User's Guide

UCD91xxx Sequencer and System Health Controller PMBus Command Reference

ABSTRACT

The UCD91xxx Power Supply Sequencer and Monitor supports a wide range of commands that allow an external host to configure, control, and monitor voltage rails. Communication between the sequencer and the host is via an I^2C electrical interface using the PMBus™ command protocol.

The PMBus specification describes the command protocol in general terms. This document describes implementation details that are specific to the UCD91xxx Power Supply Sequencer and Monitor. If a command is not described in this document and it is supported by a UCD91xxx device (see Table 21-1), it functions exactly as described in the PMBus specification. In which case, refer to PMBus specification for more details.

See the device-specific data sheet for a complete description of the features.

This document makes reference to the *UCD Sequencer Studio*. The TI UCD Sequencer Studio is provided for device configuration. This Microsoft® Windows® based, graphical user interface (GUI) offers an intuitive interface for configuring, storing, and monitoring all system operating parameters.

Note: This document does not apply to the UCD90xxx family of devices.

Table of Contents

Trademarks

PMBus™ is a trademark of SMIF, Inc.
Microsoft® and Windows® are registered trademarks of Microsoft Corporation.
All trademarks are the property of their respective owners.

1 Power Supply Sequencing

1.1 Overview

Power sequencing is an essential part of any design, especially in complex systems that utilize multiple power rails. High-performance processing devices such as FPGAs, ASICs, PLDs, DSPs and microcontrollers, require multiple voltage rails to power internal circuitry, such as the core, memory and I/O. These types of applications demand very specific voltage rail power-up and power-down sequencing and supervising to achieve reliable operation, better efficiency and overall system health.

1.2 Rail On and Off Configuration

The UCD91xxx device can control the turn-on and turn-off sequencing of up to 32 voltage rails by setting a power-supply enable pin high or low. There are multiple ways that a turn-on or turn-off event can be initiated.

- In PMBus-based designs, the system PMBus controller can initiate a turn-on event by asserting the PMBus_CNTRL pin or by sending the OPERATION command over the I2C serial bus.
  - The UCD91xxx device can also be configured so that both the PMBus_CNTRL pin must be asserted, and the OPERATION command must be sent over I2C so that a turn-on event is initiated.
- In pin-based designs, the PMBus_CNTRL pin can also be used to turn-on and turn-off.
- The auto-enable setting ignores the OPERATION command and the PMBus_CNTRL pin. Sequence-on is started at power-up after any dependencies and time delays are met for each rail.

To configure a rail's turn-on and turn-off event, use the following PMBus commands:

- PAGE: This command sets the active page, or rail, that the configuration applies to.
- ON_OFF_CONFIG: This command specifies whether the turn-on or turn-off event depends on the PMBus_CNTRL pin, the OPERATION command, both, or none (auto-enable).
- OPERATION: If the turn-on and turn-off events have been configured to have a dependency on this command, the command is used to turn the rail on and off. This command is also used to set the rail's output voltage to the upper or lower margining voltages.

1.3 Rail Power Good

Each rail has a Power Good status determined by the following rules:

- If rail voltage is monitored by an AMON (analog monitoring) pin, the Power Good status is determined by Power Good On and Power Good Off thresholds, which are configured through PMBus.
  - A rail is given Power Good status if the rail voltage is above the Power Good On threshold. Otherwise, the rail is given Not Power Good status if the rail voltage is below the Power Good Off threshold.
  - The rail remains in the current state if the voltage is neither above Power Good On nor below Power Good Off thresholds.
- If rail voltage is monitored by a DMON (digital monitoring) pin, the Power Good status is determined by the input logic level.
  - A rail is given Power Good status when the input is logic HIGH (3.3V).
  - A rail is given Not Power Good status when the input is logic LOW.
- If rail voltage is not monitored by an AMON or DMON pin, the Power Good status is determined by the turn-on and turn-off eligibility of the rail.
  - A rail is immediately given Power Good status when the rail meets all the turn-on conditions set by the user, such as ON_OFF_CONFIG, dependencies on items such as other rails and GPIO, and delays.
  - Similarly, a rail is immediately given Not Power Good status when the rail meets all the turn-off conditions set by the user.
  - The behavior is the same regardless whether a physical EN (enable) pin is assigned to the rail.

A rail is on or within regulation when the measured voltage crosses the Power Good On limit, which is configured through PMBus.

- A rail remains in regulation even if the voltage is below the power good threshold until it's commanded off (seeRail On and Off Configuration), and all sequencing dependencies and timing delays are met.

1.4 Rail Sequence Configuration

When a rail receives a turn-on or turn-off event (seeRail On and Off Configuration), the rail dependency conditions are checked.

The following sequence-on dependency options are supported for each rail:

- An OPERATION command to turn on
- Assertion of the PMBus_CNTRL pin
- One or a group of parent rails has Power Good status
- One or a group of GPIs (general-purpose inputs) has asserted
- One or a group of LGPOs (logical general-purpose outputs) has reached logical-state TRUE
- Any combination of the previous option

The following sequence-off dependency options are supported for each rail:

- An OPERATION command to turn off
- De-assertion of the PMBus_CNTRL pin
- One or a group of parent rails has Not Power Good status
- One or a group of GPIs (general-purpose inputs) has de-asserted
- One or a group of LGPOs (logical general-purpose outputs) has reached logical-state FALSE
- In response to an undervoltage, overvoltage, or max turn-on fault on the rail
- In response to a fault on a different rail when set as a fault shutdown slave to the faulted rail
  - Each rail can include a Fault Shutdown Slaves function. When a rail shuts down as a result of a fault, the associated slave rails also shut down. The device continues to monitor delays and dependencies of the slave rails during the shutdown process.
  - Fault Shutdown Slaves cannot cascade. In other words, if a rail that is acting as a slave shuts down, the associated slave rails do not shut down.
- Any combination of the previous options

When all sequence dependency conditions are fulfilled, the rail then waits for an optional delay time (turn-on delay or turn-off delay), which can be configured through PMBus, before asserting or de-asserting the rail enable pin (EN). The maximum delay time for both turn-on and turn-off delays is 3276ms.

