PMBus™
Power System Management Protocol
Application Note AN001

Using The ZONE_READ And ZONE_WRITE Protocols

Revision 1.0.1

7 Jan 2016

www.powerSIG.org

Filename: PMBus_AN001_Rev_1_0_1_20160107.docx
Last saved: 07 Jan 2016, 20:13

DISCLAIMER

This application note is provided “as is” with no warranties whatsoever, whether express, implied or statutory, including but not limited to any warranty of merchantability, non-infringement, or fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification or sample.

In no event will any application note co-owner be liable to any other party for any loss of profits, loss of use, incidental, consequential, indirect, or special damages arising out of this application note, whether or not such party had advance notice of the possibility of such damages. Further, no warranty or representation is made or implied relative to freedom from infringement of any third party patents when practicing the application note.

Other product and corporate names may be trademarks of other companies and are used only for explanation and to the owner’s benefit, without intent to infringe.

REVISION HISTORY

REV	DATE	DESCRIPTION	AUTHORS
1.0	16 Nov 2015	First release	Chris Eckhoff, Maxim Integrated Michael Jones, Linear Technology Travis Summerlin Texas Instruments Robert V. White Embedded Power Labs
1.0.1	7 Jan 2016	Corrected Figure 15	Robert V. White Embedded Power Labs

Table Of Contents

Table Of Figures

Table Of Tables

1. Introduction

The PMBus 1.2 protocol enabled the electronics industry to standardize communications to their power conversion circuits. Controlling, configuring, and monitoring of ac-dc power supplies, isolated dc converters (“bricks”), and non-isolated point-of-load (POL) converters is now possible across many vendors’ power solutions. However, the SMBus protocol, which underlies PMBus 1.2, has speed and protocol limitations that can include system design challenges, especially fault handling and sequencing of very large systems. These functions are bus-access intensive for PMBus 1.2 systems.

A PMBus committee made up of industry experts was formed to discuss the current state of PMBus. They shared similar stories: A server design engineer was lamenting over the fact that they could not get their PMBus enabled embedded controller to deal with the fault handling from the system’s point-of-load (POL) converters. “The POL’s just retry and alarm constantly, I can’t get to all of them fast enough to find out what’s going on!” A system architect was also complaining “I have systems with modular cards that include PMBus devices on them. The system needs to know exactly how it is configured on power up. I wish the system could figure that out itself, and do it quickly.” The PMBus committee needed to address deficiencies, but stay within the bounds of the existing protocols.

The PMBus committee formed a specification working group to pursue proposals that would enhance the current protocol, adding the ability to read from or to write to all (or a subset of) a system’s devices including any pages within those devices, in a single transaction. PMBus 1.3.1, which is based on the hardware specification SMBus 3.0 [R03], includes new enhancements called Zone Write and Zone Read. Zone Write and Zone Read provide faster transactions than the older PMBus 1.2 protocol while maintaining backward compatibility.

2. Overview: A Protocol For Faster Bus Transactions

The new Zone Protocol is comprised of three new features:

1. Zone Configuration
2. Zone Write
3. Zone Read

A zone is a set of slave devices on a shared PMBus that can react to Zone Write and Zone Read operations. Any slave device can be assigned to a zone with a configuration command, and a master can set the Active Write Zone and Active Read Zone to communicate with a specific zone, in much the same way a master can set the PAGE of a slave device to communicate with a subset of its functionality. The Active Write Zone and Active Read Zone do not have to be the same value. The Active Write Zone is often not the same as the Active Read Zone because it is common to “control” one subset of slave devices and “monitor” telemetry from a different subset.

2.1. Zone Configuration

Zone configuration consists of two steps. First, all devices in the system that will be participating in zone operations must be assigned to a write zone and read zone using the ZONE_CONFIG command.

Once the master has assigned devices to various write and read zones, it must tell the devices in the system the zone with which it wants to communicate. The master does this by using the ZONE_ACTIVE command.

2.2. Zone Write

A Zone Write is an operation that sends a single command to multiple devices in one transaction using the ZONE_WRITE command. For example, the ZONE_WRITE command may be used to send a PMBus OPERATION command to multiple devices to turn them on or off simultaneously.

2.3. Zone Read

A Zone Read is an operation that allows a single command to read from multiple devices in one transaction using the ZONE_READ command. A Zone Read operation is more complicated than a Zone Write operation because if a master tries to read from more than one slave device, they will all answer. Therefore, there are two modes:

- All slave devices continually transmit their data until each of the slave devices has won the bit-by-bit arbitration and been able to send its data to the master, and
- One slave device transmits its data as a result of winning arbitration

The first case results in a list of the requested data from all slave devices in the Active Read Zone. The second case results in the requested data from a single slave device, and there is provision in the protocol to determine which slave device answers (the competition rules). For example, the master may configure the response to receive the largest or smallest value. One possible use of this would be to query all of the devices in a system to find out which one has the highest output current or which one has the highest temperature.

3. The ZONE_CONFIG Command

When designing a system, the system engineer may partition the power system into “zones”. For example, a server with four processors and the supporting memory and I/O, may be divided into four zones so that the power for each processor and its supporting devices can be managed together. Or a network router with 32 high speed network ports might assign the power converters supporting each port to their own zone so that power to each port can be easily managed.

Before a device can be used for a Zone Write or Zone Read operation, it must be assigned to a write zone and a read zone. For any device, it does not have to be assigned to the same zone number for write and read. That is, a device may have different values for its assigned write zone and its assigned read zone.

3.1. Example: ZONE_CONFIG Command

As an example, consider the system shown below in Figure 1 and suppose that at system power-up the master knows the address and pages of each device (the discovery of devices in an unknown system is discussed in Section 8.1) and that it wants to assign each device to a write zone and read zone as shown in Table 1.

The sequence of commands to configure the devices to the desired zones is shown in Figure 2. If you are not familiar with the way PMBus transactions are illustrated, please see APPENDIX I.

Table 1. Example Device Zone Assignments

Device Number	Address	Page Number	Write Zone	Read Zone
1	34h	N/A	03h	04h
2	35h	00h	02h	03h
		01h	03h	03h
3	27h	N/A	02h	04h
4	38h	N/A	03h	04h
5	40h	N/A	02h	04h

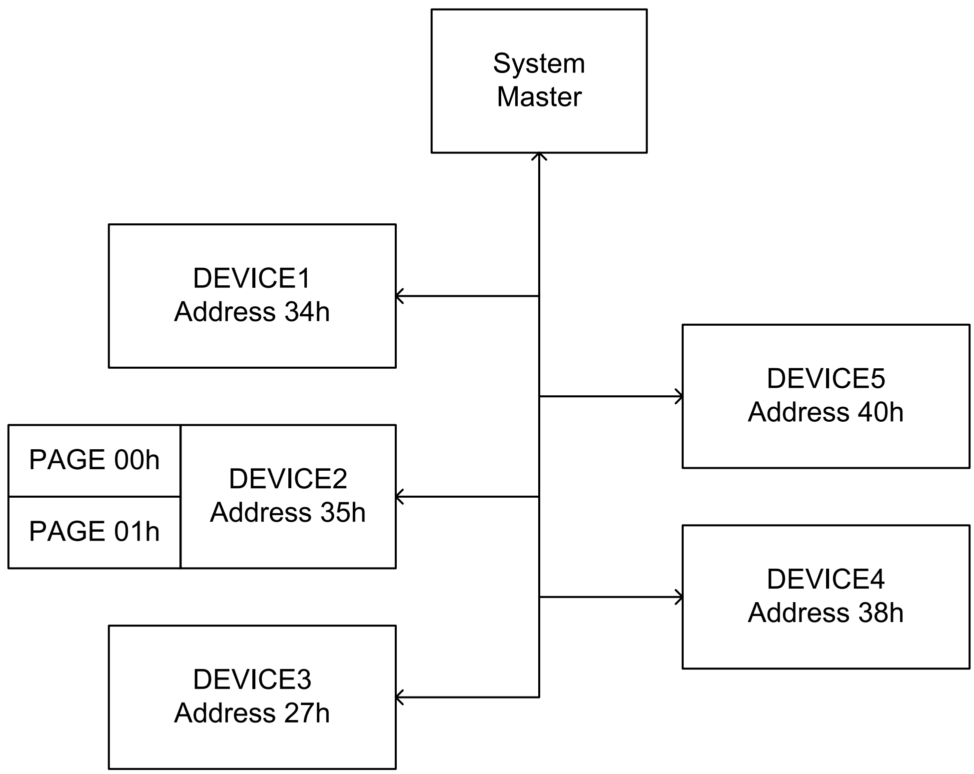

Figure 1. Example PMBus System

3.2. Zone Numbering

Users may assign any value from 00h through 7Fh to the write zone or read zone of a zone capable PMBus device.

The values from 80h through FFh are reserved as listed below:

- Manufacturer (80h – BFh)
- Reserved For Future Use (C0h – FDh)
- All Zone (FFh)
- No Zone (FEh)

The All Zone is a special zone that allows a master to address all devices participating in zone operations in one bus transaction. A device is not configured to the All Zone. Rather, when the master is sending a zone command, it may use the value FFh (All Zone) to send a

ZONE_WRITE or ZONE_READ command to all zone capable devices regardless of the zone to which they have been assigned.

The No Zone is a configuration option for a slave device when the master does not want that slave device to respond to any ZONE_WRITE or ZONE_READ commands. Note that devices assigned to the No Zone will still respond to ZONE_CONFIG and ZONE_ACTIVE commands.

The Manufacturer Zones are custom zones defined by a designer or manufacturer of a slave device and enables specialized behavior. The Reserved Zones are for future PMBus specification enhancements.

Set the zone assignments for Device1 at address 34h:

	7	1	1	8	1	8	1	8	1	
S	Device Address (34h)	Wr	A	ZONE_CONFIG (07h)	A	Assigned Write Zone (03h)	A	Assigned Read Zone (04h)	A	P

Set the PAGE for Device2 to PAGE 00h:

	7	1	1	8	1	8	1	
S	Device Address (35h)	Wr	A	PAGE (00h)	A	Page Number (00h)	A	P

Set the zone assignments for PAGE 00h of Device2 at address 35h:

	7	1	1	8	1	8	1	8	1	
S	Device Address (35h)	Wr	A	ZONE_CONFIG (07h)	A	Assigned Write Zone (02h)	A	Assigned Read Zone (03h)	A	P

Set the PAGE for Device2 to PAGE 01h:

	7	1	1	8	1	8	1	
S	Device Address (35h)	Wr	A	PAGE (00h)	A	Page Number (01h)	A	P

Set the zone assignments for PAGE 01h of Device2 at address 35h:

	7	1	1	8	1	8	1	8	1	
S	Device Address (35h)	Wr	A	ZONE_CONFIG (07h)	A	Assigned Write Zone (03h)	A	Assigned Read Zone (03h)	A	P

Set the zone assignments for Device3 at address 27h:

	7	1	1	8	1	8	1	8	1	
S	Device Address (27h)	Wr	A	ZONE_CONFIG (07h)	A	Assigned Write Zone (02h)	A	Assigned Read Zone (04h)	A	P

Set the zone assignments for Device4 at address 38h:

	7	1	1	8	1	8	1	8	1	
S	Device Address (38h)	Wr	A	ZONE_CONFIG (07h)	A	Assigned Write Zone (03h)	A	Assigned Read Zone (04h)	A	P

Set the zone assignments for Device5 at address 40h:

	7	1	1	8	1	8	1	8	1	
S	Device Address (40h)	Wr	A	ZONE_CONFIG (07h)	A	Assigned Write Zone (02h)	A	Assigned Read Zone (04h)	A	P

Figure 2. Setting The Zone Assignments Of The Devices In The Example System

3.3. ZONE_CONFIG Use Cases

When and how devices are assigned to write and read zones using the ZONE_CONFIG command primarily depends on two factors:

- Do the slave devices have non-volatile memory so that their assigned zones can be stored during power down and

- Is the system configuration fixed at the time of manufacture or can it change in time (such as a system that is based on cards in a backplane or cards that can have mezzanine cards added or removed).

