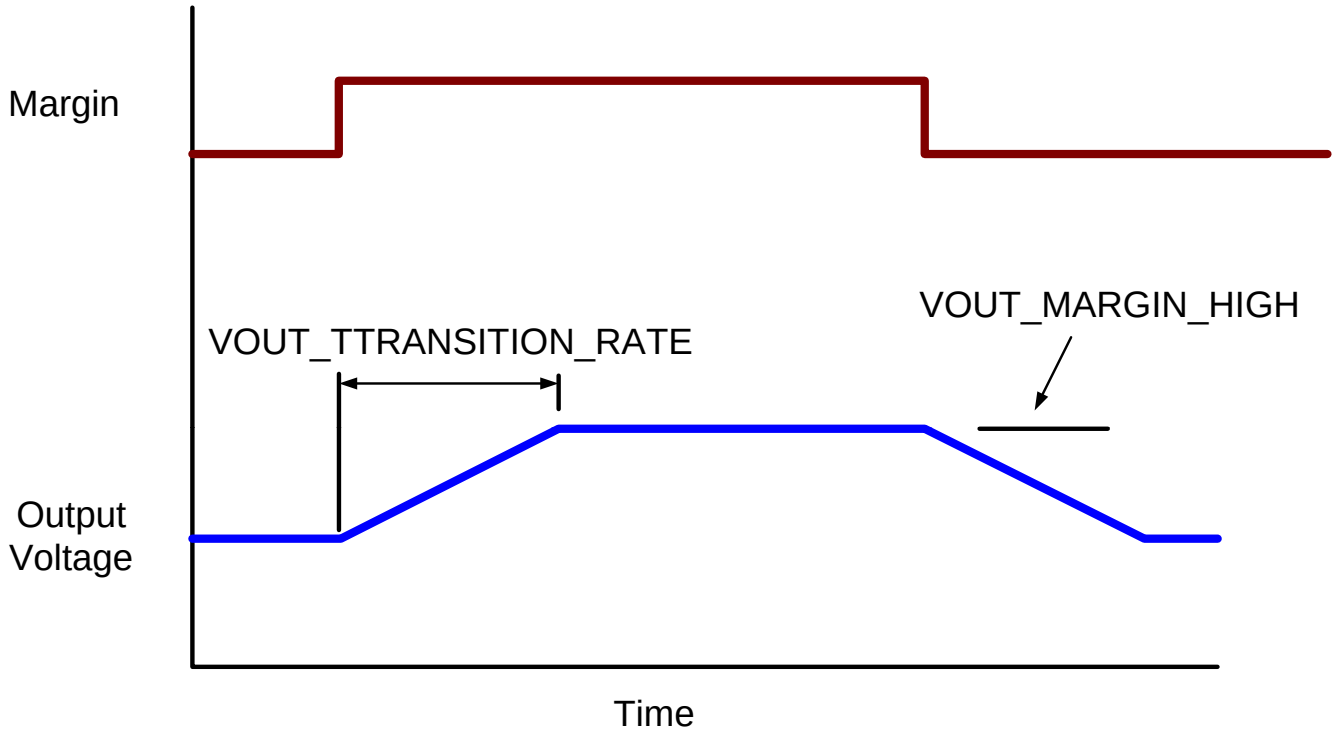


- Set down slopes equal – Requires TOFF_FALL support
- Note off delay required for lower voltage output

Margining

- Vary output voltages to test system
  - Requires either
    - Converter support
    - External device(s) to implement
- PMBus support
  - Variable margin voltages
  - Variable transition rate
  - Programmable fault response during margin
- Set margin voltages and use OPERATION command to execute