Further configuration options can be set on monitoring the rail voltage based on the EN pin:

- After the EN pin of a rail is asserted, if the rail voltage does not rise above Power Good On threshold within the maximum turn-on time, a Time On Max fault occurs.
  - Maximum turn-on time has a granularity of 1ms. A value of 0ms means that there is no limit, and the device can try to turn on the output voltage indefinitely.
- After the EN pin of a rail is de-asserted, if the rail voltage does not fall below 12.5% nominal output voltage within maximum turn-off time, a Time Off Max warning occurs.

Each rail can also set Sequencing On/Off Timeout periods. The timeout periods begin to increment when a rail receives a turn-on or turn-off event. When the timeout period elapses, the rail executes one of 3 actions including:

- Wait Indefinitely
- Enable or Disable Rail
- Re-sequence (for sequencing on only)

To configure and set the sequencing dependencies and timing delays, use the following PMBus commands:

- Timing Delays
  - TON_DELAY: This command sets the delay to wait before asserting the rail EN pin. The delay counter starts decrementing when the sequence-on dependencies, if any, are met.

- TON_MAX_FAULT_LIMIT: This command sets the upper time limit for the rail voltage to rise above the Power Good On threshold. This is more applicable for analog monitoring features. A value of 0ms means that the device can try to turn on the output voltage indefinitely.
- TOFF_DELAY: This command sets the delay to wait before de-asserting the rail EN pin. The delay counter starts decrementing when the sequence-off dependencies, if any, are met.
- TOFF_MAX_WARN_LIMIT: This command sets the upper time limit for the rail voltage to fall below 12.5% nominal output voltage. This is more applicable for analog monitoring features.

- Sequencing Dependencies
  - GPI_CONFIG: This command configures a GPIO pin as a GPI. To set a GPI as a sequencing dependency, this command must be run prior to SEQ_CONFIG.
  - SEQ_CONFIG: This command configures a rail's sequencing dependencies (other rails, GPIs, LGPOs, fault slave rails). It's also used to assign the EN pin to a rail, set the sequencing timeout periods, and configure the corresponding sequencing timeout actions.
  - GPO_CONFIG_INDEX: This command selects the index of the LGPO that will be used for later GPO_CONFIG commands.
  - GPO_CONFIG: This command configures the functionality of an LGPO, where the state of the pin is determined by a selection of GPIs and statuses processed through combinational logic. In most cases, the statuses are related to pages i.e. Power Good.

Resequence

Re-sequence is a series of actions that shuts down a rail and the Fault Shutdown Slaves, and then re-enables the rails according to sequence-on delay times and dependencies.

A re-sequencing event can be repeated for one to approximately four times or unlimited times. The Time Between Re-Sequences period begins to increment when all the relevant rails are given Not Power Good statuses. When the time period elapses, a re-sequence event begins. When the Enable Re-Sequence Abort is checked, the re-sequence event aborts if any relevant rail triggers a Max Turn Off warning. However, the Max Turn Off warning does not stop an ongoing re-sequence event. If any rails at the re-sequence state are caused by a GPI fault response, the device suspends the entire re-sequence event until the GPI fault is physically clear.

It is also configurable to ignore the POWER_GOOD_OFF and TOFF_MAX_WARN status of a rail when performing re-sequencing if the corresponding bits are set.

- RESEQUENCE command can be used to initiate a re-sequence.

1.5 Rail States

Sequencing dependencies and timing delays for each rail are tracked in a state machine for the UCD91xxx devices. See RAIL_STATE for more details.

1.6 Rail Sequencing Example Use Cases

This section outlines several basic sequencing use cases and the PMBus commands to use. These are only intended to be used for reference.

- Single rail with the following configuration:
  - PMBus-based design where a turn-on event depends on both the PMBus_CNTRL pin and the OPERATION command
  - Sequence on and off dependencies set on a GPI
  - No sequencing timeout. The device waits indefinitely for the sequencing dependencies to be satisfied.
  - No limits set on the time the rail voltage takes to rise above the Power Good On threshold and the time for the rail voltage to fall below 12.5% nominal output voltage
  - Turn-on and turn-off delays set for when the sequence on and off dependencies are met, and the EN pin is asserted and de-asserted respectively
  - PMBus commands to use:

    - PAGE: This command sets the rail that the following PMBus commands applies to. For UCD91xxx devices, this value ranges from 0 to (# of Max Rails - 1). A special value of 0xFF indicates that the following commands applies to all pages/rails.
    - ON_OFF_CONFIG: Refer to the PMBus 1.2 specification or the UCD91xxx PMBus Commander's Reference Guide for how this command is encoded. Based on how the command data byte is encoded, the turn-on and turn-off event can be configured to depend on OPERATION, PMBus_CNTRL, both, or none. The PMBus_CNTRL pin can also be set to active high (pull high to start unit) or active low polarity (pull low to start unit), and the de-assertion of the PMBus_CNTRL can also be configured so that the rail immediately turns off or follows the programmed turn off delays (soft off).
    - TON_DELAY/TOFF_DELAY: The time delay is encoded in a specific data format, LINEAR11, as per the PMBus specification. The LINEAR11 format is a two-byte value that contains an 11-bit, twos-complement mantissa and a 5-bit, twos-complement exponent. Refer to the UCD91xxx PMBus Commander's Reference Guide for more details on this data format. – For example, setting a delay value of 100ms requires sending the value 0xEB20 as part of the data.
    - GPI_CONFIG: Configure a GPIO to be a GPI. This command must be run prior to SEQ_CONFIG.
    - SEQ_CONFIG: Associate a particular EN pin to the rail and indicate that there are sequence on and off dependencies on the GPI that was configured earlier.
    - OPERATION: Based on ON_OFF_CONFIG, a turn-on event is only sent when OPERATION is sent, and PMBus_CNTRL pin is asserted. To send a turn-off event, send OPERATION and de-assert PMBus_CNTRL.