If the zone capable devices do not have non-volatile memory then they must be configured by the bus master at each power up (or reset after a brown out).

If the zone capable devices do have non-volatile memory for their assigned zone numbers, then there are more options. For example, if the device is a system with a fixed configuration, such as a large circuit board for a network router, then the devices could be programmed at the time the circuit board is manufactured using automatic test equipment. System power up is now faster as the master does not have to go through the process of configuring all devices on the bus.

If the system configuration is flexible, then the master will have to determine what devices are on the bus, using the discovery process (Section 8.1), each time the system is powered up. Newly discovered zone capable devices will need to be configured using the ZONE_CONFIG command as part of the power up process. Devices that were previously on the bus that have non-volatile memory will not need to be configured, again speeding up the system power up process.

4. The ZONE_ACTIVE Command

Once the zone capable devices have been assigned to write zones and read zones, the master needs a way to send zone commands to specific zones. It does this by sending the ZONE_ACTIVE command to all devices on the bus using the special ZONE_WRITE address (37h). Figure 3 shows and example of a ZONE_ACTIVE command.

	7	1	1	8	1	8	1	8	1	
S	ZONE WRITE Address (37h)	Wr	A	ZONE_ACTIVE (08h)	A	Active Write Zone (FFh)	A	Active Read Zone (03h)	A	P

Figure 3. ZONE_ACTIVE Command Example

In this case, the Active Write Zone is set to the All Zone (FFh), which is ideal for using the OPERATION command to turn all slave devices on and off. The Active Read Zone is set to 03h.

A ZONE_ACTIVE command can be sent at any time to change the target zone for Zone Read and Zone Write operations. A zone-capable slave device will always respond to a ZONE_ACTIVE command, regardless of its Zone Configuration.

As mentioned above, the ZONE_ACTIVE command may only be sent to the Zone Write address. If a master sends a ZONE_ACTIVE command to any other address, a slave device at that address shall NACK the command and respond as having received an unsupported command.

There is a special address for the ZONE_WRITE command because the command is global to the bus. In PMBus protocol, this is the only way to send a command to the whole bus. So to review, the ZONE_CONFIG command uses the address of each slave device, and the ZONE_ACTIVE command uses the special ZONE_WRITE address.

Before we discuss the commands for reading and writing zones, we need to clarify the behavior of slave devices based on the Active Zone and their assignment to a specific zone number, the All Zone (FFh), or the No Zone (FEh). Table 2 gives a matrix of the behavior. Manufacturer Zones may behave differently in other respects, but not in decisions about when to respond or not.

Table 2. Zone Configuration And Device Responses

Master Has Set The ACTIVE_ZONE To:	Device Has Been Configured To Zone Number		
	FFh (All Zone)	FEh (No Zone)	A Zone Number In The Range 00h-7Fh or 80h-BFh
FFh (All Zone)	Not Permitted[1]	Ignore	Respond
FEh (No Zone)	Not Permitted[1,2]	Not Permitted[2]	Not Permitted[2]
A Zone Number In The Range 00h-7Fh or 80h-BFh	Not Permitted[1]	Ignore	Respond Only If The Active Zone Number Equals The Assigned Zone Number, Otherwise Ignore The Zone Command

Note 1: It is not permitted to assign a device to the All Zone

Note 2: It is not permitted to set the Active Zone to No Zone

We start with:

- The master used the ZONE_ACTIVE command to set both the Active Write Zone and Active Read Zone to the All Zone (FFh) and
- The master initiates a Zone Write (or Zone Read) operation.

Then:

- If the device's write zone (or read zone) is assigned to the No Zone (FEh), it ignores all Zone Write (or Zone Read) operations because the assignment to the No Zone (FEh) instructs the device to ignore all zone operations,
- Else if the device's write zone (or read zone) is configured to be a zone number in the user range 00h-7Fh or the manufacturer specific range 80h-BFh then it will respond to any Zone Write (or Zone Read) operation because the Active Write Zone (or Active Read Zone) is set to the All Zone (FFh).

Now let's assume that:

- The master used the ZONE_ACTIVE command to set the Active Write Zone (or Active Read Zone) to a particular value in the user range 00h-7Fh or the manufacturer specific range 80h-BFh and
- The master then initiates a ZONE Write (or Zone Read) operation.

Then:

- If the device's write zone (or read zone) is set to the No Zone (FEh), it will ignore all Zone Write (or Zone Read) operations,
- Else if the Active Write Zone (or Active Read Zone) matches the device's assigned write zone (or read zone), then the device will respond to the Zone Write (or Zone Read) operation,
- Else if the device's configured write zone (or read zone) does not match the Active Write Zone (or Active Read Zone), then it will not respond to a Zone Write (or Zone Read) operation.

This implicitly requires that a zone-capable device keep in its memory the Active Write Zone and Active Read Zone values set by the ZONE_ACTIVE command plus the write zone and read zone values to which the device was assigned by the ZONE_CONFIG command.

This gives great flexibility to the system designer. In most cases the system will want to control the power system zone by zone. In an emergency, however, the system could set the active write zone to the All Zone and send a turn off command to all devices in one operation. This is much faster, and safer, than having to send a turn off command, one by one, to each device in the system.

There is a further implication, that the process for configuring the devices in a system to particular zones and the process for setting the Active Write Zone and Active Read Zone must be reliable. When the master communicates with devices in the active zone, there is no way for an individual slave device to NACK the transaction for the whole bus. A board management controller (BMC) might deal with this problem by running some code at boot time to ensure it has repeatable communication with all devices on the bus using read/write of benign registers. It could run this at a faster bus clock than the final clock to add some margin to the test. Generally, a well-characterized design has no bus timing problems. You must also know if any of the slave devices have the ability to be busy and NACK commands. Well-designed slave devices will never NACK a ZONE_CONFIG command. If it cares, the master must ensure it does not change the active zone while a slave device is busy.

5. Zone Read: A Solution For A Fast, Prioritized Response To A Query Of Any Data

The Zone Read operation allows the system designer to achieve faster communication for time-critical conditions. These conditions include:

- Slave device address discovery of large systems
- Priority-based fault reporting and fault handling
- Fast telemetry – quickly retrieving current status (voltage, current, temperature, etc.) on all slave devices in a system
- Priority-based telemetry – quickly performing a highest- or lowest-value read-back from all PMBus slave devices

Some of these functions, like address discovery, use the All Zone. Others like telemetry may focus on smaller zones, or may monitor on the All Zone and follow up with queries on a smaller zone.

5.1. ZONE_READ Command Details

A Zone Read operation uses the reserved Zone Read address (28h) from the SMBus 3.0 specification. Like the Zone Write address, the Zone Read address is reserved to allow a global operation.

Sending a ZONE_READ command to the Zone Read address (28h) creates a request for a response from all or some of the devices on the bus. In this case, the expression “all or some” is not referring to the All Zone vs. only the Active Zone. It is referring to two basic behaviors:

- The Zone Read allows all slave devices in a zone to respond and provide data to the master
- The Zone Read allows one slave device in a zone to win arbitration and provide its data to the master

Each of these has a different purpose:

- To get bulk data from a whole zone
- To get the most important information from a zone

Fetching telemetry falls into the first Zone Read purpose. The goal is to get some value from many devices as fast as possible. Monitoring status falls into the latter purpose. The goal is to know when slave device is in trouble and needs attention. A special case for getting bulk data is early termination. If the master can process data on the fly as the slave device is sending, it can abort the long transaction with a STOP condition and start a new transaction.

A side benefit of reading from multiple slave devices is any slave device on the bus can also monitor the data that each device sends in response. This effectively provides a means for device-to-device communication. The listening devices can act on the information, if designed to do so. However, this means the slave devices depend on the master to make the proper Zone Reads and the slave device cannot on its own initiate slave device-to-slave device communication. This benefit is most likely to be used in systems where the designer is in control of the firmware of the master and slave devices.

Three distinct command transactions must occur for a Zone Read operation.

1. Each slave device must be assigned to a zone using the PMBus command ZONE_CONFIG. This command is expected to be performed on each device during system configuration.
2. The active zone must be set using the ZONE_ACTIVE command.
3. A ZONE_READ command is a broadcast to all zone capable devices requesting data using the Zone Read address, plus a Command Control Code, followed by reading data from one or more slave devices. All devices configured with an assigned Read Zone matching the Active Read Zone setting (from the second transaction above) will respond with the requested data.

Each slave device will respond with its address and page to identify itself and with the requested data. The slave devices' responses will be arbitrated by "bit dominance arbitration" enabled by the open-drain topology of SMBus as described in Section 5.3.2, *Arbitration*, of the SMBus 3.0 specification [R03]. If a device has pages, it will arbitrate the order of data response from those pages for the Zone Read. The polarity and order of bytes of the slave device responses is customizable by the Command Control Code to achieve the desired priority of the responses.

"Bit Dominance" is just a fancy way to describe the effects of a wired AND bus. As each slave device puts data on the bus bit by bit, a zero wins against a one, because a zero means some slave device pulled the data line low. Each slave device monitors the data line to check if the value on the bus is the same as the value it is sending. If the slave device reads back a level that is different than what it expected, it lost the arbitration. This capability has always been part of SMBus and PMBus and is being further exploited here by PMBus as part of the Zone Read operation.

Customization of the device's data is important because when a query has multiple slave devices answering, it controls the order that slave devices win arbitration, such as largest or smallest value first. When only one slave device will answer, it will determine which slave device wins, such as the most critical fault in the zone. It is the combination of arbitration and customization of the slave device's data that gives Zone Read its power.

NOTE: Devices that implement Zone Read are required to support Repeated Start and Clock Stretching SMBus functions. Refer to the SMBus specification for details.

There are several PMBus commands that are prohibited by the specification to be read during Zone Read operations, specifically: PAGE, PAGE_PLUS_WRITE, and PAGE_PLUS_READ.

Also, PMBus commands defined as Send Byte format, which have no data (see Section 6.5.2 of the SMBus 3.0 specification [R03]) shall not be requested during Zone Read operations. An example would be the CLEAR_FAULTS command (i.e. PMBus, Part II, Section 15.1). Devices that support Zone Read operations shall always NACK these command operations.

5.2. The Command Control Code

The Command Control Code allows the master to control the operation of the Zone Read command. The bits in the Command Control Code data byte are listed in Table 3.