Margin Illustrated



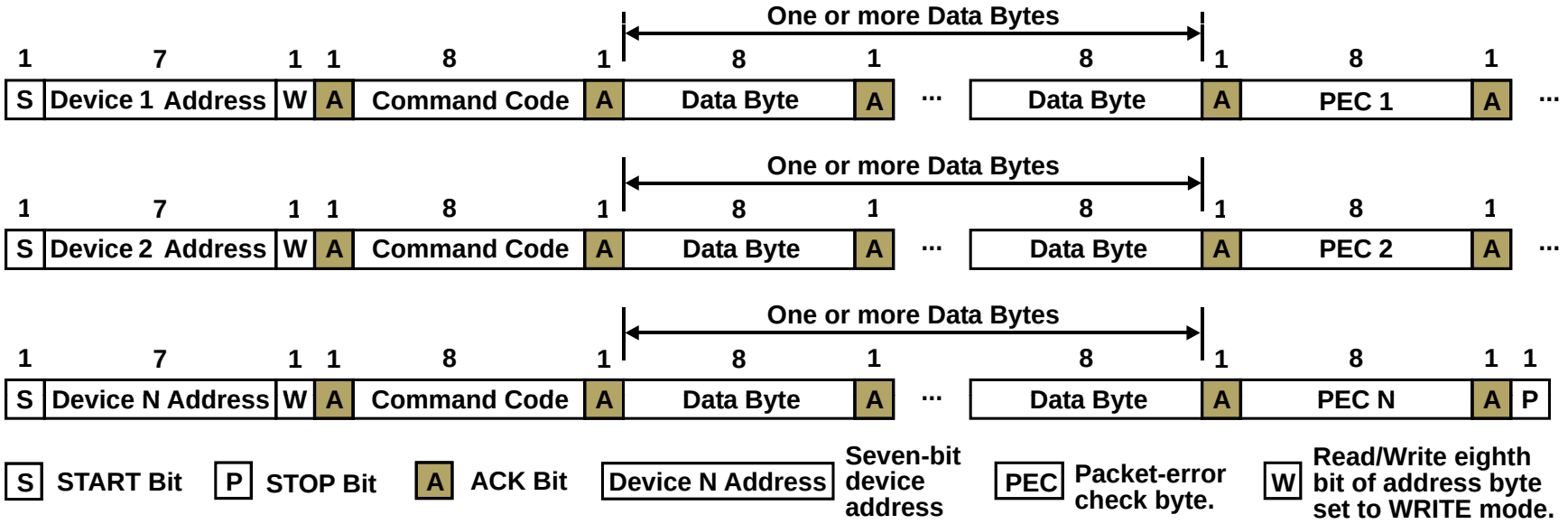
- Margin initiated using OPERATION command
- Transition rate and margin voltage are variable
- Much more flexible than hardware solution

Timing Issue

- How to synchronize PMBus commands
  - Startup/tracking require known timing relationships
  - Serial bus not inherently good
  - PMBus solution:

Group Command Protocol

PMBus Group Command



- Like single command except:
  - No STOP after first command
  - START repeats and next command is issued
  - STOP issued after final command
  - Commands simultaneously executed after the STOP

PMBus Group Command

Group command caveats:

1. No more than one command in a group can be addressed to a single slave device

2. No command in the group can require the slave to respond over the PMBus

Other Uses of PMBus

- Change system power parameters
  - Factory or field
  - No hardware modification
- Accidental overvoltage protection
  - Inadvertent/erroneous command
- System monitoring
- Compensation and capacitive loading
- Fault and incident tracking

Agenda

- PMBus overview
- Some tasks and PMBus solutions
- **Brief design example**

Design Example

- Important: Not all converters support all functions
- Determine necessary functions
- Select converter or controller accordingly