      – Note: The rail does not come on until the sequencing dependencies are met. For example, if the GPI is not asserted, the rail remains off even if OPERATION is sent, and PMBus_CNTRL is asserted. This applies similarly for a rail to turn off.

- Multiple rails where one rail has sequence on/off dependencies on a GPI, and the other rail has sequence on/off dependencies on the first rail. Different turn-on and turn-off timing delays for both.

  – PMBus commands for rail X:

    - PAGE
    - ON_OFF_CONFIG
    - TON_DELAY/TOFF_DELAY
    - GPI_CONFIG
    - SEQ_CONFIG

  – PMBus commands for rail Y, which depends on rail X:

    - PAGE: Use this command to change the rail that the following PMBus commands applies to.
    - ON_OFF_CONFIG
    - TON_DELAY/TOFF_DELAY
    - SEQ_CONFIG: Set a sequence on and off dependency on rail X.

  – To start the rail sequence:

    - PAGE: Change the active page/rail to rail X.
    - Send the turn-on event: Based on rail X's configuration, this depends on sending OPERATION and asserting PMBus_CNTRL.
    - Rail X turns on when the GPI is asserted.
    - Rail Y turns on when it receives its turn-on event (based on ON_OFF_CONFIG) and when Rail X turns on (meets sequence on dependency).

2 GPIOs

2.1 Overview

UCD91xxx has GPIO pins that can function as either inputs or outputs. Each GPIO has configurable output mode options, including open-drain or push-pull outputs that can be actively driven to 3.3 V or ground. Refer to the device-specific pinout for a list of possible uses for the GPIO pins and the maximum number of each type for each use.

GPIO pins can act as dependencies in sequencing and alarm processing. They can also be used for system-level functions such as external interrupts, power-goods, resets, or for the cascading of multiple devices. GPIOs can be sequenced up or down by configuring a rail without a MONx pin but with a GPIO set as an enable.

GPIOs can be configured to support different functionality using PMBus. The list of supported functionalities is as follows:

- Command-Controlled General-Purpose Outputs (GPOs)
- Logic-Controlled General-Purpose Outputs (LGPOs)
- General-Purpose Inputs (GPIs)
- Power-Supply Rail Enables (ENx)
- Cascading Multiple Devices
- Margining Outputs (MARx)

2.2 Command Controlled GPOs

The UCD91xxx device has GPIO pins which can be configured as Command Controlled GPOs. These GPIOs can be used to control LEDs, enable switches, and so forth and are controlled by the following PMBus commands:

- GPIO_SELECT
- GPIO_CONFIG

2.3 Logic GPOs

Logic GPOs, or LGPOs, are special GPIOs that are controlled by an internal Boolean logic builder.

Each Boolean logic builder has a top-level logic gate, which can be configured as AND, OR, or NOR gate with optional time delay.

- The inputs of the top-level logic gate are two AND paths.
  - Each AND path can select a variety of inputs including GPI states, LGPO states, and rail statuses.

Each LGPO can also be configured to act as a simple state machine. State machine mode allows for more complex LGPO behavior.

- The top-level logic gate is omitted, and only one of the two AND paths is evaluated. The result of the state machine is the result of the active AND path.
- If the evaluation result is TRUE, AND Path #1 remains active until its evaluation result becomes FALSE.
  - When AND Path #1's evaluation result becomes false, AND Path #2 becomes active in the next evaluation cycle.
  - AND Path #2 remains active until its evaluation result becomes TRUE, then AND Path #1 becomes active in the next evaluation cycle.
  - An evaluation cycle is triggered when any input signal to the state machine changes state.

Certain LGPOs are synchronized on the same clock edge to enable the LGPOs to change state together. See the device datasheet for details on which LGPOs are synchronized together. Different groups of synchronized LGPOs update their output statuses within 1 to 3us of each other.

2.3.1 Boolean Logic Builder

Each AND path can utilize a variety of inputs such as:

- GPI states

- LGPO states
- Rail statuses and enables
  - See GPO_CONFIG Status Types for more details.
- System statuses
  - Certain system statuses related to System Watchdog timeout or Single Event Upset can be used as inputs.

2.4 GPIs

Up to 32 GPIO pins of the UCD91xxx device can be configured as GPIs. GPIs in the context of the UCD91xxx device family are inputs with special functionality.

The polarity of GPI pins can be configured to be either active high or active low.

2.4.1 GPI Special Functionality

Each GPI can be used as a source of sequence dependency (see Rail Sequence Configuration). The GPI pins can be also used for cascading function (see Cascading). The first defined three GPIs regardless of their main purpose are assigned to the pin selected states function (see Pin Selected Rail States Configuration).

