flex Power Modules

APPLICATION NOTE 320

Snapshot parameter capture - digital Point of Load regulators

Abstract

The digital products can be configured, controlled and monitored through a digital serial interface using the PMBus™ power management protocol. This application note provides information on how to use the snapshot feature of the digital PoL regulators. This application note applies to the following products:

- BMR462
- BMR463
- BMR464
- BMR465
- BMR467
- BMR469

Contents

Snapshot Operation: BMR462, BMR463xxx2/6, BMR464xxx2/6

The snapshot function and flow of snapshot data is illustrated in Figure 1 and Figure 2. Refer to Appendix 1 for details of the snapshot PMBus commands.

Snapshot enable

The snapshot function is enabled or disabled by setting or clearing bit [1] in PMBus command MISC_CONFIG. By default the snapshot function is enabled. When snapshot is enabled and the output voltage is enabled, one set of parametric data is continuously copied to snapshot register in RAM every controller cycle (~4 ms). The snapshot data being in RAM can read anytime by the user through the PMBus command SNAPSHOT. When the snapshot function is disabled, or the output voltage is disabled, snapshot data in RAM is retained.

Fault capture

When an operational fault occurs while the snapshot function is enabled, parametric data will automatically be captured and stored to NVM in order to save information about the fault situation.

Note: In order for the fault capture to work properly, the fault responses must be set in latched mode, i.e. Disable, No Retry (see example in Figure 3), and the input undervoltage threshold (VIN_UV_FAULT_LIMIT) must be set to 0V to avoid corruption of snapshot data in NVM when the input voltage is cycled.
The user may access the captured data in NVM by the following steps:

1. Disable the output voltage by CTRL pin or PMBus command as applicable.
2. Disable the snapshot function by clearing bit [1] in MISC_CONFIG.
3. Perform a copy of snapshot data in NVM to RAM by writing 0x01 to command SNAPSHOT_CONTROL.
4. Read snapshot data from RAM by reading command SNAPSHOT.

Note: Faults captured by the snapshot function will overwrite previously captured faults in NVM. Thus, the snapshot data in NVM holds the information from when the last fault occurred. Therefore, after a fault, the user should keep the output disabled and keep snapshot disabled in MISC_CONFIG until the data captured in NVM has been analyzed.

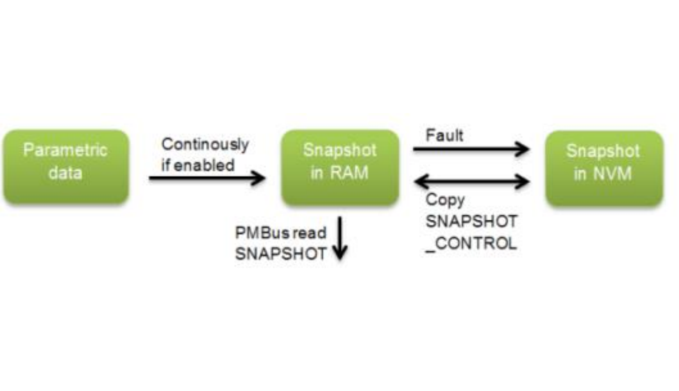


Figure 1: Flow of shapshat data

Note: There are special considerations for applications where the product is enabled by input voltage (self enable). See section *Snapshot Considerations.*

The user may manually store snapshot in RAM to NVM by the SNAPSHOT_CONTROL command. If doing this snapshot must be disabled in MISC_CONFIG and output voltage disabled.

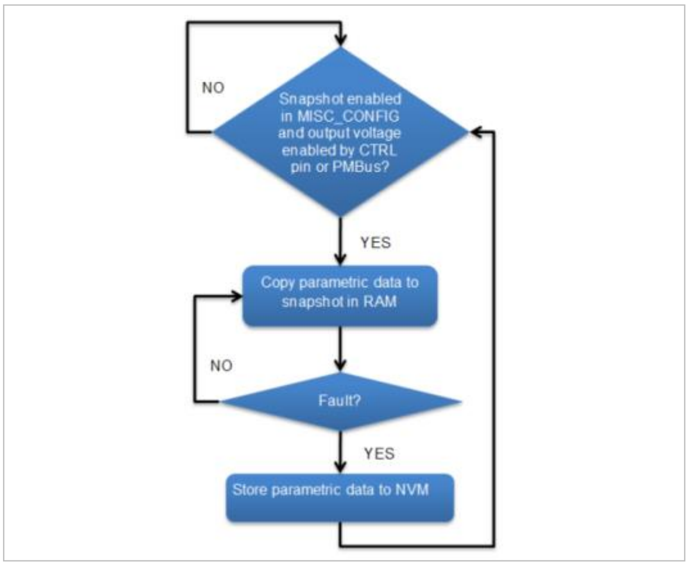
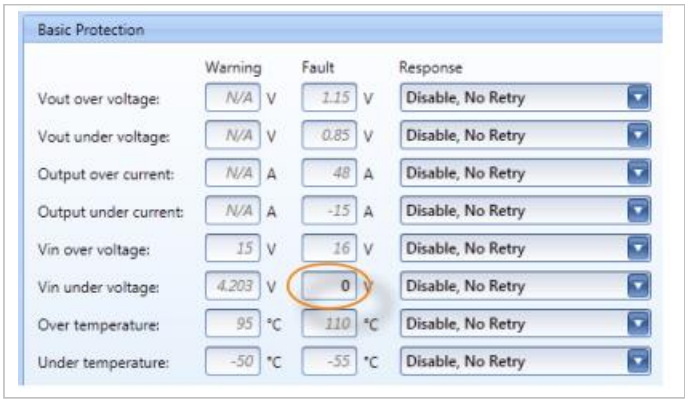

NO
Snapshot enabled in MISC_CONFIG and output voltage enabled by CTRL pin or PMBus?
YES
Copy parametric data to snapshot in RAM
NO
Fault?
YES
Store parametric data to NVM

Figure 2: Snapshot functionality flowchart

Example for BMR462, BMR463xxx2/6, BMR464xxx2/6 using Flex Power Designer

To demonstrate the snapshot function, let’s walkthrough an example with a BMR4640x02/001 to capture an overcurrent fault. For proper snapshot function we need to make sure the input undervoltage threshold is set to 0V and the fault responses are set to latched mode, Disable, No Retry, see Figure 3.