Table 3. Command Control Code Bit Definitions

Bits	Mnemonic	Description
7	AR	All Respond
6	ST	Status
5	DI	Data Bit Inversion
4	DS	Data Byte Swap
3:0	–	Reserved

5.2.1. All Respond (AR) Bit

The All Respond (AR) bit tells the slave devices whether or not they should continue to retry their responses when they lose in arbitration and do not get to send their data. Remember the two cases above where all slave devices respond or a single slave device responds (Section 2.3)? This bit determines which mode is in operation.

- When AR = 1, all slave devices keep trying to put their data on the bus until they all succeed or the master ends the transaction with a STOP condition.
- When AR = 0, all slave devices try once to put their data on the bus. This means that only one slave device succeeds in sending its data, based on it winning arbitration.

5.2.2. Status (ST) Bit

The Status (ST) bit tells the slave devices to respond with a status byte or to listen for a specific PMBus command. This is a mode that is orthogonal to AR. This bit was added because requesting status is a very common transaction and this bit allows some optimization of the status reporting.

- When ST = 1, instead of sending a command to the slave devices, all slave devices return status information. Depending on the setting of the Data Byte Swap (DS) bit (Section 5.2.4) this status information could be the STATUS_BYTE (low byte of STATUS_WORD) or the high byte of STATUS_WORD.
- When ST = 0, the device will be receiving a PMBus command.

5.2.3. Data Bit Inversion (DI) Bit

If the Data Bit Inversion bit is set (DI = 1), the slaves are instructed to bit-wise invert the returned data to affect the priority of the arbitration. This allows the master to choose between largest or smallest numbers to either determine the order the master gets data when AR = 1, or which single slave device wins arbitration when AR = 0.

If the Data Bit Inversion bit is not set (DI = 0), then the data bits are not inverted.

5.2.4. Data Byte Swap (DS) Bit

The Data Byte Swap (DS) bit affects the order of data returned by the devices and it may also affect the priority of the arbitration.

In SMBus transactions with multiple data bytes, the low order byte is returned first and the high order byte is returned last. Many PMBus commands return two bytes of data. In the cases where the master is interested determining, for example, which device has the highest temperature, it is useful to get the high order byte first. Setting the DS bit instructs the slave to return the high order data byte first (contrary to the usual SMBus practice of returning the low order byte first). If the DS byte is cleared (= 0), the data is returned in usual SMBus low byte first order.

Of particular interest is when status information is being returned (STATUS (ST) = 1). If the Data Byte Swap bit is not set (DS = 0), then the low byte of STATUS_WORD, which also is that same as the data for the STATUS_BYTE command, is returned (normal SMBus order of returning the low byte first).

However, it may be that a bit in the high byte of STATUS_WORD, such as the POWER_GOOD# bit, is of the most interest to the master. In this case, setting the Data Byte Swap bit (DS = 1) causes the high byte of STATUS_WORD to be returned first.

5.2.5. Using The Command Control Code

The combination of these bits gives the master great flexibility

	7	1	1	8	1	8	1	
S	ZONE READ Address (28h)	Wr	A	COMMAND CONTROL CODE	A	STATUS MASK or COMMAND	A	…

Figure 4: Preamble Of A Zone Read

Figure 4 shows the preamble of a Zone Read operation. It begins with the Zone Read address (28h). Following is the Command Control Code. The third byte is either a status mask or command. When ST = 1, byte 3 is a mask. The mask allows the master to ignore bits. This is important if the master wants to focus on a specific bit or bits downstream without previous bits influencing arbitration. If ST = 0, then a command code would follow the command control code in the sequence.

After the preamble, the master sends a repeated start plus the Zone Read address and begins reading data. The first byte/s from the slave device that wins arbitration is either a single status byte, or one or more data bytes from the command. Following the data is the slave devices' address, and then an optional page if the slave device has a PAGE assigned to a zone. This is illustrated below in Figure 5.

Start The Zone Read Operation With The Preamble:

	7	1	1	8	1	8	1	
S	ZONE READ Address (28h)	Wr	A	COMMAND CONTROL CODE	A	STATUS MASK or COMMAND	A	…

Continue The Zone Read Operation By Getting Data, Address, And PAGE Number
From The First Device To Respond Without Losing Arbitration:

	7	1	1	8	1	7	1	1	7	1	1	
Sr	ZONE READ Address (28h)	Wr	A	STATUS or DATA	A	SLAVE ADDRESS	1	A	SLAVE PAGE NUMBER	1	A	…

Figure 5: Zone Read Transaction: Preamble Plus First Response

The ordering here reflects a purpose. The most important thing to arbitrate is the value of the returned status or data. This is what allows min/max arbitration. If the address/page were first, data would arrive at the master in address order. The address is next because there is always an address. Page is last because it is optional. The address/page are given because when results do not come in a fixed order or only one slave device wins, the master needs to know where the data came from.

There are many combinations that are not shown here, but the above structure will hold in the examples shown below.

5.3. The STATUS_MASK

When the master is doing a Zone Read and requesting status information (ST = 1), the STATUS_MASK allows the master to select just the bit or bits of the status information that is of interest. For example, the master may only be interested in knowing the state of the Power Good Signal of the devices in a system. The STATUS_MASK can be used to cause the slave devices to suppress all status bits other than the one related to the Power Good signal.

5.3.1. How the STATUS_MASK works

Most simply:

- A bit set to 0 in the STATUS_MASK tells the slave to allow that bit to pass unaltered (not masked)
- A bit set to 1 in the STATUS_MASK tells the slave to mask that bit (set it 0)

This can be expressed as the equation:

RETURNED DATA = STATUS & INV(STATUS MASK)

5.3.2. STATUS_MASK examples

Suppose the master is only interested in bit [4] of the byte of status data being returned. It does not care about any of the other bits and in particular does not want bits [7:5] to affect the arbitration of the returned data. Remember that in the status information, one of the two bytes returned by the PMBus STATUS_WORD command, a bit that is set (= 1) indicates that condition is true.

Table 4 shows the use of the STATUS_MASK to get only the state of bit [4] of the status information returned from a slave device. In this case, bit [4] is zero and the value of bit [4] is passed through unaltered. Any other bits in the status information are masked, that is, set to 0.

Table 4. Example Of Using The STATUS_MASK

Description	Bit Number							
	7	6	5	4	3	2	1	0
Status Byte Data From Slave Device	0	1	0	0	1	1	0	0
STATUS_MASK	1	1	1	0	1	1	1	1
STATUS_MASK Bit-Wise Inverted	0	0	0	1	0	0	0	0
Returned Data (Bit-Wise AND Of Data From Slave Device And The Bit-Wise Inverted STATUS_MASK)	0	0	0	0	0	0	0	0

Now suppose in the example above that bit [4] of the status byte from the slave was set (= 1). With the STATUS_MASK set to select bit [4], what data is returned to the master? The result is shown in Table 5.

Table 5. Another Example Of Using The STATUS_MASK

Description	Bit Number							
	7	6	5	4	3	2	1	0
Status Byte Data From Slave Device	0	1	0	1	1	1	0	0
STATUS_MASK	1	1	1	0	1	1	1	1
STATUS_MASK Bit-Wise Inverted	0	0	0	1	0	0	0	0
Returned Data (Bit-Wise AND Of Data From Slave Device And The Bit-Wise Inverted STATUS_MASK)	0	0	0	1	0	0	0	0

Again we see that the value of bit [4] is returned to master without alteration while all other bits are forced to be 0.

6. Zone Write: A Solution For Synchronized Data Execution

The Zone Write operation uses the Zone Write address (37h), from the SMBus 3.0 specification [R03]). A Zone Write operation causes synchronized execution of commands to all or some of the slave devices on the bus. Remember, a zone is a subset of slave devices on a bus but it can include all slave devices on the bus.

Zone Write has three transactions associated with it:

1. Each slave device must be assigned to a zone using the ZONE_CONFIG command.
2. The Active Write Zone must be set using the ZONE_ACTIVE command.
3. A Zone Write operation is the broadcast of a PMBus command and its data to the Zone Write address. All devices configured with a write zone value matching the Active Write Zone setting will act upon the written command and data when the STOP condition occurs.

This means the time alignment of execution is controlled and limited by the internal delays of the slave devices as the slave device is not allowed to begin processing before the STOP condition is received. A slave device may prepare to execute early, but it must not begin to execute the command until the STOP condition is detected.

There are several PMBus commands that are prohibited by the specification to be written during Zone Write operations, specifically: PAGE, PAGE_PLUS_READ, and ZONE_CONFIG. Devices that support Zone Write operations shall always NACK command operations to PAGE, PAGE_PLUS_READ, and the ZONE_CONFIG command.

	7	1	1	8	1	8	1	
S	ZONE WRITE Address (37h)	Wr	A	OPERATION (01h)	A	DATA (80h)	A	P

Figure 6. Zone Write With OPERATION Command Example

Figure 6 shows and example of a Zone Write. The command is issued to address 37h, which is the global address used to write to the Active Write Zone. The OPERATION command is the second byte, followed by the data.

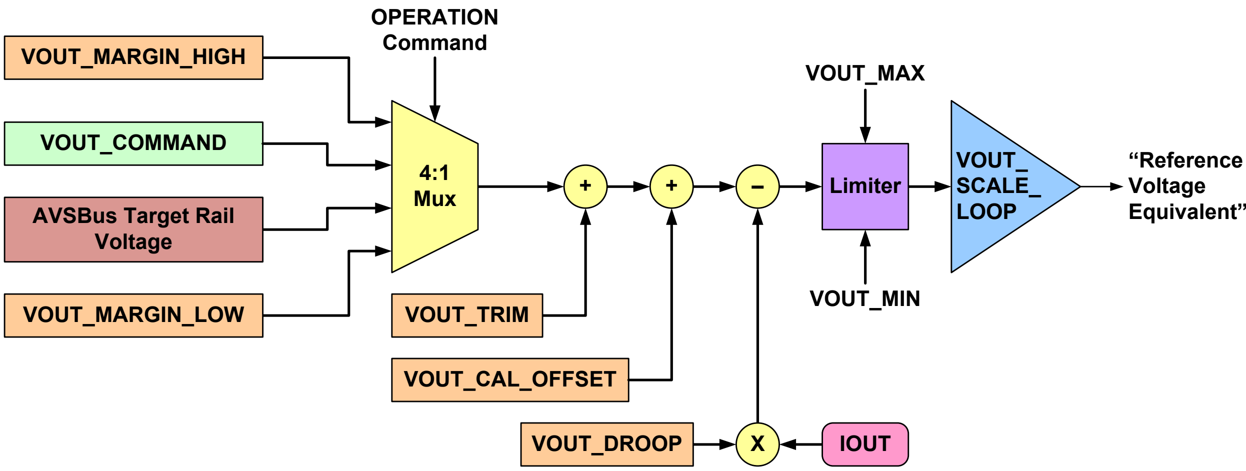

Figure 7. Conceptual View Of How Output Voltage Related Commands Are Applied

Figure 7, taken from Part II of the PMBus specification [R01], shows that the OPERATION command chooses between MARGIN, VOUT, or none of them. The data value 80h means apply VOUT_COMMAND, or turn on the output. In this case, all devices in the active write zone will turn on, right after the STOP.

7. Example System

Below are several examples that illustrate the details of using the Zone Write and Zone Read operations. For these examples the system shown in Figure 8 is used.