Additional special behaviors can be assigned to each GPI pin:

- **GPI Fault**: The de-assertion of this pin is treated as a fault, which can trigger shutdown actions for any voltage rail (see GPI Fault Responses).
- **Latched Statuses Clear Source**: This pin can be used to clear latched-type statuses (_LATCH) (see GPI Latched Status Clearing).
- **Input Source for Margin Enable**: When this pin is asserted, all rails with margining enabled enter into a margined state (low or high).
  - This special behavior can be assigned to only one GPI.
- **Input Source for Margin Low and Not-High**: When this pin is asserted, all margined rails are set to Margin Low as long as the Margin Enable is asserted. When this pin is de-asserted the rails are set to Margin High as long as the Margin Enable is asserted.
  - This special behavior can be assigned to only one GPI.
- **Configured as Debug Pin**: When the pin is asserted, the device is put into Debug mode. See GPI Debug Pin for more details.
  - This special behavior can be assigned to only one GPI.
- **Configured as Fault Pin**: GPI fault enable functionality must be set to enable this feature. When set, if there is no fault on a fault bus. The FAULT pin is digital input pin and it monitors the fault bus. When one or more UCD91xxx devices detect a rail fault, the corresponding FAULT pin is turned into active driven low state, pulling down the fault bus voltage and informing all other UCD91xxx devices of the corresponding fault. This behavior allows a coordinated action to be taken across multiple devices. After the fault is cleared, the state of the FAULT pin reverts to that of an input pin. See Cascading for more details.
  - This special behavior can be assigned to up to four GPIs.

2.4.2 GPI Fault Responses

Faults and the corresponding fault responses can be tied to various threshold values i.e. A fault is generated when a rail crosses thresholds set on voltage, temperature, and turn-on and turn-off times. In response to a fault event, the UCD91xxx does the following:

- Asserts the PMBus Alert line
- Logs the fault event into nonvolatile memory and sets the relevant status register bits
- Executes the user-configurable fault responses

See Fault Handling for more details.

On UCD91xxx, GPIs can be configured such that the de-assertion of a GPI is treated as a fault. Fault responses for GPIs are very similar to rail fault responses, except that GPI fault responses do not support the retry action. To enable GPI fault responses, use the following PMBus commands:

- GPI_CONFIG
- GPI_FAULT_RESPONSES

Note

GPI Faults are edge-triggered, and therefore occur only when a pin goes from the asserted state to the deasserted state.

2.4.3 GPI Latched Status Clearing

A GPI can be configured as a Clear Latched Status pin. Latched statuses are used for LGPO functionality, and these statuses can only be cleared in the following ways:

- Using the PMBus command CLEAR_FAULTS
- Using a GPI with the Clear Latched Status functionality enabled

To configure a GPI with this functionality, use the GPI_CONFIG command.

2.4.4 GPI Debug Pin

A GPI can be configured to act as a Debug pin. When this pin is asserted, the device enters Debug mode, which has the following behavior:

- The device does not alert the PMBALERT pin.
- No faults responses are executed. No faults are logged.
- Rail sequence ON and OFF dependency conditions are ignored.
- Rail sequence timeout actions are ignored.
  - As soon as the sequence ON and OFF timeout expires, the rails are sequenced ON or OFF accordingly, regardless of the timeout action. If the sequence ON or OFF timeout value is set to 0, the rails are sequenced ON or OFF immediately.
- The GPI Fault Pins do not pull the fault bus low.
- LGPOs affected by these events return to the original states.
- The System Watchdog function is disabled.

This function is mainly used for debugging purposes and is not recommended for final production.

To configure a GPI pin with this functionality, use the GPI_CONFIG command.

2.4.5 GPI Fault Pin

Up to 4 GPIs can be configured to act as Fault Pins on UCD91xxx. Fault Pins are used for cascading, allowing multiple UCD91xxx devices to take synchronized fault responses. See Cascading Pins and Cascading for more details.

To configure GPIs to act as Fault Pins, use the following commands:

- FAULT_PIN_CONFIG
  - To select which GPIs should act as Fault Pins and select which faulted rails/system faults cause the Fault Pin to de-assert
- GPI_CONFIG
  - To configure the GPIs (pin polarity, etc.) and select which GPIs can trigger a fault
- GPI_FAULT_RESPONSE
  - To configure the fault response of fault-enabled GPIs

2.5 Power Supply Enable Pins

Rail enable pins can be configured with either active-low or active-high polarity. Output mode options include open-drain or push-pull outputs that can be actively driven to 3.3 V or ground. During reset, the GPIO pins are high-impedance except for MARx/GPIO pins.

External pull-down resistors or pull-up resistors can be tied to the enable pins to hold the power supplies off during reset. Refer to the device-specific pinout for the maximum number of supported ENx pins.

Rail enables are used for sequencing purposes. See Power Supply Sequencing for more details.

2.6 Cascading Pins

In advanced applications, the system may require more rails than what a single UCD91xxx device can support. Therefore, in these systems, multiple sequencer devices are required. Using pins, these multiple devices can be cascaded together, to ensure proper sequencing across all devices. See Cascading for more details.

2.7 Margining Pins

UCD91xxx provides PWM output pins for both general-purpose PWMs and closed-loop margining outputs. See Margining for more details.

3 GPI State Machine

3.1 Overview

UCD91xx allows users to use up to 3 GPI pins to control up to eight rail states. Each rail state can have certain rails enabled or disabled. This feature is useful in implementing system low-power modes, such as those compliant with the Advanced Configuration and Power Interface (ACPI) specification.

The GPI State Machine feature is also referred to as Pin-Selected Rail States.

When a new state is present on the GPI pins, and a rail is commanded to turn on, it will attempt to turn on according to its sequence-on dependencies and delays. If a rail is commanded to turn off, the rail can be configured to immediately turn off (immediate off) or attempt to turn off following its sequence-off dependencies and delays (soft off). If a rail is commanded to remain in the same on or off state, no action is taken.

When a rail must change its state, the device modifies the OPERATION command setting for that particular rail. Therefore, for GPI rail state functionality to work, the ON_OFF_CONFIG settings for the relevant rails must be set to depend on OPERATION.