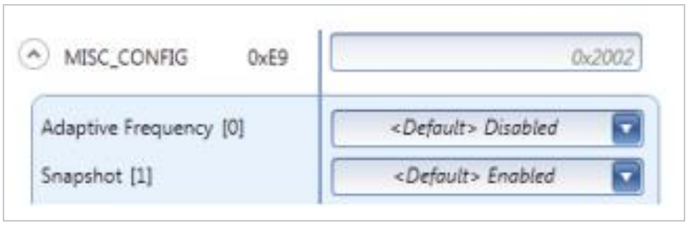


Basic Protection

	Warning		Fault		Response
Vout over voltage:	N/A	V	1.15	V	Disable, No Retry
Vout under voltage:	N/A	V	0.85	V	Disable, No Retry
Output over current:	N/A	A	48	A	Disable, No Retry
Output under current:	N/A	A	-15	A	Disable, No Retry
Vin over voltage:	15	V	16	V	Disable, No Retry
Vin under voltage:	4.203	V	0	V	Disable, No Retry
Over temperature:	95	°C	110	°C	Disable, No Retry
Under temperature:	-50	°C	-55	°C	Disable, No Retry

Figure 3: Protection setting for proper snapshot function

Then, we make sure the snapshot function is enabled in MISC_CONFIG in Registers view, see Figure 4. The settings are stored to RAM and NVM.



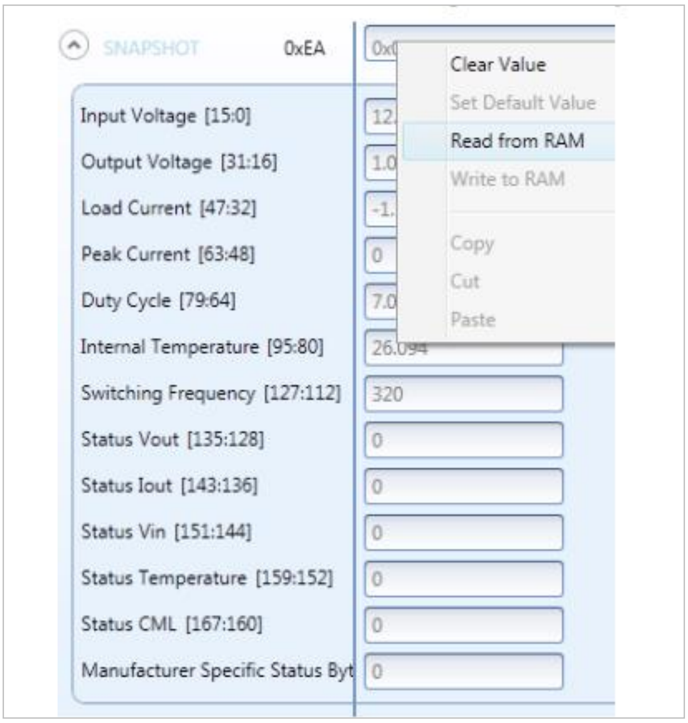
MISC_CONFIG 0xE9 0x2002

Adaptive Frequency [0] <Default> Disabled

Snapshot [1] <Default> Enabled

Figure 4: Snapshot function enabled MISC_CONFIG in register view

While the snapshot function is enabled, we can read the SNAPSHOT command in the Registers view by right click and choose Read from RAM, see Figure 5. If making subsequent reads from SNAPSHOT we can see that we get realtime values from operating parameters such as input voltage and temperature.



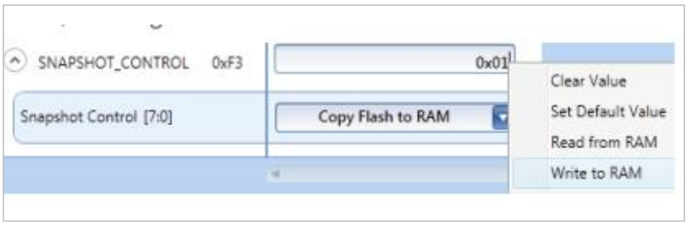
SNAPSHOT 0xEA 0x0

Input Voltage [15:0]	12.
Output Voltage [31:16]	1.0
Load Current [47:32]	-1.
Peak Current [63:48]	0
Duty Cycle [79:64]	7.0
Internal Temperature [95:80]	26.094
Switching Frequency [127:112]	320
Status Vout [135:128]	0
Status Iout [143:136]	0
Status Vin [151:144]	0
Status Temperature [159:152]	0
Status CML [167:160]	0
Manufacturer Specific Status Byt	0

Figure 5: Read of SNAPSHOT command in register view

As a next step we will provoke an overcurrent fault by shorting the output of the BMR464. After the fault we follow the steps provided on previous page to recover the snapshot data from the fault moment:

- Disable the output voltage, in this case by setting the CTRL pin in disable state.
- Disable the snapshot function by clearing bit [1] of MISC_CONFIG in Registers view and perform a Write to RAM operation. If we now again make reads from SNAPSHOT we will see that snapshot data in RAM is no longer updated.
- Restore snapshot data from NVM to RAM by writing 0x01 to SNAPSHOT_CONTROL. This can be done in the Registers view by right-click and choose Write to RAM, see Figure 6.

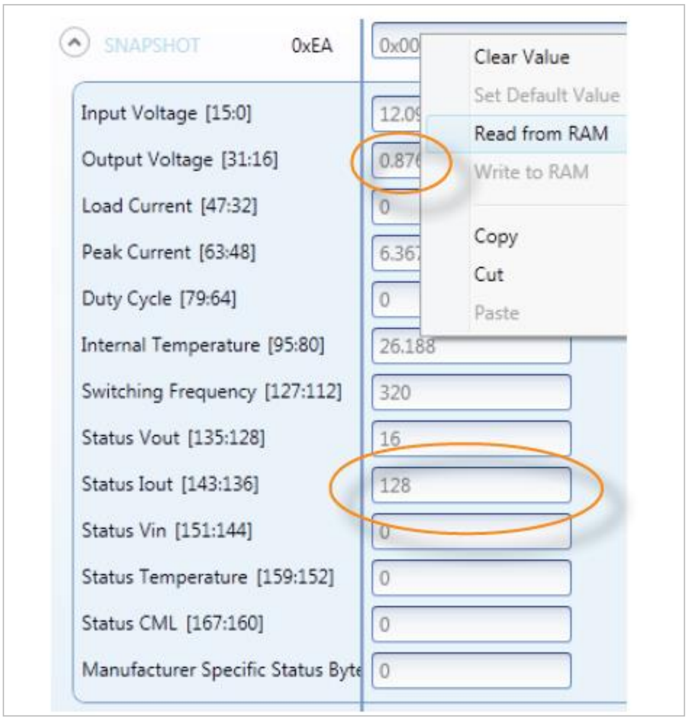


SNAPSHOT_CONTROL 0xF3 0x01

Snapshot Control [7:0] Copy Flash to RAM

Figure 6: Write of SNAPSHOT in register view

- The snapshot data we are looking for is now in RAM and we can read the SNAPSHOT command to find out status and parameters from the fault moment, see Figure 7. For example we can see a fault indication is set in Status Iout field and that the output voltage had dropped to 0.87V when snapshot was captured.