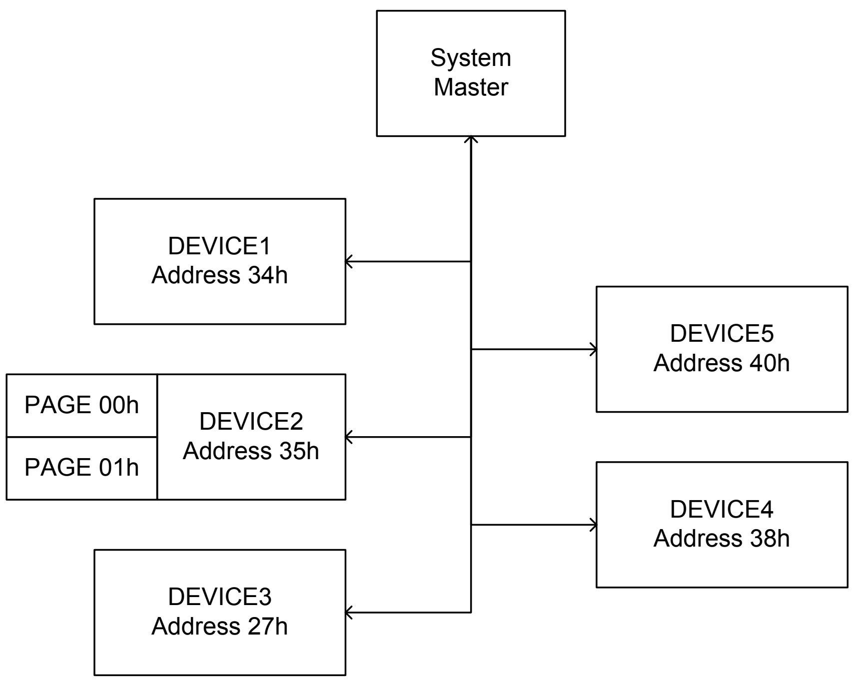

Figure 8: Example System Configuration

Many of the examples will assume that some devices in the system are reporting warning and faults as shown in Figure 9. The current operating condition of each device in the example system is shown in Table 6.

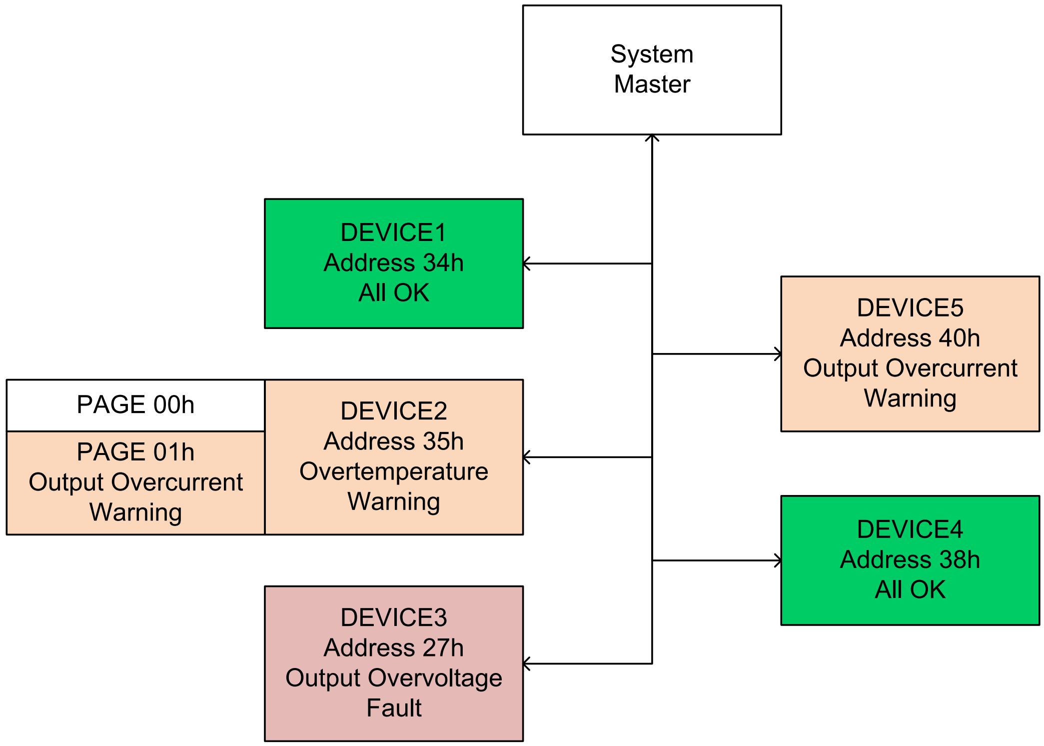

Figure 9: Example PMBus System With Warning And Fault Conditions

Table 6. Operating Condition Of Devices In Example System

Device/ Address/ Page	Condition	Temperature		Current	
		°C	LINEAR11 Format	Amperes	LINEAR11 Format
1 (34h)	Operating Normally	55	E370h	18	DA40h
2 (35h) Global Page FFh	Overtemperature Warning	95	EAF8h	N/A	N/A
2 (35h) Page 00	Overtemperature Warning	95	EAF8h	24	DB00h
2 (35h) Page 01	Overtemperature Warning Output Overcurrent Warning	95	EAF8h	28	F3E0h
3 (27h)	Output Overvoltage Fault Power Good Negated	25	DB20h	0	0000h
4 (38h)	Operating Normally	48	E300h	12	D300h
5 (40h)	Output Overcurrent Warning	75	EA58h	22	DAC0h

The condition of DEVICE2 merits some discussion. In this example, this device has only one temperature sensor for the whole device. In this sense, temperature and any overtemperature warnings or faults are “global” to that device. That is, if the master were to send the READ_TEMPERATURE_1 command to DEVICE2, it gets back the same value regardless of whether the page number is set to 00h, 01h, of FFh. While this is valid for this example, you must check the data sheet of any device with pages to understand how that device handles and responds to data that is “global” within that device.

If the master were to send a STATUS_WORD command to each of the devices in the system, it would get back the data shown in Table 7. Figure 10 shows the data bits returned for the PMBus STATUS_WORD command.

Table 7. STATUS_WORD Registers Of The Example System With Faults And Warnings

DEVICE/ADDRESS/PAGE	STATUS_WORD															
	High Byte								Low Byte							
	VOUT	IOUT/POUT	INPUT	MFR_SPECIFIC	POWER_GOOD Negated	FAN	OTHER	UNKNOWN	BUSY	OFF	VOUT_OV_FAULT	IOUT_OC_FAULT	VI_UV_FAULT	TEMPERATURE	CML	NONE OF THE ABOVE
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1 (34h)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2 (35h) Page FFh	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2 (35h) Page 00	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2 (35h) Page 01	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
3 (27h)	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
4 (38h)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5 (40h)	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

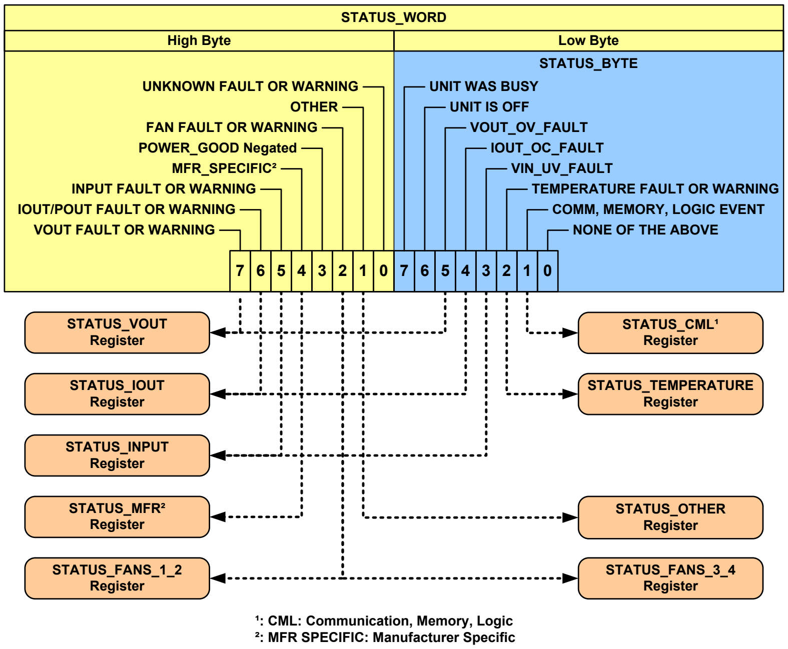

Figure 10. STATUS_WORD Data Bits

Unless specified otherwise, all examples in this section assume that the Active Write Zone Active Read Zone are set to the All Zone (FFh) so that every device in the system responds to all Zone Writer and Zone Read operations. This is done to focus the examples on the Zone Write and Zone Read behavior rather being distracted by tests of whether a device's assigned read zone matches the Active Read Zone.

Also, unless specified otherwise, all numeric data is in the LINEAR11 format.

8. READ_ZONE Command Examples

The Zone Read operation can be used for many different purposes, such as:

- Slave Device Address Discovery
- Priority-Based Fault Reporting and Fault Handling
- Highest Data Value Request
- Fast Telemetry
- Priority-Based Telemetry
- Reconfigurable Sequencing Based On The Status Of Power Good Signals

8.1. Discovering Zone Active Devices In A System

Device address discovery is a feature that many system designers need when the number of devices on the bus can differ for each system configuration. Modular systems can have a different number of devices on the bus for different assembly-builds. This situation occurs, for example, when a system consists of optional cards.

The PMBus 1.2 specification had no provision for device discovery, so engineers devised all kinds of tricks to figure out what slave devices are on the bus, including which devices were PMBus and which devices where SMBus or I2C. A Zone Read operation can solve this problem nicely.

As an example, suppose a system is powered up with the devices shown in Figure 8 but the master has no knowledge of any installed zone capable devices. To discover the available devices, the master uses the ZONE_READ command to get all of the address and page numbers of all devices in one transaction – without scanning every address! Because the Zone Read address is a reserved address by virtue of the SMBus Specification, non-PMBus slave devices ignore the transaction, reducing a lot of ad hoc complexity from the system design.

It is important to note that the discovery process should be done at the first initialization of the system and before any other commands are sent to the power system.

Before starting discovery, the master sets the Active Read Zone to the All Zone to assure that all zone capable devices will respond to the coming ZONE_READ command.

Then the master sends the ZONE_READ command with the Command Control Code configured so that all devices will respond with their status. Specifically the Command Control Code bits are:

- Bit [7]: All Respond (AR) = 1
- Bit [6]: Status (ST) = 1
- Bit [5]: Data Bit Inversion (DI) = 0
- Bit [4]: Data Byte Order Swap (DS) = 0
- Bits [3:0]: Reserved = 0000

The resulting Command Control Code data byte is 11000000b or C0h.

Since the Command Control Code is calling for Status information, the Status Mask must be configured. In this case, all status bits will be allowed to pass

- STATUS_MASK = 00000000b = 00h

Basically, this is a command to fetch the status of all devices. Zones would typically be setup after discovery.

In this example, slave devices at address 27h, 34h, 35h, 38h and 40h exist on the bus and these addresses are returned. However, the slave device at address 35h has two pages, and both are returned. So in fact, discovery includes the numbers of all pages of the slave device.