Note: The pin selected rail states feature simply commands rails to turn on/off when entering a state. If the rails have sequencing on and off dependencies that are not met, the rails will continue to wait for those dependencies before actually asserting or de-asserting (if the soft off option is selected) the EN pin.

3.2 GPI State Machine Configuration

Any GPIO-capable pin can be assigned to GPIs 1, 2, or 3, but the first three configured GPIs on UCD91xxx are reserved for the GPI state machine, or pin selected rail states feature.

If less than 8 system states are needed, the system states need to be configured so that changes on GPIs 1 - 3 will not result in execution of unintended system states.

Note

If this functionality is used, then the lower three GPIs should not be used for SEQ_ON/SEQ_OFF dependencies for these rails, to prevent conflicts in sequencing that may arise. Use the upper GPI slots for sequencing dependencies if they are needed.

For example, if only 2 system states are required, GPI 1 is sufficient to cover those 2 states. However, if the device needs more GPIs (for margining, debug mode, etc.), extraneous system states must be configured so that changes on GPIs 2 and 3 will not trigger a state change. In this case, the UCD91xxx should be configured with the following:

- States (GPIs 3, 2, 1 values)
  - 000 - Enabled, rails configured for system state S0 (GPI 1 de-asserted)
  - 001 - Enabled, rails configured for system state S1 (GPI 1 asserted)
  - 010 - Enable. Configure rails to match system state S0
  - 011 - Enable. Configure rails to match system state S1
  - 100 - Enable. Configure rails to match system state S0
  - 101 - Enable. Configure rails to match system state S1
  - 110 - Enable. Configure rails to match system state S0
  - 111 - Enable. Configure rails to match system state S1
- The result of the above configuration essentially makes GPIs 2 and 3 don't cares during state selection.

Similarly, if only 4 system states are required, the following configuration makes GPI 3 a don't care for state selection.

- States
  - 000 - Enabled, rails configured for system state S0
  - 001 - Enabled, rails configured for system state S1
  - 010 - Enabled, rails configured for system state S2
  - 011 - Enabled, rails configured for system state S3

- 100 - Enable. Configure rails to match system state S0
- 101 - Enable. Configure rails to match system state S1
- 110 - Enable. Configure rails to match system state S2
- 111 - Enable. Configure rails to match system state S3

Note: When selecting a new system state, the state changes on the GPI pins must be completed with 1us, otherwise an unintended system state may be selected.

4 Monitoring

UCD91xxx can monitor analog inputs, including voltages and temperature, and digital inputs (POWER_GOOD). See MONITOR_CONFIG for more details.

5 Rail Profiles

When monitoring the voltage of a particular rail, a fault or warning event occurs when the voltage exceeds the user-configurable voltage thresholds. When a fault is detected, the device responds with user-defined actions (fault response).

For faster response and more flexible configuration, UCD91xxx devices support the Rail Profile feature, which is only available for rails monitoring voltage.

Each rail can have up to for profiles, where the application can switch between profiles using two assigned GPIs. A Rail Profile consists of nine thresholds, set by the following PMBus commands:

- VOUT_COMMAND
- VOUT_OV_FAULT_LIMIT
- VOUT_OV_WARNING_LIMIT
- VOUT_MARGIN_HIGH
- POWER_GOOD_ON
- VOUT_MARGIN_LOW
- POWER_GOOD_OFF
- VOUT_UV_WARNING_LIMIT
- VOUT_UV_FAULT_LIMIT

UCD91xxx offers a total of 50 individual profiles shared among all rails, where each rail can have at least one, but not more than four profiles. Switching between rails is controlled by two GPIs, and a programmable block-out period is used to block all voltage-related faults on the given rail when switching profiles.

The following PMBus commands are used for Rail Profile configuration:

- RAIL_PROFILE
- GPI_FAULT_RESPONSE
  – For configuration the GPIs used for switching and the switching block-out period

6 Margining

6.1 Overview

The UCD91x devices implement optional closed-loop margining for up to 32 voltage rails. During four-corner testing, a system operates at the minimum and maximum expected ambient temperature and with each power supply set to the minimum and maximum output voltage, commonly referred to as margining. Margining can be activated, and switched from low to high level margining via the PMBus interface using the OPERATION command, or by configuring two GPI pins as MARGIN-EN and MARGIN-LOW_nHIGH inputs with GPI_CONFIG.

The MARGIN_CONFIG command allows for the setting of several margining options, including ignoring faults while margining, and using closed-loop pwm feedback to trim the rail output voltage when not margining. In order to accommodate positive and negative feedback relationships, the command also allows the user to invert the relationship between the PWM Duty cycle. The specific PWM pin for a given page, and its behavior when margining is not active, is also configured with this command.

The UCD91x devices also allow for the PWM duty cycle when margining to operate in a nominal duty cycle mode, wherein the PWM duty cycle will always start from the nominal duty cycle set in the PWM_CONFIG command for the given PWM Output. When margining is stopped or disabled on a rail with nominal duty cycle mode enabled, the PWM will also return to the PWM_CONFIG command duty cycle before fully stopping. When not in nominal duty cycle mode, margining and active trimming, the PWM duty cycle will start from 0% and slowly ramp up, and disable immediately. Using nominal duty cycle mode can help prevent glitches on the power supply outputs as margining is enabled/disabled.

When set to ignore faults while margining, no over or undervoltage faults will be triggered or handled while a margining state is active. Both the OPERATION command and MARGIN_CONFIG commands can be used to dictate the UCD91xx fault status update behavior when margining. The most recently written command will determine if faults are ignored or not.