SNAPSHOT 0xEA 0x00

Input Voltage [15:0] 12.0

Output Voltage [31:16] 0.87

Load Current [47:32] 0

Peak Current [63:48] 6.36

Duty Cycle [79:64] 0

Internal Temperature [95:80] 26.188

Switching Frequency [127:112] 320

Status Vout [135:128] 16

Status Iout [143:136] 128

Status Vin [151:144] 0

Status Temperature [159:152] 0

Status CML [167:160] 0

Manufacturer Specific Status Byte 0

Figure 7: Read of SNAPSHOT command after restoring from NVM

Snapshot operation - BMR463xxx8/9, BMR464xxx8/9

The snapshot function and flow of snapshot data is illustrated in Figure 8 and Figure 9. Refer to Appendix 1 for details of the snapshot PMBus commands.

Snapshot enable

The snapshot function is enabled or disabled by setting or clearing bit [1] in PMBus command MISC_CONFIG. By default the snapshot function is enabled.

When snapshot is enabled, one set of parametric data is continuously copied to snapshot register in RAM every controller cycle (~4 ms) . The snapshot data being in RAM can anytime be read by the user through the PMBus command SNAPSHOT. When the snapshot function is disabled, snapshot data in RAM is retained.

Fault capture

When an operational fault occurs while the snapshot function is enabled, parametric data will automatically be captured to snapshot register in RAM and stored to NVM. At the same time the continous copy of parametric data to RAM is paused until the output voltage is disabled by CTRL pin or PMBus (see Figure 9), thus parametric data from the fault situation is retained in RAM and can be read via the SNAPSHOT command.

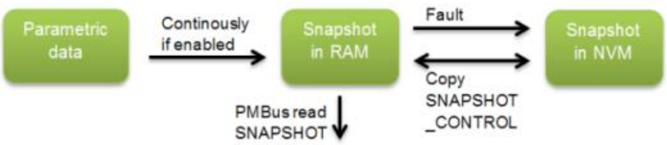


Figure 8: Flow of snapshot data

Note: In order for the fault capture to work properly, the fault responses must be set in latched mode, i.e. Disable, No Retry (see example in Figure 10). In most cases it is also appropriate to set the input undervoltage threshold (VIN_UV_FAULT_LIMIT) to 0V to avoid that snapshot data in NVM is overwritten when the input voltage is cycled. See section Snapshot Considerations for more details.

The user may access the captured data in NVM by the following steps:

1. Disable the output voltage by CTRL pin or PMBus command as applicable.
2. Disable the snapshot function by clearing bit [1] in MISC_CONFIG.
3. Perform a copy of snapshot data in NVM to RAM by writing 0x01 to command SNAPSHOT_CONTROL.
4. Read snapshot data from RAM by reading command SNAPSHOT.

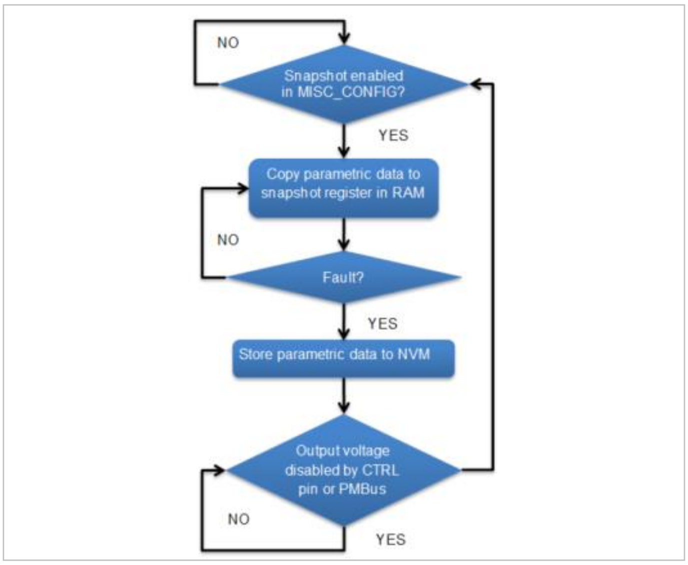

NO
Snapshot enabled in MISC_CONFIG?
YES
Copy parametric data to snapshot register in RAM
NO
Fault?
YES
Store parametric data to NVM
Output voltage disabled by CTRL pin or PMBus
NO
YES

Figure 9: Snapshot funtionality flowchart

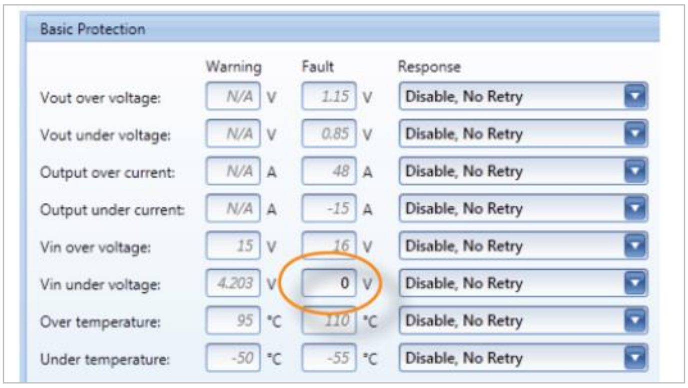
Note: Faults captured by the snapshot function will overwrite previously captured faults in NVM. Thus, the snapshot data in NVM holds the information from when the last fault occurred. Therefore, after a fault, the user should keep the output disabled and keep snapshot disabled in MISC_CONFIG until the data captured in NVM has been analyzed.

The user may manually store snapshot in RAM to NVM by the SNAPSHOT_CONTROL command. If doing this snapshot must be disabled in MISC_CONFIG and output voltage disabled.