The bus transactions to discover the devices in the system shown above in Figure 8 are shown in Figure 11.

For this to work, the master has to start the Zone Read operation and then keep reading until a NACK of the zone address. During each read from each slave device, the first data byte includes the device's address in bits [7:1]. Bit [0] is the Page flag. If bit [0] is 0, that means that the device has no pages and no further data is to be expected. If bit [0] is 1, that means that the device has pages and the next data byte will be the number of responding page.

8.2. Priority-Based Fault Reporting And Fault Handling

Fault handling requires fast communication, and should be dealt with quickly at the system level to minimize circuit damage and maximize system availability. With PMBus 1.2, slave devices only have the open-drain SMBALRT# pin to tell the master that there is a problem. The master will not know which device(s) pulled the SMBALRT# pin low; it will only get an interrupt. It must therefore poll every slave device to query its status, or use Alert Response Address (ARA) to determine which slave device alerted and then query its status. Only then can the master act on the information that it has received.

Start The Discovery Process By Setting The Active Read Zone To The All Zone (FFh)

	7	1	1	8	1	8	1	8	1	
S	ZONE WRITE Address (37h)	Wr	A	ZONE_ACTIVE (08h)	A	Active Write Zone (FFh)	A	Active Read Zone (FFh)	A	P

Use The ZONE_READ Command To Get The Address, Page Number, And Status Of All Zone Capable Devices

	7	1	1	8	1	8	1	
S	ZONE READ Address (28h)	Wr	A	COMMAND CONTROL CODE (C0h)	A	STATUS MASK (FFh)	A	…

	7	1	1	8	1	7	1	1	
Sr	ZONE READ Address (28h)	R	A	STATUS_WORD[15:8] (00h)	A	SLAVE ADDRESS (27h)	0	A	…

	7	1	1	8	1	7	1	1	
Sr	ZONE READ Address (28h)	R	A	STATUS_WORD[15:8] (00h)	A	SLAVE ADDRESS (34h)	0	A	…

	7	1	1	8	1	7	1	1	8	1	
Sr	ZONE READ Address (28h)	R	A	STATUS_WORD[15:8] (00h)	A	SLAVE ADDRESS (35h)	1	A	SLAVE PAGE NUMBER (00h)	A	…

	7	1	1	8	1	7	1	1	8	1	
Sr	ZONE READ Address (28h)	R	A	STATUS_WORD[15:8] (00h)	A	SLAVE ADDRESS (35h)	1	A	SLAVE PAGE NUMBER (01h)	A	…

	7	1	1	8	1	7	1	1	
Sr	ZONE READ Address (28h)	R	A	STATUS_WORD[15:8] (00h)	A	SLAVE ADDRESS (38h)	0	A	…

	7	1	1	8	1	7	1	1	
Sr	ZONE READ Address (28h)	R	A	STATUS_WORD[15:8] (00h)	A	SLAVE ADDRESS (40h)	0	A	…

	7	1	1	
Sr	ZONE READ Address (28h)	R	N	P

Figure 11: Example Of Bus Transactions During Device Address Discovery

The master could retrieve this information by sending a STATUS_WORD command to each device. Depending on how the master treats DEVICE2, this could be six or seven STATUS_WORD command operations.

A Zone Read operation offers another way to query all the fault information in one transaction. The trick is simple: the master reads status from a zone in such a way that slave devices with faults return their values first, and when the master sees a slave device without faults, it terminates the transaction.

Suppose the master sends a ZONE_READ command with the Command Control Code bits configured:

- All Respond (AR) = 1, All devices will send data
- Status (ST) = 1, The master is requesting status information rather than sending a PMBus command
- Data Bit Inversion (DI) = 1, The slaves are instructed to bit-wise invert the returned data
- Data Byte Swap (DS) = 1, The slaves are instructed to send the high byte of the STATUS_WORD command
- Reserved bits are all set to 0

The resulting Command Control Code data byte is 11110000b (F0h).

The master wants to see all of the data bits so none are masked. The Status Mask data byte is set to 00000000b (00h).

DEVICE3, at address 27h, calculates the return data byte as shown in Table 8. The other devices calculate their return data in the same way.

Table 8. Example Of Returned Data Calculation

Description	Bit Number							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Upper byte of the STATUS_WORD (from Table 7)	1	0	0	0	1	0	0	0
Because the Data Inversion (DI) bit is set, each bit of the data is inverted	0	1	1	1	0	1	1	1
STATUS_MASK	0	0	0	0	0	0	0	0
STATUS_MASK Bit-Wise Inverted	1	1	1	0	1	1	1	1
Returned Data (Bit-Wise AND Of The Inverted Data From The Slave Device And The Bit-Wise Inverted STATUS_MASK)	0	1	1	1	0	1	1	1

The resulting Zone Read operation is illustrated in Figure 12.

If you compare this to using the Alert Response Address (ARA), it is similar except for two things:

- Status comes with each poll
- Status comes in order of worst fault first

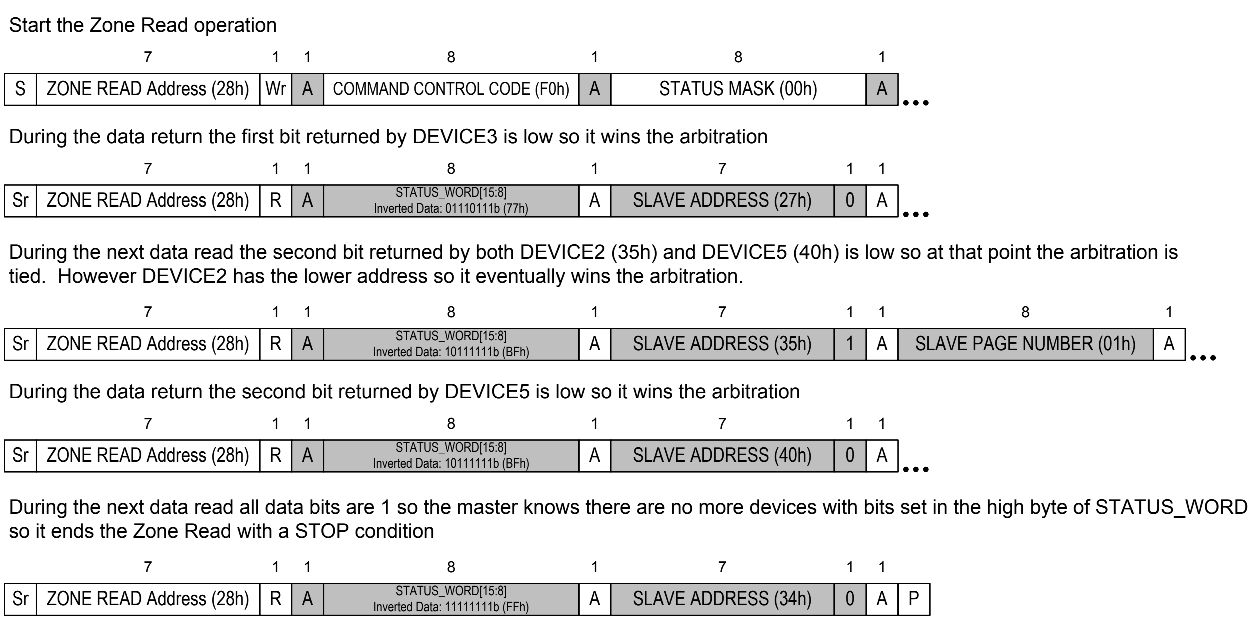

Figure 12. Zone Read Operation Returning Data Based On Priority

In the high byte of the STATUS_WORD data, VOUT fault or warning bit comes before the IOUT/POUT warning or fault bit, which in turn is the INPUT fault or warning bit, etc. So if there is a problem with VOUT, the master can issue a ZONE_WRITE command with the OPERATION command configured to turn off the whole system. An ARA cannot return fault information in this order, and instead returns in the order of address. This is because there is arbitration where the lower (smaller value) address wins over the higher (larger value). Data in a Zone Read operation arrives before address, which makes it possible to order the most important information first in the transaction.

A master has the choice during the Zone Read operation to terminate at any time and act. A master can also do this with ARA, but it takes an added transaction, so it is less efficient.

Notice that this choice of returning the high byte of the PMBus STATUS_WORD command does detect conditions that most system engineers would call high priority, such the condition of a Power Good status signal. However, the data returned with this choice of returning status information (ST = 1) does not reveal to the master the overtemperature condition of DEVICE5.

If it was important to the master to know about any bits in the STATUS_WORD data that have been set, it could use a Zone Read operation with the PMBus command STATUS_WORD. To have only devices that have at least one bit set, the master would instruct the slaves to invert the bits of the returned data (DI = 1). For this example, we assume that the master does not care about which byte of the STATUS_WORD data is returned first, so it does not set the Date Byte Swap bit (DS = 0). The Command Control code is generated as:

- All Respond (AR) = 1, All devices will send data
- Status (ST) = 0, The master is sending a PMBus command
- Data Bit Inversion (DI) = 1, The slaves are instructed to bit-wise invert the returned data
- Data Byte Swap (DS) = 0, The slaves are instructed to send the data in the ordinary SMBus low first order.
- Reserved bits are all set to 0

The resulting Command Control Code data byte is 10100000b (A0h).

The PMBus command code for STATUS_WORD is 79h.

Assuming the master ends the Zone Read with a STOP condition as soon as it see that no more devices have any of the STATUS_WORD bits set, the resulting Zone Read transaction is shown in Figure 13.

8.3. Highest Data Value Request

A Zone Read operation can be used to quickly receive a single, highest or lowest value of data for a telemetry read command. For example, a Zone Read operation can be used to find the device, and thus the area of the board, which has the highest temperature. The Command Control Code bits are set as follows:

- All Respond (AR) = 0, Only one device will respond
- Status (ST) = 0, A PMBus command will be sent next
- Data Inversion Bit (DI) = 1, Invert the data so that be bits that are high in the data become lows on the bus, assuring that they will win the arbitration,
- Data Byte Swap (DS) = 1, Return the high order data byte first

The resulting Command Control Code is 0011000b or 30h.

Start the Zone Read operation with the PMBus command STATUS_WORD

	7	1	1	8	1	8	1	
S	ZONE READ Address (28h)	Wr	A	COMMAND CONTROL CODE (A0h)	A	STATUS_WORD (79h)	A	…

During the data return the first low bit is returned by DEVICE3 is low so it wins the arbitration

	7	1	1	8	1	8	1	7	1	1	
Sr	ZONE READ Address (28h)	R	A	STATUS_WORD[7:0] Inverted Data: 11011111b (D7h)	A	STATUS_WORD[15:8] Inverted Data: 01110111b (77h)	A	SLAVE ADDRESS (27h)	0	A	…

During the next data read the data returned by Page 1 of DEVICE2 wins the arbitration

	7	1	1	8	1	8	1
Sr	ZONE READ Address (28h)	R	A	STATUS_WORD[7:08] Inverted Data: 11111011b (FDh)	A	STATUS_WORD[15:8] Inverted Data: 10111111b (BFh)	A

7	1	1	8	1	
SLAVE ADDRESS (35h)	1	A	SLAVE PAGE NUMBER (01h)	A	…

During the next data read the data returned by Page 0 of DEVICE2 wins the arbitration

	7	1	1	8	1	8	1
Sr	ZONE READ Address (28h)	R	A	STATUS_WORD[7:08] Inverted Data: 11111011b (FDh)	A	STATUS_WORD[15:8] Inverted Data: 11111111b (FFh)	A

7	1	1	8	1	
SLAVE ADDRESS (35h)	1	A	SLAVE PAGE NUMBER (00h)	A	…

During the next data return the data returned by DEVICE5 wins the arbitration

	7	1	1	8	1	8	1	7	1	1	
Sr	ZONE READ Address (28h)	R	A	STATUS_WORD[7:0] Inverted Data: 11111111b (FFh)	A	STATUS_WORD[15:8] Inverted Data: 10111111b (BFh)	A	SLAVE ADDRESS (40h)	0	A	…

During the next data read, all data bits are 1 so the master knows there are no more devices with bits set in the high byte of STATUS_WORD so it ends the Zone Read with a STOP condition. Note that DEVICE1 with address 34h wins the arbitration.