Target Specifications

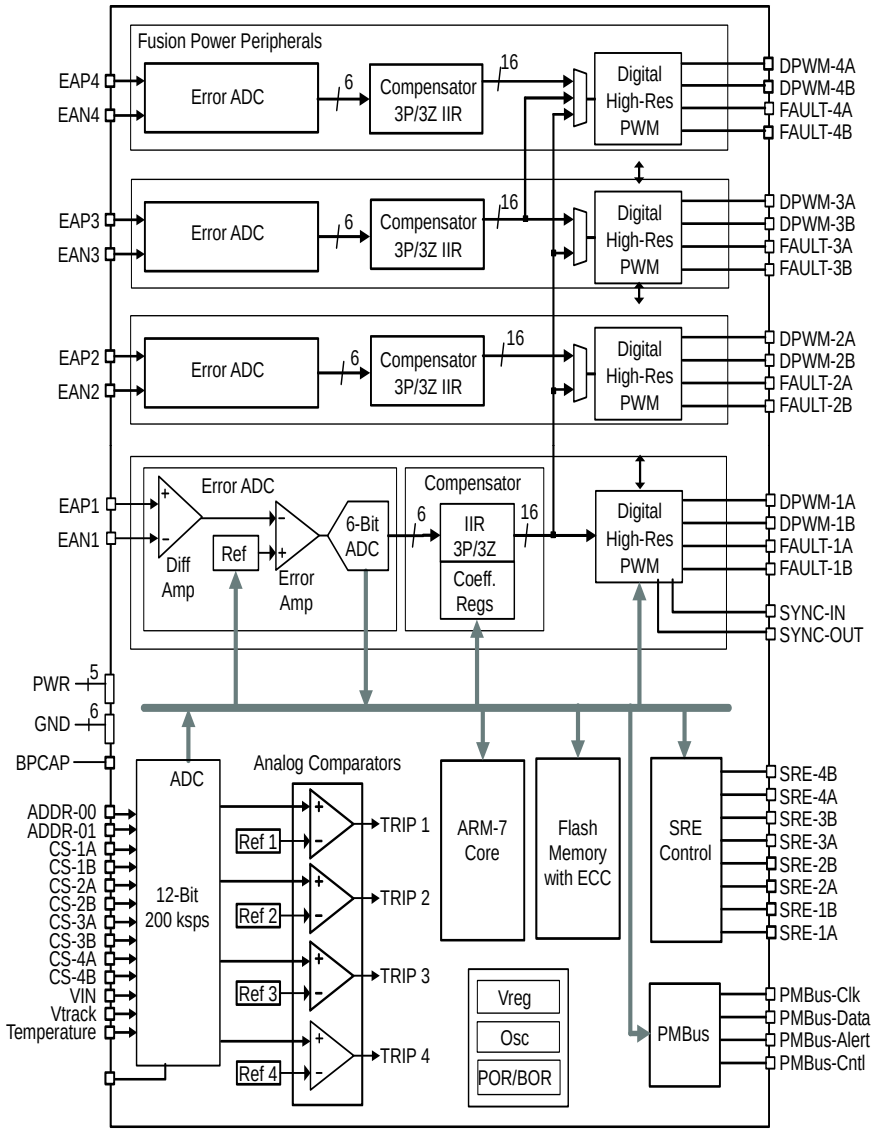
- Power supply for two processors in communications rack
  - Core and I/O for each
- Input voltage: 12 V
- Core output voltage: 1.2 V at 10 A per processor
  - Must be able to change from 1.1 V to 1.2 V
- I/O output voltage: 3.3 V at 3 A per processor
- Sequencing required: Core up first, then I/O's
- Margin testing support for field and factory
- Output voltage and current monitoring
- Minimal cost