6.2 Operation

The figure below shows the block diagram of single margining circuit. An external R-C network converts the PWM pulses into a DC margining voltage. The margining voltage is connected to the power supply feedback node through a resistor. The feedback node voltage is thus slightly pulled up or down by the margining voltage, causing the rail output voltage to change. The UCD91x device monitors the rail output voltages. The device adjusts the margining PWM duty cycle accordingly such that the rail output voltage is regulated at the VOUT_MARGIN_HIGH or VOUT_MARGIN_LOW voltages defined by the user. Effectively, the margin control loop of the UCD91x device overwrites the DC set point of the margined power supply.

Block Diagram of Single Margining Circuit

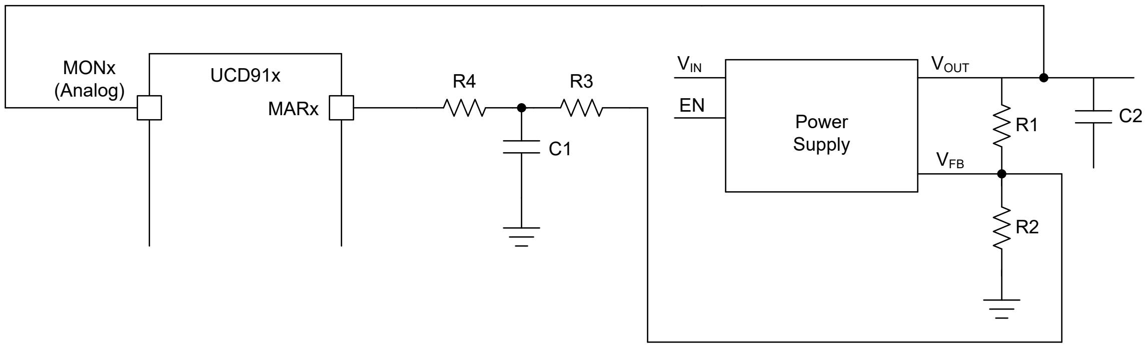


Figure 6-1. Margining Circuit

The margin control loop should be as slow as possible in order in order to not minimize interference with the power supply control loop, and reduce the likelihood that instability is introduced. As such, the adjustments to the duty cycle of the PWM are done only incrementally. The first 8 MARx pins of UCD91x make use of an 80MHz PWM clock to generate their output, the second 8 MARx pins make use of a 40MHz oscillator as the PWM clock. The effective number of quantization steps is given by the below relationship:

$$n_{DutyCycle} = \frac{F_{CLK}}{F_{PWM}} \quad (1)$$

Where:

- n is the number of duty cycle quantization steps available
- FCLK is the PWM clock frequency
- FPWM is the frequency

The upper and lower bounds of the output voltage when margining is determined by the sizing of the filter resistor (R4), and the isolation resistor (R3). The combination of these, and number of duty cycle steps available, determines the minimum step in output voltage that is possible. For more information please see the Voltage Margining Design Guide.

6.3 Idle Behavior of Margining Pins

When not margining, a margining, or MARx, pin can be configured to operate in one of three modes:

- Tri-state
- Active trim
- Fixed Duty Cycle

Tri-state mode sets the margin pin to high-impedance. Active Trim mode continuously trims the DC output voltage to try and optimize it to the VOUT_COMMAND set voltage, and fixed duty cycle mode provides a user-defined fixed PWM duty cycle output. The PWM Duty Cycle setting for fixed duty cycle mode can be configured via the PWM_CONFIG command.

7 Cascading

7.1 Overview

In advanced applications, the system may require more rails than what a single UCD91xxx device can support. Therefore, in these systems, multiple sequencer devices are required.

There are different types of cascading, depending on an application's needs.

- Power On Sequencing
- Power On and Off Sequencing
- Coordinated Fault Response

7.2 Power On Cascading

For systems where only power-on sequencing is a concern, the general approach is to:

1. Configure an LGPO to output a device's POWER_GOOD.
2. Connect the LGPO POWER_GOOD signal to another device.
   a. This allows the rails of a device to have a dependency on the power-good output of the previous device in the chain, and imposes a controller-target relationship between multiple devices.
3. Optionally, connect the LGPO POWER_GOOD signal of the last device back to the MONx or GPIx pin of the first device.

There are two different approaches for how to connect a device's POWER_GOOD signal to another device.

1. Connect the LGPO POWER_GOOD signal to the PMBUS_CNTRL pin of another device.
2. Connect the LGPO POWER_GOOD signal to a MONx or GPIx pin of another device.

At startup, once the controller has completed its start sequence, and all of the controller's rails have reached regulation voltages, the target devices can initiate their own start sequences.

During shutdown, as soon as the controller starts to sequence-off, the controller's POWER_GOOD signal will de-assert, which initiates the shutdown of the target devices.

- Since POWER_GOOD is a signal that depends on **all** configured rails reaching regulation voltages, a shutdown on one or more of the controller's rails can initiate the shutdown of the target devices.
- The controller's shutdown can be initiated intentionally or by a fault condition.
- If a fault condition occurs on a target device, the target device's rails will shutdown (if shutdown fault response is enabled), and the target's POWER_GOOD signal is also de-asserted.
- If a fault condition occurs on the last device in the chain, the last device will de-assert its POWER_GOOD signal. The first device in the chain (the controller) will treat this in one of two ways, depending on whether the POWER_GOOD signal is connected to MON or GPI pin.
  – If connected to a MON pin, the controller will treat the de-assertion as an UV fault.
  – If connected to a GPI pin, the controller will treat the de-assertion as a GPI fault. GPIs can be used if not enough MON pins are available.

7.3 Power On and Off Cascading

If Power Off sequencing is required, an additional LGPO is needed. The new LGPO should be configured to output POWER_GOOD_OFF when all rails controlled by the device are below the POWER_GOOD_OFF threshold. This indicates that the rails are properly shutdown.

The upstream UCD can take this signal in via GPI pin and use this as part of its sequencing off dependencies.