Example for BMR463xxx8/9, BMR464xxx8/9 using Flex Power Designer

To demonstrate the snapshot function, let's walkthrough an example with a BMR4640x08/001 to capture an overcurrent fault.

For proper snapshot function we need to make sure the fault responses are set to latched mode, Disable, No Retry, see Figure 10. We also set the input undervoltage threshold to 0V to avoid snapshot being triggered when the input voltage is cycled.

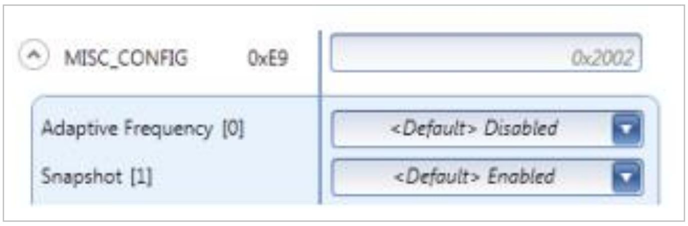


Basic Protection

	Warning		Fault		Response
Vout over voltage:	N/A	V	1.15	V	Disable, No Retry
Vout under voltage:	N/A	V	0.85	V	Disable, No Retry
Output over current:	N/A	A	48	A	Disable, No Retry
Output under current:	N/A	A	-15	A	Disable, No Retry
Vin over voltage:	15	V	16	V	Disable, No Retry
Vin under voltage:	4.203	V	0	V	Disable, No Retry
Over temperature:	95	°C	110	°C	Disable, No Retry
Under temperature:	-50	°C	-55	°C	Disable, No Retry

Figure 10: Protection settings for proper snapshot function

Then, we make sure the snapshot function is enabled in MISC_CONFIG in Registers view, see Figure 11.

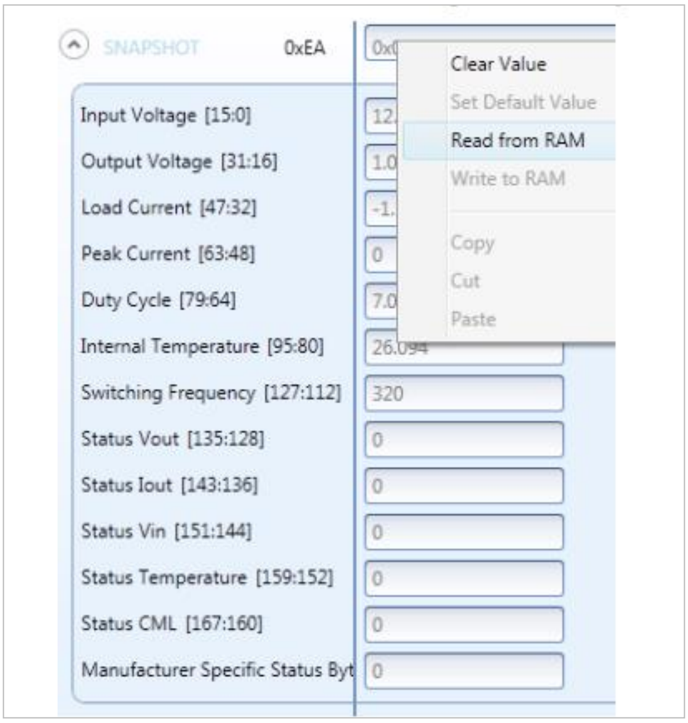


MISC_CONFIG	0xE9	0x2002
Adaptive Frequency [0]		<Default> Disabled
Snapshot [1]		<Default> Enabled

Figure 11: Snapshot function enabled in MISC_CONFIG in register

The settings are stored to RAM and NVM.

While the snapshot function is enabled, we can read the SNAPSHOT command in the Registers view by right-click and choose Read from RAM, see Figure 12. If making subsequent reads from SNAPSHOT we can see that we get realtime values from operating parameters such as input voltage and temperature.



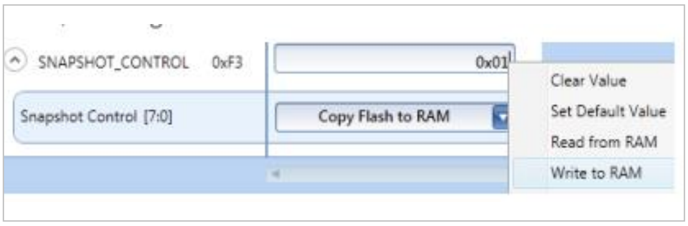
SNAPSHOT	0xEA	0x(
Input Voltage [15:0]		12.
Output Voltage [31:16]		1.0
Load Current [47:32]		-1.
Peak Current [63:48]		0
Duty Cycle [79:64]		7.0
Internal Temperature [95:80]		26.094
Switching Frequency [127:112]		320
Status Vout [135:128]		0
Status Iout [143:136]		0
Status Vin [151:144]		0
Status Temperature [159:152]		0
Status CML [167:160]		0
Manufacturer Specific Status Byt		0

Figure 12: Read of SNAPSHOT command in register view

As a next step we will provoke an overcurrent fault by shorting the output of the BMR464. After the fault we can again return to reading the SNAPSHOT command and now see that the snapshot data in RAM is no longer updated but retained with the data from the fault moment.

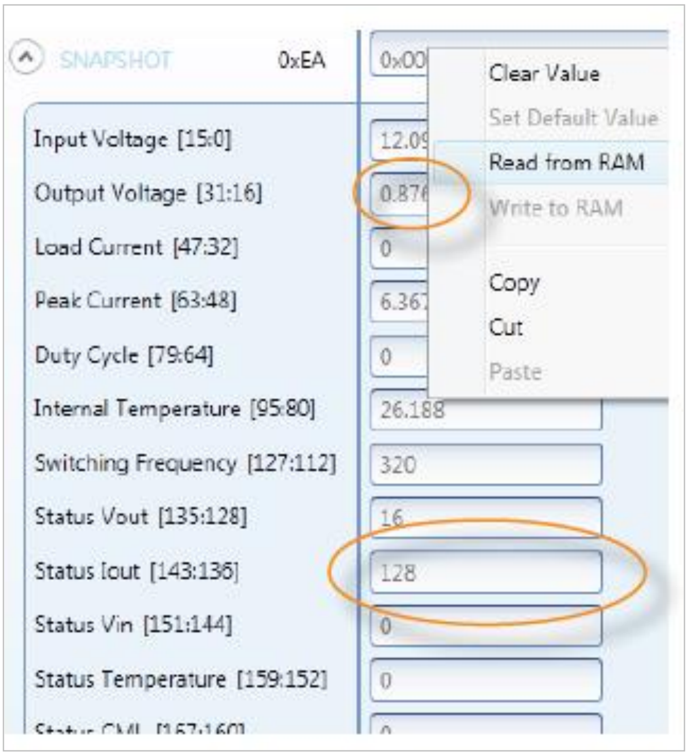
Now we cycle the input voltage, which means the snapshot data in RAM is lost. We follow the steps provided on previous page to recover the snapshot data in NVM from the fault moment:

1. Disable the output voltage, in this case by setting the CTRL pin in disable state.
2. Disable the snapshot function by clearing bit [1] of MISC_CONFIG in Registers view and perform a Write to RAM operation.
3. Restore snapshot data from NVM to RAM by writing 0x01 to SNAPSHOT_CONTROL. This can be done in the Registers view by right-click and choose Write to RAM, see Figure 13.
4. The snapshot data we are looking for is now in RAM and we can read the SNAPSHOT command to find out status and parameters from the fault moment, see Figure 14. For example we can see a fault indication is set in Status Iout field and that the output voltage had dropped to 0.87V when snapshot was captured.



SNAPSHOT_CONTROL	0xF3	0x01
Snapshot Control [7:0]		Copy Flash to RAM

Figure 13: Write of SNAPSHOT_CONTROL in register view



SNAPSHOT	0xEA	0x00
Input Voltage [15:0]		12.09
Output Voltage [31:16]		0.876
Load Current [47:32]		0
Peak Current [63:48]		6.367
Duty Cycle [79:64]		0
Internal Temperature [95:80]		26.188
Switching Frequency [127:112]		320
Status Vout [135:128]		16
Status Iout [143:136]		128
Status Vin [151:144]		0
Status Temperature [159:152]		0
Status CML [167:160]		0

Figure 14: Read of SNAPSHfOT command after restoring from NVM

Snapshot Operation: BMR465, BMR467, BMR469

The snapshot function and flow of snapshot data is illustrated in Figure 15 and Figure 16.

Snapshot enabled

During normal operation, regardless of the output voltage being enabled or disabled, one set of parametric data is continuously copied to snapshot register in RAM every controller cycle (~2.5 ms). The snapshot data in RAM can anytime be read by the user through the PMBus command SNAPSHOT. Refer to Appendix 1 for a detailed description of the information contained in the snapshot register.

Snapshot disabled

When an operational fault occurs, which has not been masked through the command SNAPSHOT_FAULT_MASK, parametric data will automatically be stored to NVM in order to capture information about the fault situation. When this happens, the continuous update of snapshot data in RAM is disabled, as shown in Figure 16. This means the snapshot data in RAM is contained to the same information as captured at the fault and stored to NVM. The disabled state is indicated by the NVM status bits [183:176] in command SNAPSHOT.

During the disabled state the user can read out snapshot data as captured at the fault, using the SNAPSHOT command.

Note: If input voltage is cycled when the snapshot function is disabled, the snapshot function continues to be disabled and snapshot data in RAM is automatically restored from NVM. Thus, snapshot data from the fault moment is always accessed by reading the SNAPSHOT command.

In order to re-enable the update of snapshot data in RAM the user has to erase snapshot data in NVM by writing 0x03 to the SNAPSHOT_CONTROL command. The output voltage must be disabled when doing this.

Note: Data is captured only for the first fault that occurs. After that, no faults will capture data before the snapshot erase command has been sent. This means that the user should initate a snapshot erase before the output voltage is enabled to guarantee that no previous fault is keeping the snapshot function in the disabled state.

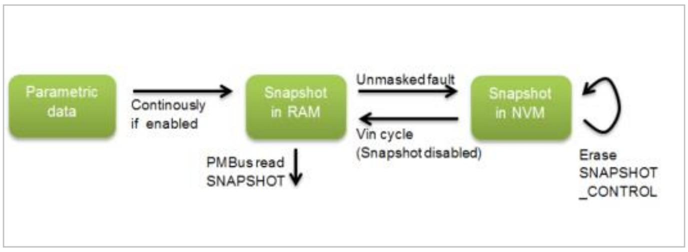


Figure 15: Flow of shapshat data

Note: There are special considerations for applications where the product is enabled by input voltage (self-enable). See section *Snapshot Considerations.*

Refer to Appendix 1 for a detailed description of the PMBus commands SNAPSHOT, SNAPSHOT_FAULT_MASK and SNAPSHOT_CONTROL.

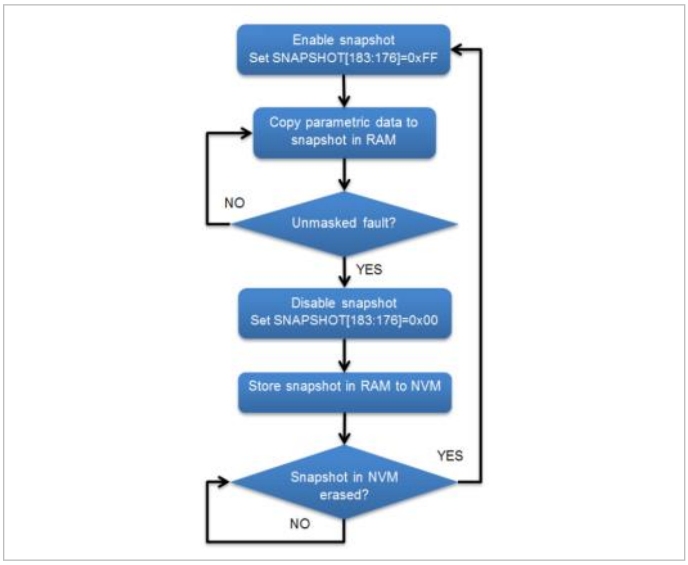
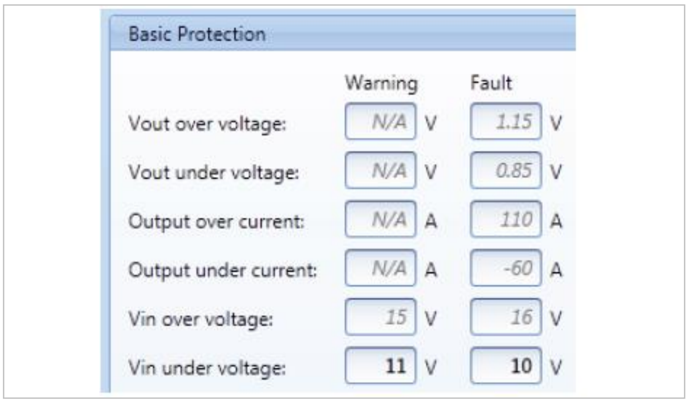

Enable snapshot
Set SNAPSHOT[183:176]=0xFF
Copy parametric data to snapshot in RAM
NO
Unmasked fault?
YES
Disable snapshot
Set SNAPSHOT[183:176]=0x00
Store snapshot in RAM to NVM
YES
Snapshot in NVM erased?
NO

Figure 16: Snapshot functionality flowchart

Example for BMR465 using Flex Power Designer

To demonstrate the snapshot function, let's walkthrough an example with a BMR465 to capture an input under voltage fault.