	7	1	1	8	1	8	1	7	1	1	
Sr	ZONE READ Address (28h)	R	A	STATUS_WORD[7:08] Inverted Data: 11111111b (FFh)	A	STATUS_WORD[15:8] Inverted Data: 11111111b (FFh)	A	SLAVE ADDRESS (34h)	0	A	P

Figure 13. Read Zone With The PMBus STATUS_WORD Command

The reason Data Byte Inversion is used (DI=1) is that the example is expecting the returned data to be in the LINEAR11 format. The largest data value will contain the smallest exponent (largest negative number in 2's complement). Thus inverting the data will allow the device with the largest number to win the arbitration. The Data Byte Swap bit is set to ensure the highest data byte is analyzed first.

Then a READ_TEMPERATURE_1 command would be issued. Figure 14 illustrates a Zone Read used to receive the highest temperature data from a system. A high temperature of 95 °C from page 00h of DEVICE2 is reported. Please note the discussion in the previous example about how this device reports temperature.

	7	1	1	8	1	8	1	
S	ZONE READ Address (28h)	Wr	A	COMMAND CONTROL CODE (30h)	A	READ_TEMPERATURE_1 PMBUS COMMAND CODE (8Dh)	A	…

	7	1	1	8	1	8	1	
Sr	ZONE READ Address (28h)	R	A	READ_TEMPERATURE_1[15:8] (EAh)	A	READ_TEMPERATURE_1[7:0] (F8h)	A	…

7	1	1	8	1	
SLAVE ADDRESS (35h)	1	A	SLAVE PAGE NUMBER (00h)	A	P

Figure 14: Highest Data Value Request

8.4. Fast Telemetry

Many applications require access to fast telemetry. PMBus 1.2 implements telemetry by using multiple reads, which involves redundant information, in that the command must be issued for each slave device being queried. The Zone Read operation can speed this up by removing redundant data from the transaction by putting the whole telemetry request into one Zone Read operation.

Figure 15 illustrates a Zone Read operation used to retrieve the output current of every device on the bus. This is the system's telemetry data.

The Command Control Code bits are:

- All Respond (AR) = 1, All devices are to respond
- Status (ST) = 0, The master is sending a PMBus command
- Data Inversion Bit (DI) = 0, Returned data bits are not inverted
- Data Byte Swap (DS) = 0, The data bytes are returned in the normal SMBus manner (low byte first)

The resulting Command Control Code data byte is 10000000b or 80h.

The order of the reporting devices depends on the arbitration of the low data byte of the telemetry data, since low data byte first is the SMBus norm.

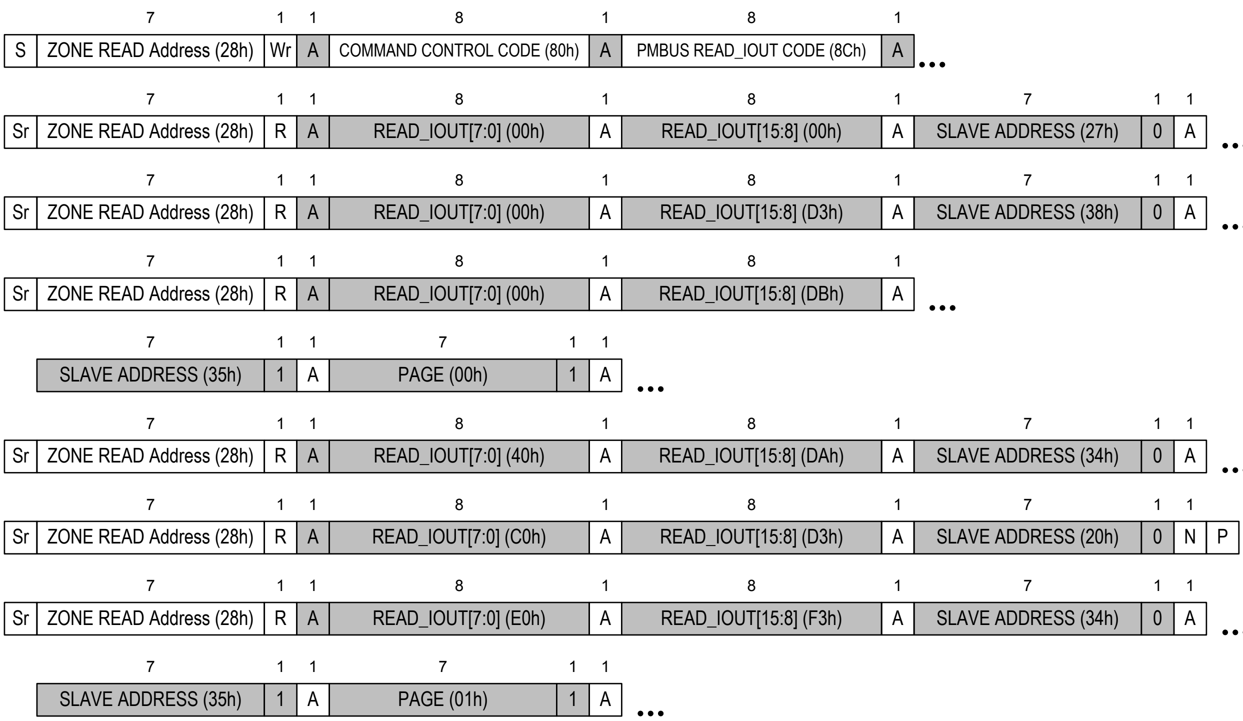

Figure 15: Fast Telemetry Using READ_IOUT Command

8.5. Priority-Based Telemetry

If the order and priority of the responded data needs to be controlled, the Control Command Code can be used to do that.

Figure 16 shows an example of receiving output current data in "highest value first" order. The Command Control Code bits are:

- All Respond (AR) = 1, All devices respond
- Status (ST) = 0, Master is sending a PMBus command
- Data Bit Inversion (DI) = 1, Cause the highest value to win the arbitration
- Data Byte Swap (DS) = 1, Return the high byte first

The resulting Command Control Code data byte is 10110000b or B0h.

The PMBus command code for READ_IOUT is 8Ch.

Setting the Data Byte Swap bit gets the high byte first and setting the Data Bit Inversion bit causes the highest value to win arbitration. This process performs a highest- or lowest-value read-back 11 times faster than the standard PMBus command method for a bus with 15 devices on it.

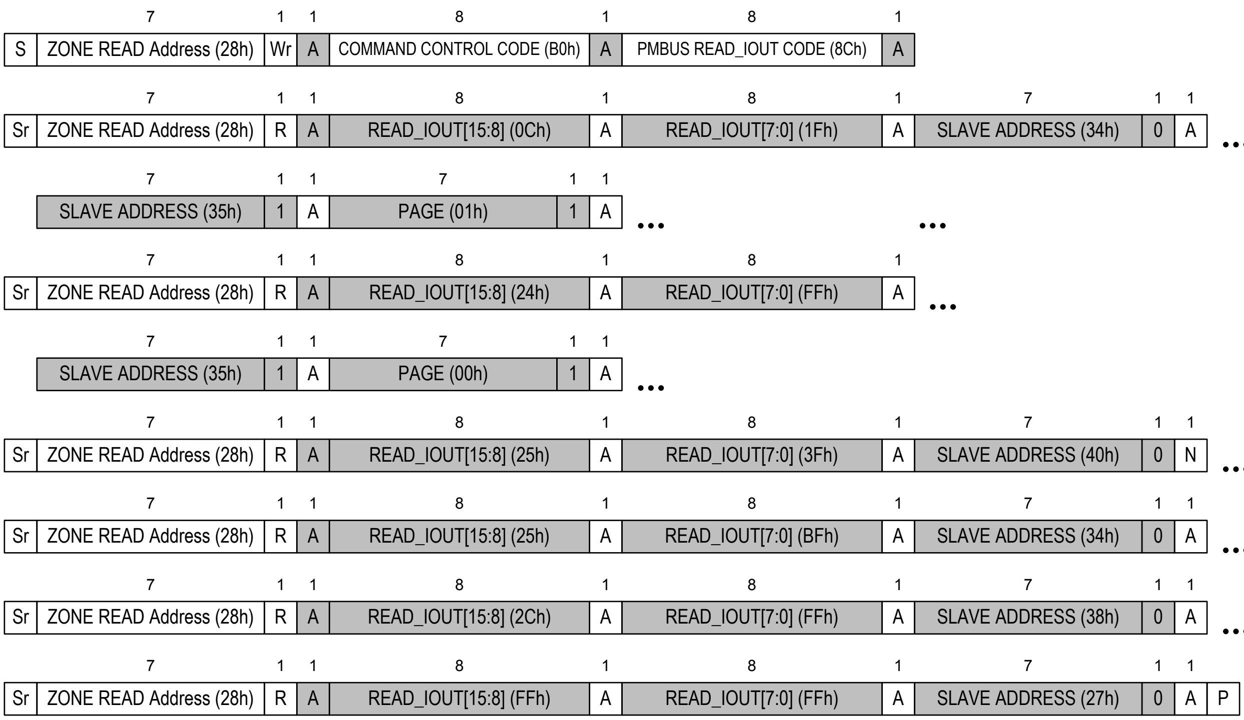

Figure 16: Priority-Based Telemetry Using READ_IOUT Command

8.6. Reconfigurable Sequencing Based On The Power Good Signal

A device starting up in a specific sequence based on another device's power-good status is called event-based sequencing. Event-based sequencing requires the system to have some form of device to device communication to report and act on order-of-sequence information. In many existing implementations, event based sequencing is implemented in hardware by daisy-chaining the power good output signal of one converter to the enable input signal of another, as shown in Figure 17. This method has obvious hardware limitations and does not lend itself to easy resequencing the devices if required.

Another method is to have the device power good signal pins individually monitored by the host controller and have the host controller drive the enable input of the next device in the sequence. This method has complications regarding the use of many discrete traces routed to each device from the controller. A third method could utilize SMBus by having a device set its SMBALRT# pin then wait to be polled by the host controller. The host controller must learn which device alerted and learn that device's power-good status, then communicate to the next device in the sequence to start up. In real applications, a sequence order may require a millisecond device-to-device startup time. The current SMBus transport mechanism does not allow fast enough transactions to achieve this requirement. In addition, the user does not want the added complexity of implementing SMBus multi-mastering to enable two-way communication over the existing bus.