When the controller de-asserts its POWER_EN pin to power down the entire system, the rails controlled by the last UCD in the chain will shut off first. Then, all rails controlled by the first UCD device (the controller) will shut off after all target devices' rails are off.

7.4 Fault Cascading

Up to four GPIs can be configured as Fault Pins, where each Fault Pin is connected to a Fault Bus. Each Fault Bus is pulled up to 3.3V by a 10k resistor, and all UCD91xxx devices on the same Fault Bus are informed of the same fault condition.

Where there aren't any faults on a Fault Bus, the Fault Pins act as digital inputs and listen to the bus. When one or more UCD91xxx devices detects a fault, the corresponding Fault Pin pulls down the Fault Bus, which informs all other devices that a fault occurred. A coordinated fault response can be performed. After the fault is cleared, the Fault Pin goes back to acting as a digital input pin.

7.5 Cascading Requirements

The following features are required to implement cascading:

- LGPOs
  - For power on cascading, an LGPO must be configured to output POWER_GOOD.
  - For power on and off cascading, an additional LGPO must be configured to output POWER_GOOD_OFF.
- Sequencing on and off dependencies
- UV Fault Response
  - For power cascading purposes, when the POWER_GOOD output of the target device is fed back via MON pins.
- GPI Fault Response
  - For power cascading purposes, when POWER_GOOD and/or POWER_GOOD_OFF signals are fed back via GPI pins.
- Fault Pin Configuration
  - For fault cascading purposes.
- SYNC_CLK
  - For fault cascading purposes.

8 Fault Handling

In the previous sections, various fault and warn notification thresholds have been configured to monitor voltage and temperature, and turn-ON time and turn-OFF time. When a fault threshold is reached, a fault event occurs. The device performs the following three actions in response of a fault event.

- Asserts the PMBus ALERT line
- Logs the fault event into nonvolatile memory (data flash), set status register bit
- Executes fault responses defined by users

The Fault Responses can be configured using FAULT_RESPONSES.

A programmable glitch filter can be enabled or disabled for each type of fault. When a fault remains present after the glitch filter time expires, the device performs of the three selectable actions:

- Log the fault and take no further action.
- Log the fault and shut down the rail immediately.
- Log the fault and shut down the rail with Turn Off Delay.

After shutting down the rail, the device performs one of the three selectable actions:

- Do not restart the rail until a new turn-on command is received.
- Restart the rail. If the restart is unsuccessful, retry up to a user-defined number of times (up to a maximum of 14) and then remain off until the fault is cleared.
- Restart the rail. If the restart is unsuccessful, retry for an unlimited number of times unless the rail is commanded off by a signal defined in On/Off Config. After the rail exhausts the restart attempts, Re-sequence can be initiated (see the Rail Sequence Configuration section).

When the rail enable is turned off to restart, the device will not wait for sequence off dependencies. If the Rail is already under retry due to a fault, and if a new fault occurs in the same rail, the device behavior is always conservative. i.e. the minimum count between the ‘remaining retry count of existing fault’ and the ‘configured retry count of the new fault’ will be taken.

For more details on Re-sequence refer to Power Supply Sequencing.

Voltage, current, and temperature monitoring are based on results from the ADC(AMON) and DMON. The ADC results are compared with the programmed thresholds. The time to respond to an individual event is determined by when the event occurs within the ADC conversion cycle and the configured fault responses (glitch filters, time delays, and so forth).

GPI pins can also trigger faults if desired. The GPI Fault Responses options are the same as the Fault Responses discussed earlier in this section, with one exception: the GPI Fault Responses option does not support the retry action. See GPI_FAULT_RESPONSES.

Commands used to configure the fault response are:

- FAULT_RESPONSES – This paged command configures the response to each fault condition
- GPI_FAULT_RESPONSES – Configures the response to GPI faults
- MISC_CONFIG – Configures re-sequence related parameters
- SMBALERT_MASK – To mask the faults from asserting the PMBALERT line
- CLEAR_FAULTS – To clear the fault status, and de-assert the PMBALERT line

9 Fault Logging

The device provides fault log and device reset counter readings. When a fault occurs, the data will be logged to the flash immediately.

Faults are stored in Flash memory and are accessible over PMBus. Each logged fault includes the following information:

- Rail number
- Fault type
- Fault time since previous device reset
- Last measured rail voltage

The total number of device resets is also stored to Flash memory. The value can be reset using PMBus.

Commands used are,

- LOGGED_FAULTS – Returns the faults logged into Flash memory
- LOGGED_FAULT_DETAIL_INDEX – Returns the number of fault entries in the Flash
- LOGGED_FAULT_DETAIL – Returns the details of the given fault in the specific page
- LOG_FAULT_DETAIL_ENABLES – Used to enable/disable fault logging to flash memory, for the specific fault types and page
- MISC_CONFIG – Configures FIFO mode for fault logs

10 Memory

10.1 Overview

As per the PMBus requirements, the UCD91xxx must follow a particular memory storage and loading scheme.

The PMBus requirements are as follows:

- PMBus devices must be able to start up without bus communication.
- PMBus devices can be used both with or without a power system manager or controller.
- PMBus devices should support a "set and forget" scheme, where they can be programmed once at time of manufacture and operate forever without bus communication.
- Default PMBus variables, in order to support the above requirements, should be loaded from non-volatile memory or through hardware pin-programming (if any). See Memory Model for more details on memory loading precedence.

The following sections will discuss the memory organization of the UCD91xxx and how the device interacts with the memory during different stages of operation.

10.2 Flash Memory

The UCD91xxx flash memory is composed of the following sections: Program flash, where the device firmware is stored, and Data flash, where the user configuration data is stored.