The module is supplied with 12 V and the fault thresholds for input voltage are changed, see Figure 17

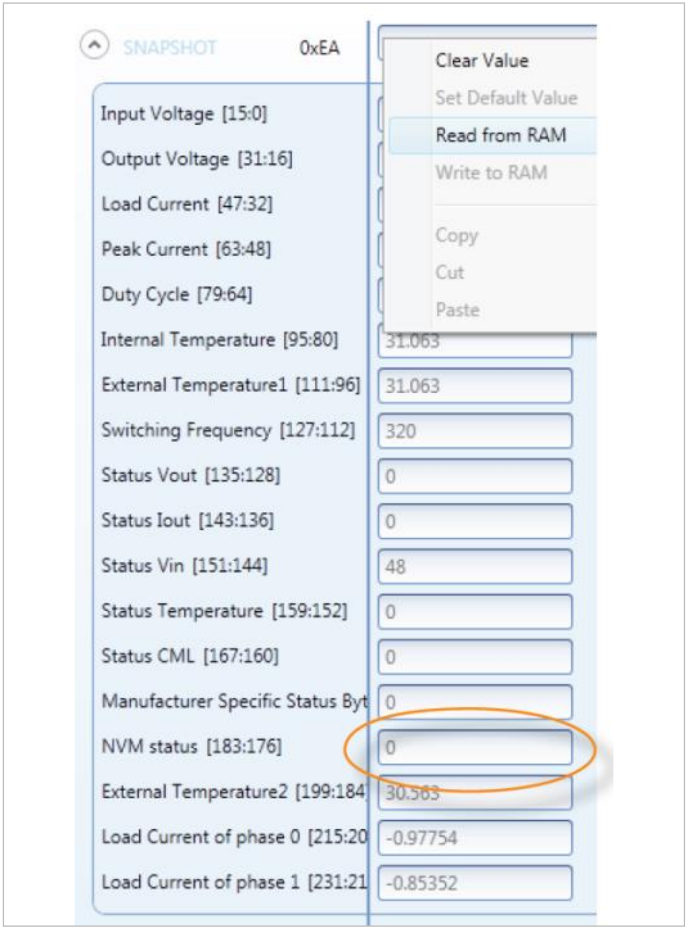


Basic Protection

	Warning		Fault	
Vout over voltage:	N/A	V	1.15	V
Vout under voltage:	N/A	V	0.85	V
Output over current:	N/A	A	110	A
Output under current:	N/A	A	-60	A
Vin over voltage:	15	V	16	V
Vin under voltage:	11	V	10	V

Figure 17: Change Vin under voltage thresholds

As a start we will check if the snapshot function of the BMR465 we are using is enabled or disabled. This can be done by reading the SNAPSHOT command in the Registers view by right-click and choose Read from RAM, and checking the value of the NVM status bits [183:176], see Figure 18. In this case the value is 0 which means the snapshot function is disabled.

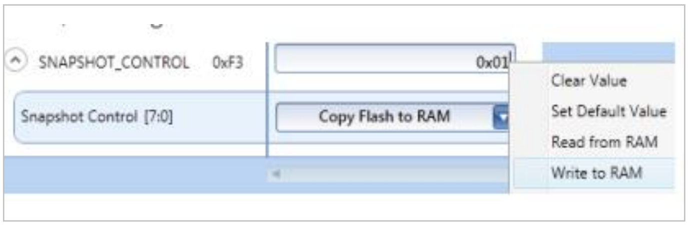


Input Voltage [15:0]	
Output Voltage [31:16]	
Load Current [47:32]	
Peak Current [63:48]	
Duty Cycle [79:64]	
Internal Temperature [95:80]	31.063
External Temperature1 [111:96]	31.063
Switching Frequency [127:112]	320
Status Vout [135:128]	0
Status Iout [143:136]	0
Status Vin [151:144]	48
Status Temperature [159:152]	0
Status CML [167:160]	0
Manufacturer Specific Status Byt	0
NVM status [183:176]	0
External Temperature2 [199:184]	30.563
Load Current of phase 0 [215:20	-0.97754
Load Current of phase 1 [231:21	-0.85352

Figure 18: Read of SNAPSHOT command in register view

This also means that the values we read from SNAPSHOT are not up-to-date values but rather values from the time when the snapshot function was disabled due to a fault.

In order to re-enable the snapshot function we need to make sure the output voltage is disabled and then write 0x03 to SNAPSHOT_CONTROL. This can be done in the Registers view by right-click and choose Write to RAM, see Figure 19.



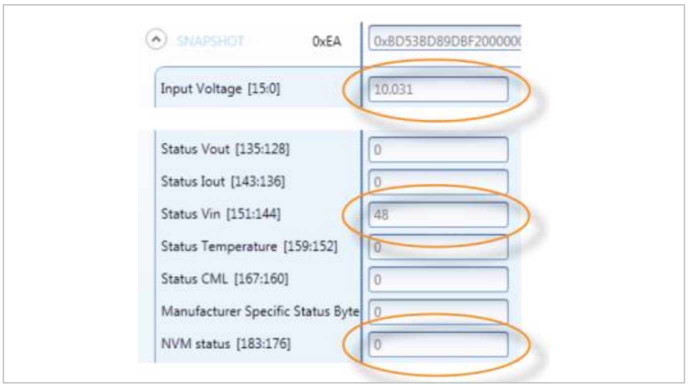
Snapshot Control [7:0]: Copy Flash to RAM

Figure 19: Write of SNAPSHOT_CONTROL in register view

If we now again return to read command SNAPSHOT we will see NVM status bits [183:176] read 255. If making subsequent reads from SNAPSHOT we can also see that we now get realtime values from operating parameters such as input voltage and temperature.