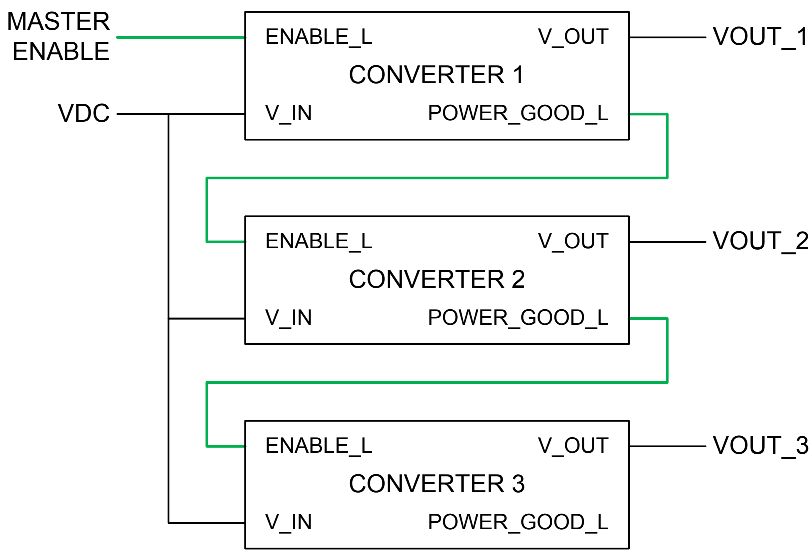

Figure 17. Hardware Implemented Event Based Sequencing

A Zone Read operation allows a slave device to report the value of its power good signal through the POWER_GOOD# bit in the STATUS_WORD data (see Figure 10). Note that in the PMBus protocol, if the POWER_GOOD# bit is asserted, this means that the power good signal is indicating that the output power is not valid. You can think of the hash symbol (#) in the POWER_GOOD# bit name as meaning "not". Thus POWER_GOOD# asserted (bit value equals 1) means "power not good".

During this status mode Zone Read operation, each device on the bus could also listen for a positive power good response from specific address in order to enable itself, thus achieving another method for event based sequencing. The master can continue sending ZONE_READ commands until all devices have returned a successful power good status. This event based sequencing method allows the enable sequence to be reconfigurable.

Referring to our example system of devices, suppose that a requirement has been established that Page 0 of DEVICE2 (address 35h) must come up before DEVICE1 (address 34h). DEVICE1 has been configured to listen for the page 0 of DEVICE2 to report a positive power good during a Zone Read of device status. When DEVICE2 detects this power good status, it is to enable itself. Figure 18 shows the first transaction, setting the Active Read Zone, to achieve this inter-device communication. Note that there are only two devices with assigned read zone values of 03h, page 0 and page 1 of DEVICE2 (Table 1).

	7	1	1	8	1	8	1	8	1	
S	ZONE WRITE Address (37h)	Wr	A	ZONE_ACTIVE (08h)	A	Active Write Zone (FFh)	A	Active Read Zone (03h)	A	P

Figure 18: The ZONE_ACTIVE Command Sets The Zone Read Zone To 03h

For the Zone Read operation, the Control Command Code is:

- All Respond (AR) = 1 to have all devices respond
- Status (ST) = 1 to instruct the devices being read to return status information
- Data Inversion Bit (DI) = 1 to invert the data
- Data Byte Swap Bit (DS) =1 so that the high byte of STATUS_WORD, containing the POWER_GOOD# bit, is returned instead of the low byte

The resulting Command Control Code is 11110000b = F0h.

The Status Mask is set to only allow the POWER_GOOD# bit to be received. The POWER_GOOD# is bit [3] of the upper byte of the STATUS_WORD. Therefore the mask value should be 11110111b (F7h) to select bit [3] of the returned data.

The power up sequence proceeds as follows:

- The master commands DEVICE2 to run on through some combination of the PMBus CONTROL signal and the PMBUS OPERATION command.
- The master then sends a ZONE_READ command with the Command Control Code (F0h) and STATUS_MASK (F7h).

	7	1	1	8	1	8	1	
S	ZONE READ Address (28h)	Wr	A	COMMAND CONTROL CODE (F0h)	A	STATUS MASK (F7h)	A	...

- The first device to respond is page 0 of DEVICE2. Given that its output power is valid, then the POWER_GOOD# bit value is 0. After masking all other bits and passing the POWER_GOOD# bit unaltered, page 0 of DEVICE2 returns 00h to the master

	7	1	1	8	1	7	1	1	8	1	
Sr	ZONE READ Address (28h)	R	A	STATUS_WORD[15:8] (00h)	A	SLAVE ADDRESS (35h)	0	A	SLAVE PAGE NUMBER (01h)	A	...

- DEVICE1 detects that bit [3] of the returned status is 0, meaning that page 0 of DEVICE2 is indicating that its output power is valid. DEVICE1 then powers up.

The master can at this point either continue with the Zone Read to get the status of page 1 of DEVICE2 (the only other device in read zone 3) or it can issue a STOP condition to end the transaction. At this point the master may want to check the status of DEVICE1 to determine if it successfully powered up or not.

Note that there is no standard PMBus command to configure a device to monitor the bus and listen for status information from other converters. It is entirely possible to include this functionality but it would have to be accessed through manufacturer specific commands.

9. ZONE_WRITE Command Examples

The Zone Write operation can be used in multiple ways. Here are a few possible uses of a Zone Write operation:

1. Simultaneously turning on, or turning off, the output of a group of devices,
2. Simultaneously margining the output voltage of a group of devices, and
3. In a system with paralleled ac-dc power supplies, simultaneously changing the speed of fans in all the power supplies.

The key advantage to the Zone Write protocol is that the same command can be sent to multiple devices in one bus transaction (START, zone function address, command, data, STOP) rather than having to send the command to each device, one at time. This can really help reduce the bus traffic in a large system.

If a slave device does not support the command, the Zone Write transaction is not disrupted. However, under the PMBus protocol, the device that does not handle zone commands would handle that as it would any other unsupported command.

9.1. Synchronizing Device Turn On And Turn Off

The PMBus OPERATION command is a good candidate for a Zone Write operation. Refer to Figure 19 below and PMBus Part II [R01], section 12.1, Table 9.

The commands that address the time aspect of turning a device on and off are TON_DELAY, TON_RISE, TOFF_DELAY and TOFF_FALL commands.

Figure 19 shows and example including turn on and turn off waveforms. The problem is how to turn on (or off) multiple slave devices so that all slave devices turn on (or off) at the same time. If this problem is solved, the delays and rise/fall times will take care of themselves.

In the PMBus 1.2 specification, the Group Command Protocol was the solution.

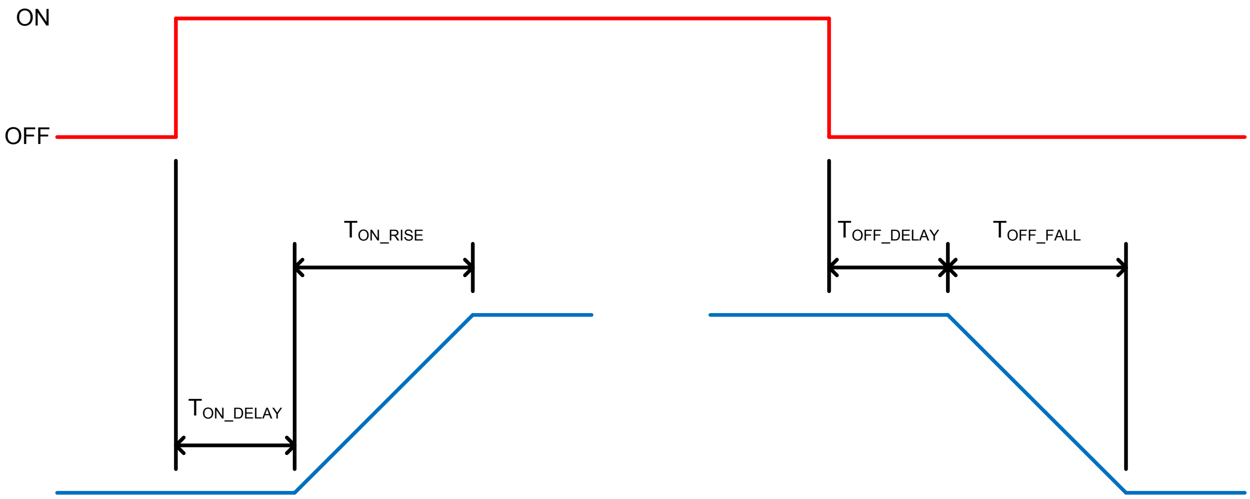

Figure 19. PMBus Power On And Power Off Sequencing Commands

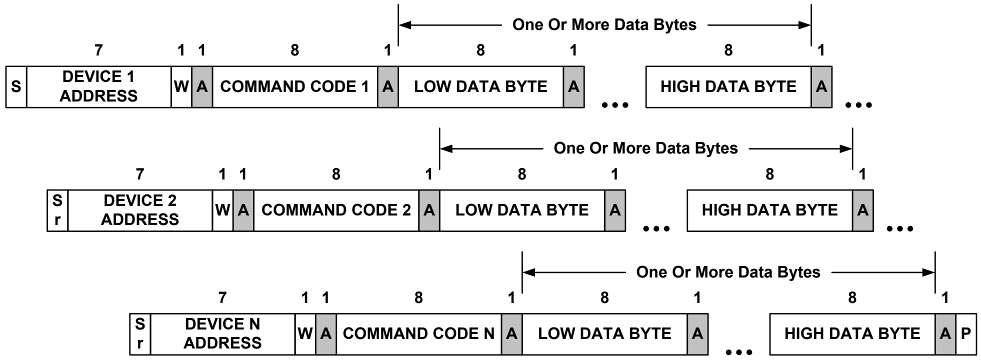

Figure 20. PMBus Group Command Protocol Without PEC

Figure 20 shows that the Group Command Protocol allows the master to send one Command to each device on a long transaction. If the goal is to turn on all devices, this can be done by using the same Command code and the same data, and all the addresses.

However, there is a lot of redundancy and the master has to do a lot of work building up the transaction. This transaction uses up bus and controller bandwidth.

The Zone Write operation can be used to generate the same result with more efficiency. Figure 21 shows an example of a ZONE_WRITE command that includes the PMBus OPERATION command being written to all slave devices on the bus by writing to the All Zone.

	7	1	1	8	1	8	1	8	1	
S	ZONE WRITE Address (37h)	Wr	A	ZONE_ACTIVE (08h)	A	Active Write Zone (FFh)	A	Active Read Zone (0Ah)	A	P

	7	1	1	8	1	8	1	
S	ZONE WRITE Address (37h)	Wr	A	OPERATION (01h)	A	DATA (80h)	A	P

Figure 21: Zone Write Example Of Synchronized Device Turn On

The ZONE_WRITE command synchronizes the execution of written data upon assertion of the STOP at the end of the command, similar to the Group Command Protocol. However, with 7 bytes, all devices on the bus turned on regardless of the number of slave devices. With sixteen devices on a bus, the Group Command Protocol would require 48 bytes to turn them all on. Furthermore, a master could power on a system zone by zone very efficiently, yet power down all zones with one simple transaction.

