Simple Adaptive Voltage Scaling Through Standard PMBus
Agenda

• What is AVS and Why is it important?
• How is AVS implemented?
• How can PMBus Implement AVS?
• What is Latency and how does it affect AVS with PMBus?
Break-Down of a Fixed Supply Voltage Spec Tolerances

Driving Spec
Power Supply Datasheet Specification
Sensing Tolerance
Design Margin (Temperature, CLK, etc)
Range of Minimum Supply Voltage for Valid Operation

Breakdown of Power Supply Voltage specification and how it can be improved

How do we improve (Reduce)
Improve DC set-point accuracy, Improve transient response
Remote Sensing, On-die Sensing
Dynamic AVS
Static AVS
The Problem with Fixed Supply Voltages

- For a Fixed supply voltage, that voltage must work for all yielded devices.
  - “Nominal” Supply voltage must be higher than needed by most parts for good yield.
  - “Weak” devices may need a higher supply voltage and be deemed “bad”
  - “Strong” devices may dissipate too much power and also deemed bad.
- Single Supply voltage must tolerance for all variables
  - Temperature, CLK frequency, Operational Mode, etc
- “Binning” often used to sort devices
  - “Weak” devices rated for lower clock frequencies
  - “Strong” devices rated for higher clock frequencies
  - What happens when yield distribution doesn’t match demand distribution?
    - Sell “Premium” parts at a discount rate?
    - Excess inventory of discount parts without demand?
The Supply Voltage (Vdd) is one of the biggest knobs that can be tweaked to compensate for the effects of process variability –
- For silicon samples stronger than target, Vdd can be set to a lower value to get lower power.
- For silicon samples weaker than target, Vdd can be set higher to ensure proper performance.
Static, Dynamic, Open and Closed Loop AVS

• **Static AVS** –
  – At manufacturing test, bits are set to indicate the minimum voltage required to run the chip at the rated performance.
  – During system operation, the programmed value is read out to the external power-supply (SMPS).
  – This type of AVS can only compensate for process variation. Other variations like temperature, regulator set-point, clock tolerance, etc. have to be margined for.

• **Open Loop Dynamic AVS** –
  – Provides look-up tables or offsets to account for detectable variations like temperature and operating state.

• **Closed Loop Dynamic AVS** –
  – Sensors compare the current transistor performance as powered against targets, which can include the provided Oscillator.
  – Sensor-performance targets (called “Ntargets”) for each sensor may be programmed into the device, or reference logic compared to the system clock.
  – Performance is tracked periodically during operation, by comparing the sensor-performance to the respective targets. If Logic is running faster than required, voltage is reduced. If Logic is running slower than required, supply voltage is increased.
  – In addition to process variation, temperature and operating state, Closed Loop AVS can compensate for supply voltage regulator error, clock oscillator tolerance, aging and drift over time.
How to Implement AVS?

• Parallel VID
  – Simple Implementation for low-resolution systems
    • 1 wire / bit
    • Dedicated Device to Supply communications channel

• Serial VID (sVID)
  – 2-wire dedicated communications channel
  – Upto 50MHz, <5μs Latency possible
  – Frequently restricted by Patent / Copy Right (Royalties)

• AVSBus
  – 2 or 3-wire dedicated communications channel
  – Upto 50MHz, <5μs Latency possible
  – Open Standard as part of PMBus 1.3

• PMBus
  – 3-wire shared communications channel
  – Upto 1MHz, Latency of 50μs – 250μs typical
  – Can be shared with operation, monitoring, fault detection and reporting
Implementing AVS with PMBus

- AVSBus
  - Part of PMBus 1.3
    - Covered in another presentation

- Standard PMBus Command – VOUT_COMMAND
  - Part of PMBus 1.0 – 1.3
  - Compatible across multiple vendors (Standard!)
    - PMBus does Allows Different VOUT_COMMAND Formats
    - READ VOUT_MODE First!
    - May impose increased system complexity

- MFR_SPECIFIC Commands
  - Not Standard, may be specific to vendor or even specific parts from a vendor
    - Typical imposes much less system complexity
    - May require custom software module to translate
Dedicated PMBus for AVS