As a next step we will provoke an input under voltage fault by lowering the input voltage below 10 V while the output voltage is enabled. The SNAPSHOT_FAULT_ MASK value is kept at default which means input under voltage faults are not masked. After the fault we now can read the SNAPSHOT command to find out status and parameters from the fault moment, see Figure 20. For example, we can see a fault indication is set in Status Vin field and the Input Voltage field shows the input voltage level that triggered the fault. We also see the NVM status bits are now 0 again, thus snapshot data has been stored to NVM and the snapshot function has been disabled.

As a last step we cycle the input voltage and again read SNAPSHOT. We find that the snapshot function is still disabled and the snapshot values from the fault moment has been restored from NVM.



SNAPSHOT	0xEA	0xBD53BD89DBF200000
Input Voltage [15:0]		10.031
Status Vout [135:128]		0
Status Iout [143:136]		0
Status Vin [151:144]		48
Status Temperature [159:152]		0
Status CML [167:160]		0
Manufacturer Specific Status Byte		0
NVM status [183:176]		0

Figure 20: Read of SNAPSHOT command after input undervoltage fault

Snapshot considerations

Snapshot and input undervoltage faults

For BMR462, BMR463xxx2/6 and BMR464xxx2/6, input undervoltage faults cannot be captured by the snapshot function since VIN_UV_FAULT_LIMIT should be set to 0V.

For other products capturing snapshot data in the case where an input undervoltage fault occurs due to the input voltage drops below the operating range of the product, may not work depending on how fast the input voltage drops. If the input voltage drops too fast there will not be enough time to complete the store of snapshot data to NVM. A way of gaining more time is to set VIN_UV_FAULT_LIMIT as high as possible, in order to increase the time between the point where the fault is triggered and the point where the input voltage falls below the operating range of the converter's controller.

Applications with self-enable by input voltage

Regarding the snapshot function there are two problems with applications where the output voltage is self-enabled by the input voltage (e.g. by using the internal pull-up on the CTRL pin):

1. An input undervoltage fault will occur each time the device is turned off. This means that the snapshot function will trigger a store of snapshot data to NVM each time the device is turned off, possibly overwriting snapshot data in NVM from a previous fault.

2. In order to perform the SNAPSHOT_CONTROL command functions of restoring data from NVM (BMR462-464) or erasing data in NVM (BMR465, BMR467 and BMR469), the output voltage must be disabled. Thus it is required that the module is supplied with input voltage but the output voltage being disabled.

The possibilities of dealing with these two problems is different depending on model and product.

For BMR465, BMR467 and BMR469 the user may overcome problem 1 by masking input undervoltage faults in SNAPSHOT_FAULT_MASK. Regarding problem 2 it is required that the output voltage is disabled while performing the NVM erase by the SNAPSHOT_CONTROL command. The user must disable by the CTRL pin or by PMBus OPERATION command after temporarily changing to PMBus enable mode in ON_OFF_CONFIG.

For BMR462-BMR464 the user may overcome problem 1 by by setting the input undervoltage threshold (VIN_UV_FAULT_LIMIT) to 0 V to make sure input undervoltage faults never occur. Note that in such case the level at which the output voltage turns off will not be well defined. Regarding problem 2 it is required that the output voltage is disabled while performing the restore from NVM by the SNAPSHOT_CONTROL command. The user must disable the output voltage by the CTRL pin or by PMBus OPERATION, depending on what enable/disable mode is set in ON_OFF_CONFIG.

Applications with timestamp concerns

Due to the fact that the modules do not track time, there is no timestamp provided in the snapshot readback. Therefore, it is recomended that the system controller detects when the fault occurs and records a timestamp on its own. Additionally, to mitigate the possibility of reading stale snapshot data, you may optionally 'reset' snapshot during a non-faulting operating condition by writing 0x02 to the SNAPSHOT_CONTROL command.

Applications with modules powering the host controller

For BMR462-BMR46 the reading of a stored snapshot capture in NVM requires that the module is disabled. Therefore, the snapshot feature may not be used if the module is powering the same host controller used for PMBus communication.

Reading snapshot—a system controller guide

While the earlier steps in using snapshot using the Flex Power Designer are great for testing the functionality, it is worth considering how your system controller (i.e. PMBus host) will handle the whole lifecycle that involves reading snapshot. By this we mean all the steps from detecting a fault through ensuring the correct snapshot data is always read back. Let's walk through a scenario where we detect, read, and optionally reset snapshot:

- Assume the power system is enabled and currently operating.

- A fault occurs on a module (such as an overvoltage or overcurrent fault, triggering the module to disable).

- The system controller detects fault occurred by looking at SALERT going low, then reading through the devices that have faulted by reading their STATUS_WORD. It may read status word and AND it with 0x0800, which will indicate if the output voltage is not reaching power good, if the result of the ANDing results in a nonzero value. If the output voltage is not reaching power good and device is expected to be enabled, then look at the STATUS_WORD and confirm it is a fault by ANDing with a mask for fault bits, 0xF73F, which will show that some fault has occurred.

```
//Check to see if an unexpected fault
occurred,

//to be done before a power-cycle event.

uint16_t status = PMBusReceiveWord
(0x79); //STATUS_WORD

if(status && 0x0800 & expected_on)

if(status && 0xF73F)

{

//fault occurred and module is off.

//

// read snapshot (and any

prerequisite steps) here

}
```

- Assuming we now confirmed it is a device expected to be on but is now off and has a fault, we can look into obtaining its snapshot readback for further information.

- Read and interpret the SNAPSHOT command (0xEA). To help with interpreting the data readback, use the sample code provided at https://bitbucket.org/flexpowermodules/snapshot-parameter-capture-sample-code/src . On the BMR463 XXX8/9, BMR464 XXX8/9 and BMR465, BMR467 and BMR469, you can read the snapshot after the shutdown-causing fault occurs – the firmware on these modules automatically copy the snapshot-capture in to RAM such that it can be read immediately after the fault.

On the BMR462, BMR463 XXX2/6 and BMR464XXX2/6, you will need to perform the steps of

1. Disable the controller (either via PMBus or CTRL pin)
2. Disable snapshot by clearing bit 1 in the MISC_CONFIG command.
3. Write 0x01 to the SNAPSHOT_CONTROL command. This will copy the snapshot data in NVM to RAM.
4. Finally read back the snapshot command SNAPSHOT.
5. Re-enabling snapshot by setting bit 1 in the MISC_CONFIG command.