PMBus Features Required

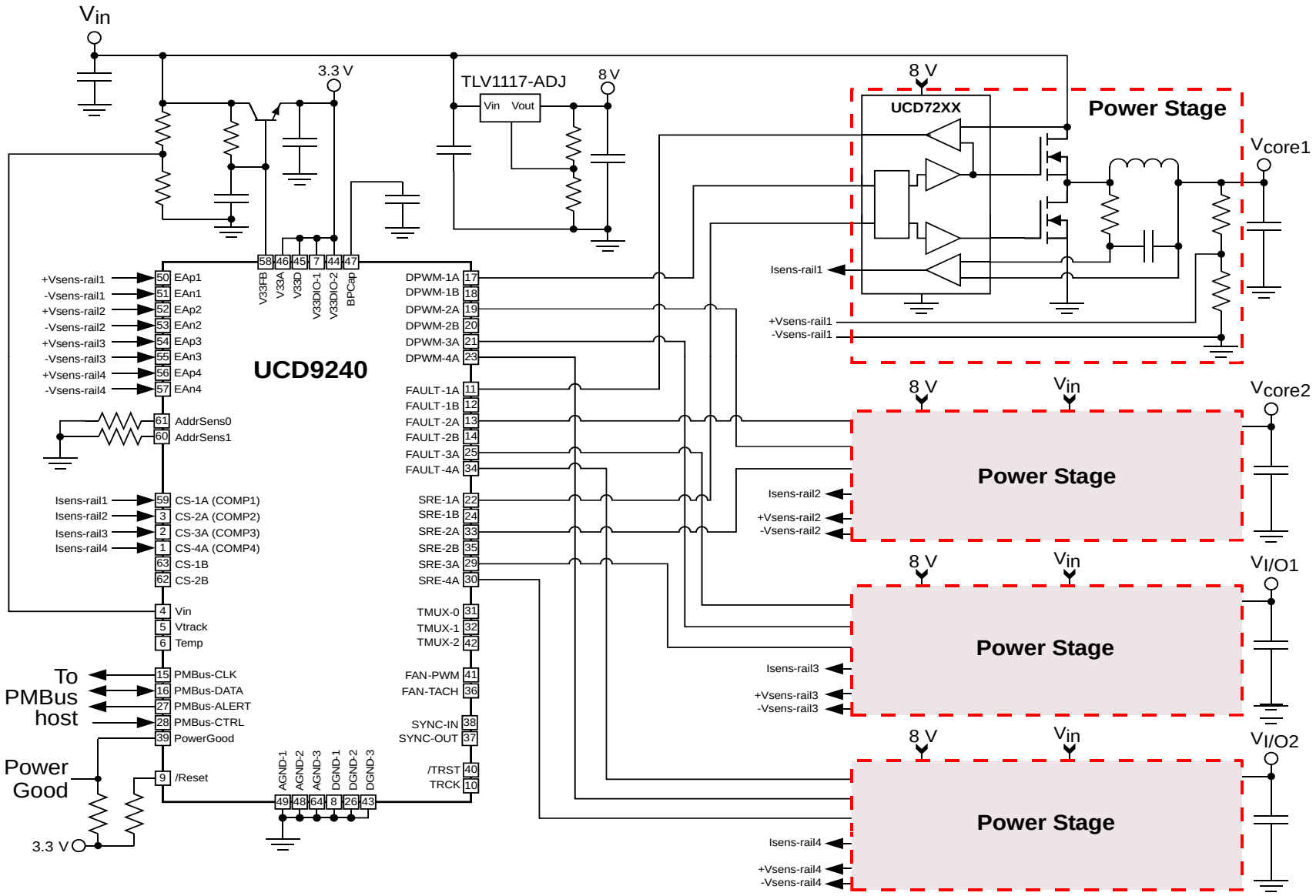
- Nonvolatile memory
  - Store default operating parameters
- Power-on sequencing
  - TON_DELAY
- VOUT_TRIM command
  - To adjust nominal output voltage on cores
- Margin commands
  - VOUT_MARGIN_HIGH, VOUT_MARGIN_LOW, OPERATION
  - VOUT_TRANSITION_RATE nice to have
- READ_VOUT, READ_IOUT
  - Monitor output power
  - READ_POUT would work too

Proposed Solution

- UCD9240
- Digital PMBus controller
- Features
  - Up to 4 independent rail control
  - ±1-mV feedback resolution
  - Hardware accelerated digital compensator
  - 12-bit monitoring of supply parameters
  - Programmable softstart and stop
  - Nonvolatile memory with ECC
  - Supports voltage tracking
  - Programmable margining and sequencing
  - PC-based design tools



Proposed Solution



Conclusions

- PMBus as a standard supports a wide range of functionality
- Many system-level tasks can be accomplished by the facilities provided by PMBus
- Converters won't generally fully support PMBus command set
  - Match system requirements to converter capability