Separate PMBus for each device (AVS Device as Host)
Dedicated PMBus for AVS

• PMBus Host
  – AVS Device serves as PMBus Host
  – Handles its own AVS Operation
  – Handles Operation, Fault Reporting and Telemetry if used

• Simplifies design, but loads AVS device with operating PMBus

• Common in System on a Chip and Embedded designs
PMBus Host as AVS Bridge
PMBus Host as Bridge

• PMBus Host
  – Normally PMBus Master
  – Handles Telemetry, Fault Reporting, Operation / Control
  – Reads AVS information from AVS Device
    • Can be compatible with non PMBus AVS
  – Writes AVS information to PMBus

• AVS Device
  – Does not need connection to PMBus
    • Needs to communicate with PMBus Host

• Common when interfacing Parallel VID to VOUT_COMMAND, Few AVS Devices
Multi-Point w/ Common Interrupt

PMBus Host

PS1

Single PMBus

PS2

Single Wire Interrupt

To other PMBus Devices

AVS Device 1

AVS Device 2

PMBus Host w/ Common AVS Interrupt
Multi-Point w/ Common Interrupt

• PMBus Host
  – PMBus Master
  – Handles Telemetry, Fault Reporting, Operation / Control

• AVS Device
  – Normally Slave on PMBus
  – Single GPIO interrupt wire to PMBus Host
    • Can be shared by multiple AVS Devices
  – Host will “pause” PMBus Transmissions when AVS Device needs to send AVS data
    • Finish current PMBus transmission, then idle BUS

• Simpler Host Implementation w/ multiple AVS devices

• Common in systems with several AVS devices with low AVS update rates
AVS Latency Using PMBus

- **Response Latency**
  - Delay from receiving command to target output voltage
  - Typically very short – μs
  - **VOUT_TRANSITION_RATE**
    - Defined in mV/μs

- **Communication Latency**
  - Delay from deciding to change VOUT to power supply receiving command
  - Depends on Bus Speed, Error Correction & Master Architecture
    - 50μs – 100s μs
    - 100kHz, 400kHz, 1MHz
    - PEC / No PEC?
    - Dedicated Bus? Multi-Master?
How Long is a PMBus Transmission?
Latency with different schemes

- **Dedicated PMBus**
  - Only need to wait for VOUT_COMMAND Length
    - 38-bits (No PEC) 47-bits (w/ PEC)
    - 95μs / 118μs (400kHz)
- **Multi-Point w/ Interrupt & PMBus Host as Bridge**
  - May need to wait for current transmission to complete
    - READ WORD w/ PEC = 57 bits (48 w/o PEC)
  - Need to allow Idle time after last transmission
    - 4.7μs
  - Send VOUT_COMMAND
    - 38-bits (No PEC) or 47-bits (w/ PEC)
    - Upto 220μs / 265μs (400kHz)
Implications of using VOUT_COMMAND?

• Initial “Default” Voltage on Power-up?
  – Default Reference w/ Resistor Divider
    • Requires VOUT_SCALE_LOOP Programming
    • Variable Resolution & Rounding limitations
  – Pre-Power On Programming of VOUT
    • Production In-Circuit Test Programming, Power-On Sequencing, System Management Host
  – Pin Programmed Values (PPV)
    • Range & Resolution verse # of Resistors/Pins

• Re-Start Conditions?
  – VOUT_COMMAND not reset on restart per PMBus
    • How do you change VOUT_COMMAND if the AVS source is the Powered Device?
Summary

• AVS is becoming more common
  – Reduce End User Cost (Power)
  – Reduce Manufacturer Cost (Improve Yield)
  – Improve Performance!

• Parallel AVS is changing to Serial AVS
  – Easier Routing
  – Greater Resolution
  – More Flexibility

• Open Standards like PMBus driving the future
  – Standard Library Modules
  – ASIC and Power Supply Vendor inter-compatibility

• Flexible Solutions to meet variety of needs
  – Dedicated PMBus for simpler system
  – Multi-Point PMBus for complex multi-processor systems
  – Host as Bridge for Non-PMBus AVS needs
  – AVS Bus for low-latency, high-speed dynamic requirements!