The disable step must be done even if the controller has already shut down due to the fault condition.

Optional step: ‘Reset’ snapshot in NVM to a non-fault state – one thing that may be done to ensure a stale snapshot is not read back after it has already been read and processed by the controller. One way to do this for BMR462-464 is by doing the following:

1. Read/process the snapshot after fault detection (if the host controller has not already done this).

2. Since you’ve already captured the earlier faults by reading snapshot, clear any faults still existing by performing a Send Byte transaction with the CLEAR_FAULTS command.

3. Write 0x02 to the SNAPSHOT_CONTROL command.

This will copy the current snapshot readback into NVM.

For BMR465, BMR467 and BMR469 a reset is required and simply done by writing 0x03 to the SNAPSHOT_CONTROL command to erase snapshot data in NVM and re-enable the snapshot function.

Appendix 1 - Snapshot command reference

SNAPSHOT command

Snapshot in RAM is read by the PMBus command SNAPSHOT. For details about the format of each field, refer to the specification of the corresponding READ_x or STATUS_x PMBus command. Note: The seven highest bytes (bits 231:176) apply only to BMR465/467. Bits 176:231 also apply on BMR469.

Bits	Parameter	Description	Corresponding PMBus command
231:216	Load Current Phase 1	BMR465 and BMR467 only. Measured output current from phase 1	READ_IOUT1
215:200	Load Current Phase 0	BMR465 and BMR467 only. Measured output current from phase 0	READ_IOUT0
199:184	Ext.Temperature2	BMR465 and BMR467 only. Not used	N/A
183:176	NVM Status	BMR465, BMR467 and BMR469 only. Shows if snapshot data is written to NVM due to fault. 0x00: NVM is written and snapshot function is disabled. 0xFF: Snapshot function is enabled.	N/A
175:168	Status Mfr Specific	Status related to other specific functions	STATUS_MFR_SPECIFIC
167:160	Status CML	Status related to memory and communication	STATUS_CML
159:152	Status Temperature	Status related to the internal temperature	STATUS_TEMPERATURE
151:144	Status Vin	Status related to the input voltage	STATUS_INPUT
143:136	Status Iout	Status related to the output current	STATUS_IOUT
135:128	Status Vout	Status related to the output voltage	STATUS_VOUT
127:112	Sw. Frequency	Actual switching frequency	READ_FREQUENCY
111:96	Ext. Temperature1	Not used	N/A
95:80	Int. Temperature	Measured temperature of the controller of the device	READ_TEMPERATURE_1
79:64	Duty Cycle	Actual duty cycle	READ_DUTY_CYCLE
63:48	Peak Current	Maximum measured average output current since the output was enabled. Value is reset by an enable/disable cycle (or for BMR465, 467, 469)by writing 0x03 to SNAPSHOT_CONTROL).	N/A
47:32	Load Current	Measured total output current	READ_IOUT
31:16	Output voltage	Measured output voltage	READ_VOUT
15:0	Input Voltage	Measured input voltage	READ_VIN

SNAPSHOT_FAULT_MASK command (BMR465/467/469)

It is possible to control fault(s) will actually trigger a store of snapshot data to NVM. This is controlled through the PMBus command SNAPSHOT_FAULT_

MASK according to Table 2. By setting a bit to '1' the corresponding fault is masked and will not cause a snapshot store to NVM.

SNAPSHOT_CONTROL command

By writing different values to the SNAPSHOT_CONTROL command, different operations related to the snapshot function is performed, see Table 3.

Bit	Function	Description	Corresponding status command fault
13	Group	Mask for rail fault in a fault group	STATUS_MFR_SPECIFIC[2]
12	Phase	Mask for phase fault in a current sharing rail	STATUS_MFR_SPECIFIC[7]
11	CPU	Mask for general-purpose controller fault.	STATUS_CML[2]
10	CRC	Mask for memory packet error check fault.	STATUS_CML[3]
9	VMON UV	Mask for VMON voltage undervoltage fault. The VMON voltage reflects the temperature of the power switches of the device. See specification of command READ_VMON.	STATUS_MFR_SPECIFIC[1]
8	VMON OV	Mask for VMON voltage overvoltage fault. The VMON voltage reflects the temperature of the power switches of the device. See specification of command READ_VMON.	STATUS_MFR_SPECIFIC[0]
7	Iout UC	Mask for output current undercurrent fault	STATUS_IOUT[4]
6	Iout OC	Mask for output current overcurrent fault	STATUS_IOUT[7]
5	Vin UV	Mask for input voltage undervoltage fault	STATUS_INPUT[4]
4	Vin OV	Mask for input voltage overvoltage fault	STATUS_INPUT[7]
3	UT	Mask for undertemperature of temperature of the controller of the device	STATUS_TEMPERATURE[4]
2	OT	Mask for overtemperature of temperature of the controller of the device	STATUS_TEMPERATURE[7]
1	Vout UV	Mask for output voltage undervoltage fault	STATUS_VOUT[4]
0	Vout OV	Mask for output voltage overvoltage fault	STATUS_VOUT[7]

Note: The output voltage must be disabled when performing the SNAPSHOT_CONTROL operations.

Note: The erase NVM operation applies to BMR465 and BMR467 only.

Write Value	Operation
0x01	Copy snapshot data in NVM to RAM
0x02	Store snapshot data in RAM to NVM
0x03	BMR465 only. Erase snapshot data in NVM and re-enable snapshot function. Snapshot data will not be updated or written to NVM after a fault occurs, until this value is written.

flex Power Modules

Flex Power Modules, a business line of Flex, is a leading manufacturer and solution provider of scalable DC/DC power converters primarily serving the data processing, communications, industrial and transportation markets. Offering a wide range of both isolated and non-isolated solutions, its digitally-enabled DC/DC converters include PMBus compatibility supported by the powerful Flex Power Designer.

EMEA (Headquarters) | Torshamnsgatan 28 A, 16440 Kista, Sweden

APAC | 33 Fuhua Road, Jiading District, Shanghai, China 201818

Americas | 6201 America Center Drive, San Jose, CA 95002, USA

✉ pm.info@flex.com
flexpowermodules.com
flexpowerdesigner.com
youtube.com/flexintl

twitter.com/flexpowermodule
flexpowermodules.com/wechat
linkedin.com/showcase/flex-power-modules