We can summarize by saying a Zone Write operation reduced this example from a transaction length proportional to the system size to a compact fixed length transaction.

9.2. Simultaneous Output Voltage Margin

When testing a system it is common to increase or decrease the system voltages to assure that all devices will work properly under all conditions. This is called Margin Testing. The PMBus protocol supports Margin Testing by preprogramming high and low margin voltages. The OPERATION command is then used to select whether the output voltage is at the normal setpoint value, the high (upper) margin voltage, or the low (lower) margin voltage.

To cause all devices in the Active Write Zone to set their output voltages to the upper margin voltage, the ZONE_WRITE command including the OPERATION command is sent with the data byte 54h (10011000b, see Table 9, OPERATION Command Data Byte Contents of [R01] for the details of the data byte). This is illustrated in Figure 22.

	7	1	1	8	1	8	1	
S	ZONE WRITE Address (37h)	Wr	A	OPERATION (01h)	A	DATA (54h)	A	P

Figure 22. Zone Write Operation To Margin High All Devices In The Active Write Zone

9.3. Simultaneous Configuration Storage

Another possible use of a Zone Write operation would be, after the devices have been configured by the user, to cause all devices in the Active Write Zone to store configuration in the User Store memory. This is done by sending ZONE_WRITE command with the PMBus STORE_USER_ALL command to the Active Write Zone as shown in Figure 23. Note that there is no data sent for the STORE_USER_ALL command.

	7	1	1	8	1	
S	ZONE WRITE Address (37h)	Wr	A	STORE_USER_ALL (15h)	A	P

Figure 23. Zone Write Operation To Have All Devices In The Active Write Zone Save Their Configuration To The User Store Memory

10. Conclusion

The zone operations bring an important, and powerful, new functionality to PMBus systems. For example, a Read Zone operation can be used to discover all of the devices attached to a PMBus in one transaction – something that previously could require up to 255 bus operations. Read Zone operations also offer improved methods of retrieving system information and providing a means to receive the information of highest priority to the master first. Write Zone operations offer a way to send commands to some or all of the devices on a PMBus in simple, short transaction – a huge improvement over the previous Group Command protocol.

This application note has provided a more in depth discussion of the zone operations. Several detailed examples were also given to provide guidance to anyone implementing zone operations in their system. The PMBus Specification Working Group hopes you find this application note helpful.

References

[R01] PMBus™ Power System Management Protocol, Part II, Command Language, System Management Interface Forum, Revision 1.3.1, March 2015

[R02] PMBus Power System Management Protocol, Part III, AVSBus, System Management Interface Forum, Revision 1.3.1, March 2015

[R03] System Management Bus (SMBus) Specification, System Management Interface Forum, Version 3.0, 21 December 2014

[R04] I²C-bus specification and user manual, Revision 6, NXP Semiconductors, 4 April 2014

APPENDIX I. Reference Information

11. Reference Information

11.1. Signal and Parameter Names

The names of signals, commands and parameters are given in capital letters. Underscores are used to separate words rather than embedded spaces (example: SIGNAL_NAME).

The names of signals that are active low and parameters that are true when the value is 0 are indicated with a number sign (#) suffix (example: WRITE# means that the device can be written when the signal is low).

11.2. Numerical Formats

All numbers are decimal unless explicitly designated otherwise.

11.2.1. Decimal Numbers

Numbers explicitly identified as decimal are identified with a suffix of “d”.

11.2.2. Floating Point Numbers

Numbers explicitly identified as floating point are identified with a suffix of “f”.

11.2.3. Binary Numbers

Numbers in binary format are indicated by a suffix of “b”. Unless otherwise indicated, all binary numbers are unsigned.

All signed binary numbers are two’s complement.

11.2.4. Hexadecimal Numbers

Numbers in hexadecimal format are indicated by a suffix of “h”.

11.2.5. Examples

255d⇔ FFh ⇔ 11111111b

175d⇔ AFh ⇔ 10101111b

1.2f

11.3. Byte And Bit Order

As specified in the SMBus specification, Version 3.0 [R03]:

- When data is transmitted, the lowest order byte is sent first and the highest order byte is sent last.
- Within any byte, the most significant bit (MSB) is sent first and the least significant bit (LSB) is sent last.

11.4. Bit And Byte Illustrations

The transmission of bits, bytes and packets is illustrated in this section.

In all cases, the least significant bit is indicated as Bit 0. The most significant bit of a byte is always Bit 7, as shown below in Figure 24.

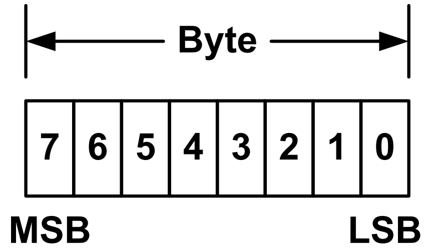

Figure 24. Bit Order Within A Byte

Within this specification, transactions over the PMBus are described. The symbols used to describe the details of those transactions and protocols are shown in Table 9.

Table 9. Bit And Byte Symbols Used In This Specification

Symbol	Meaning
1 1 7	A unshaded vertical rectangle indicates a single bit sent from the host (bus master) to a slave
1 1 7	A shaded vertical rectangle with a shaded interior indicates a bit sent from a slave device to the bus master.
8 8 **DATA NAME** **DATA VALUE**	An unshaded rectangle with a number over it represents one or more bits, as indicated by the number, sent from the master to the slave. The name of the data or bit field may be included within the rectangle. If the data has a specific value, as might be shown in an example of a command, the value is written below the data or bit field name.
8 8 **DATA NAME** **DATA VALUE**	A shaded rectangle with a number over it represents one or more bits, as indicated by the number, sent from the slave to the master. The name of the data or bit field may be included within the rectangle. If the data has a specific value, as might be shown in an example of a command, the value is written below the data or bit field name.
S	The START condition sent from a bus master device. The START condition is not a bit and does not have a number 1 over it.
Sr	A REPEATED START condition sent from a bus master device. The REPEATED START condition is not a bit and does not have a number 1 over it.
1 **A**	An Acknowledge (ACK) condition send from the host

Symbol	Meaning
1 NA	A Not Acknowledge (NACK) condition sent from the host
1 A	An Acknowledge (ACK) condition sent from a slave device
1 NA	A Not Acknowledge (NACK) condition sent from a slave device
P	A STOP condition sent by a bus master device. The STOP condition is not a bit and does not have a number 1 over it.
7 SLAVE ADDRESS	The first seven bits of the address byte, generally corresponding to the physical address of the device.
1 R	The bit [0] of the address byte with a value of 1, indicating the device is being addressed with a read.
1 W	The bit [0] of the address byte with a value of 0, indicating the device is being addressed with a write.
7 BROADCAST ADDRESS	The SMBus broadcast address to which all devices must respond. The value is 0000000b. This is always used only with the bit [0] equal to 0 (write).

11.5. Abbreviations, Acronyms And Definitions

Term	Definition
ACK	ACKnowedge. The response from a receiving unit indicating that it has received a byte. See the SMBus specification [R03] for more information.

Term	Definition
Assert, Asserted	A signal is asserted when the signal is true. For example, a signal called FAULT is asserted when a fault has been detected. See Negate.
AVS	Adaptive Voltage Scaling. AVS is used by a device to control its supply voltage, generally to minimize power consumption for a given operating condition.
AVSBus	AVSBus is an interface designed to facilitate and expedite point-to-point communication between an ASIC, FPGA, or other logic, memory, or processor devices and a POL control device on a system for the purpose of adaptive voltage scaling.
Bias, Bias Power	Power to the PMBus device's control circuit or ICs
Clear	When referring to a bit or bits, this means setting the value to zero.
Default Store	A non-volatile memory store most typically used by the PMBus device manufacturer to store default values
Disable, Disable Output	To instruct the PMBus device to stop the power conversion process and to stop delivering energy to the output. The device's control circuitry remains active and the device can communicate via the SMBus.
Enable, Enable Output	To instruct the PMBus device to start the power conversion process and to start delivering energy to the output.
Host	A host is a specialized master that provides the main interface to the system's CPU. A host must be a master-slave and must support the SMBus host notify protocol. There may be at most one host in a system. See the SMBus specification [R03] for more information.
IIN	Input current
Inhibit	To stop the transfer of energy to the output while a give condition, such as excessive internal temperature, is present.
IOUT	Output current
LSB	Least significant bit
Master	A master is a device that issues commands, generates the clocks, and terminates the transfer. See the SMBus specification [R03] for more information.
MFR	Manufacturer
MSB	Most significant bit
NACK	Not ACKnowledge. The response from a receiving unit that it has received invalid data. See the SMBus specification [R03] for more information.

Term	Definition
Negate, Negated	A signal is negated when the signal is false. For example, a signal called FAULT is negated when no fault has been detected. See Assert.
Negative Output Current	Current that flows into the converter's output.
OC	Overcurrent
OP	Overpower
Operating Memory	The conceptual location where a PMBus maintains the data and parameters it uses operate.
OT	Overtemperature
OV	Overvoltage
PEC	Packet Error Checking. See the SMBus specification [R03] for more information.
PIN	Input power
Pin Programmed Values	Values entered into the PMBus device through physical pins. Values can be set, for example, by connecting a pin to ground, connecting a pin to bias power, leaving the pin unconnected or connecting the pin to ground or bias through a resistor.
Plain Text	Characters stored according to ISO/IEC 8859-1
POL	Point-of-load
Positive Output Current	Current that flows out of the converter's output.
POUT	Output power
Product Literature	Data sheets, product briefs, application notes or any other documentation describing the operation and application of a device.
Set	When referring to a bit or bits, this means setting the value to one.
Shut Down	Disable or turn off the output. This generally implies that the output remains off until the device is instructed to turn it back on. The device's control circuit remains active and the device can respond to commands received from the SMBus port.
Sink (Current)	A power converter sinks current when current is flowing from the load into the converter's output. The current in this condition is declared to be negative.
Slave	A slave is a device that is receiving or responding to a command. See the SMBus specification [R03] for more information.
SMBus	System Management Bus - See the SMBus specification [R03] for more information.
Source (Current)	A power converter sources current when current is flowing from the converter's output to the load. The current in this condition is declared to be positive.

Term	Definition
Turn Off	Turn Off means to “turn off the output”, that is, stop the delivery of energy to the device’s output. The device’s control circuit remains active and the device can respond to commands received from the SMBus port. The same as Disable. See Turn On.
Turn On	Turn On means to “turn on the output”, that is, start the delivery of energy to the device’s output. The same as Enable. See Turn Off.
UC	Undercurrent (Excessive sink current by a synchronous rectifier)
User Store	A non-volatile memory store most often used by the PMBus device user to store an image, or snapshot, of the Operating Memory.
UT	Undertemperature
UV	Undervoltage
VIN	Input voltage
VOUT	Output voltage
X	When used to define a binary value X means that the value of that bit is “don’t care”.

APPENDIX II. Summary Of Changes

DISCLAIMER: The section is provided for reference only and for the convenience of the reader. No suggestion, statement or guarantee is made that the description of the changes listed below is sufficient to design a device compliant with this document.

Revision 1.0.01: Corrected incorrect addresses in Figure 15