Program flash is written during Texas Instruments' manufacturing process. It contains device firmware, what makes the UCD91xxx device behave as a digital power sequencer.

Data flash holds non-volatile device configuration and other operational data. The UCD91xxx devices also log information about faults to data flash.

The UCD91xxx stores the firmware, configuration settings, and fault log information in flash. The configuration data settings will have a main and backup copy in memory. Each section has its own unique cyclic redundancy check (CRC).

10.3 Power Up

On device power-up, the device performs verification on the following items:

- Sequencer firmware
- PMBus configuration data

See Device Reset and Internal Fault Management for more details.

10.4 Program Lifetime

When the host (or controller) writes PMBus configuration data to the target, it does not write directly to target's flash memory. Instead, the host writes to the target's volatile memory (RAM). The target device accepts these incoming commands and maps these to an area in RAM, possibly translating the configuration data to an internal device-specific format.

Since the PMBus command writes are to RAM, engineers can safely test changes, and if there is an error, they can recover the previous configuration by issuing a device reset. When an engineer determines that the configuration is valid, they can commit the configuration to non-volatile flash memory by issuing the STORE_DEFAULT_ALL PMBus command, which handles copying parts of RAM to flash memory.

11 Internal Fault Management

The device verifies the firmware by using a checksum algorithm at each power up. If the checksum does not match, the device resets. If the device continues to reset, the SYNC_CLK pin outputs repeated pulses with an approximate 250-ms pulse width that can be observed externally.

The device performs a configuration checksum verification at power up. If the checksum does not match, the device discards all the configuration data. The PMBALERT pin is asserted and a flag is set in the status register.

A fault-log checksum verification in Flash is also performed at power up. Each log entry has a checksum. The device ignores the corrupted log entries.

12 Status Monitoring

The device has status registers for each rail. Faults and warnings are logged into flash memory to assist system troubleshooting. The status registers and the fault log can be accessed from the TI Sequencer Studio GUI as well as the PMBus interface.

See MFR_STATUS and the PMBus 1.2 Specification for detailed descriptions of each status register and supported PMBus commands.

- STATUS_WORD – Returns 2 bytes with a summary of unit's fault condition
- STATUS_BYTE – Returns 1 byte with a summary of critical faults. It is lower byte of STATUS_WORD
- STATUS_VOUT – Returns Voltage faults of the specific Rail
- STATUS_IOUT – Returns Current faults of the specific Rail
- STATUS_TEMP – Returns Temperature faults of the specific Rail
- STATUS_CML – Returns the communication fault status
- STATUS_MFR – Returns Manufacturer specific faults. This includes GPI faults, Sequence on/off timeout status, Memory faults etc.

13 Device Reset

UCD91xxx has an integrated power-on reset (POR) circuit which monitors the supply voltage. At power up, the POR detects the V33D pin voltage rise. When the V33D voltage is greater than VRESET, the device comes out of reset.

The device can be forced into the reset state by an external circuit connected to the $\overline{\text{RESET}}$ pin. A logic-low voltage on this pin for longer than tRESET sets the device into reset state. The device comes out of reset within tIRT after RESET is released to logic-high level.

Any time the device comes out of reset, it begins an initialization routine that lasts typically 40 ms. A data flash checksum verification is performed at power up. If the checksum verification does not match, the device configuration settings are cleared, the PMBALERT pin is asserted, and a flag is set in the status register. A faultlog checksum verification is also performed at power up. Each log entry includes the checksum verification status. Only a corrupted log entry is discarded. During the initialization routine, all I/O pins are held at high impedance state. At the end of initialization, the device begins normal operation as defined by the device configuration.

14 ADC Reference

Using the V33A pin as ADC reference voltage by default provides a cost-effective solution. However, internal voltage reference has a higher Total Unadjusted Error. Also, voltage variations on the V33A pin affect ADC readings, such as when the device is powered down. To achieve better ADC accuracy, an external voltage reference can be connected to the VREFA+ and VREFA- pins. Ensure that the external reference voltage stays in regulation whenever V33D is above VBOR threshold. This limitation allows accurate ADC readings in full V33D operating range.

The external reference voltage level must be configured in the GUI to give correct ADC readings. See MISC_CONFIG for more details.

15 System Watchdog

UCD91xxx provides a System Watchdog timer (WDT), which can be reset by toggling a watchdog input (WDI) pin. If WDI is not toggled within a programmed period, the WDT times out. As a result, a watchdog output (WDO) pin is asserted (generates a pulse) to provide a system reset signal.

The WDI and WDO pins are GPIO pins and are optional.

- WDI can be replaced by PMBus command SYSTEM_WATCHDOG_RESET.
- WDO can be manifested through LGPO, or its function can be integrated into the System Reset pin (RESET) configured by the System Reset function. See the System Reset for more details.

After a timeout, the WDT can be restarted by toggling the WDI pin or by sending SYSTEM_WATCHDOG_RESET.

The System Watchdog timer can be immediately active at device power-up or after an initial wait time. See SYSTEM_WATCHDOG_CONFIG for configuration details.

16 System Reset

The System Reset function can generate a programmable system reset signal through a GPIO pin. The System Reset signal is de-asserted when the selected rail voltages reach their respective Power Good On thresholds, and the selected GPIs are asserted, plus a programmable delay time.

The System Reset signal can be asserted immediately when any of the selected rail voltage falls below Power Good Off threshold, or any selected GPI is de-asserted. Alternatively, the System Reset signal can be configured as a pulse once Power Good On is achieved.

The System Reset signal can also integrate the System Watchdog timer. The watchdog is configured with a Start Time and a Reset Time. If these times expire and timeout occurs, it means that the CPU providing the WDI signal is not operating. The System Reset signal is then toggled either using a Delay or GPI Tracking Release Delay to determine if the CPU recovers.

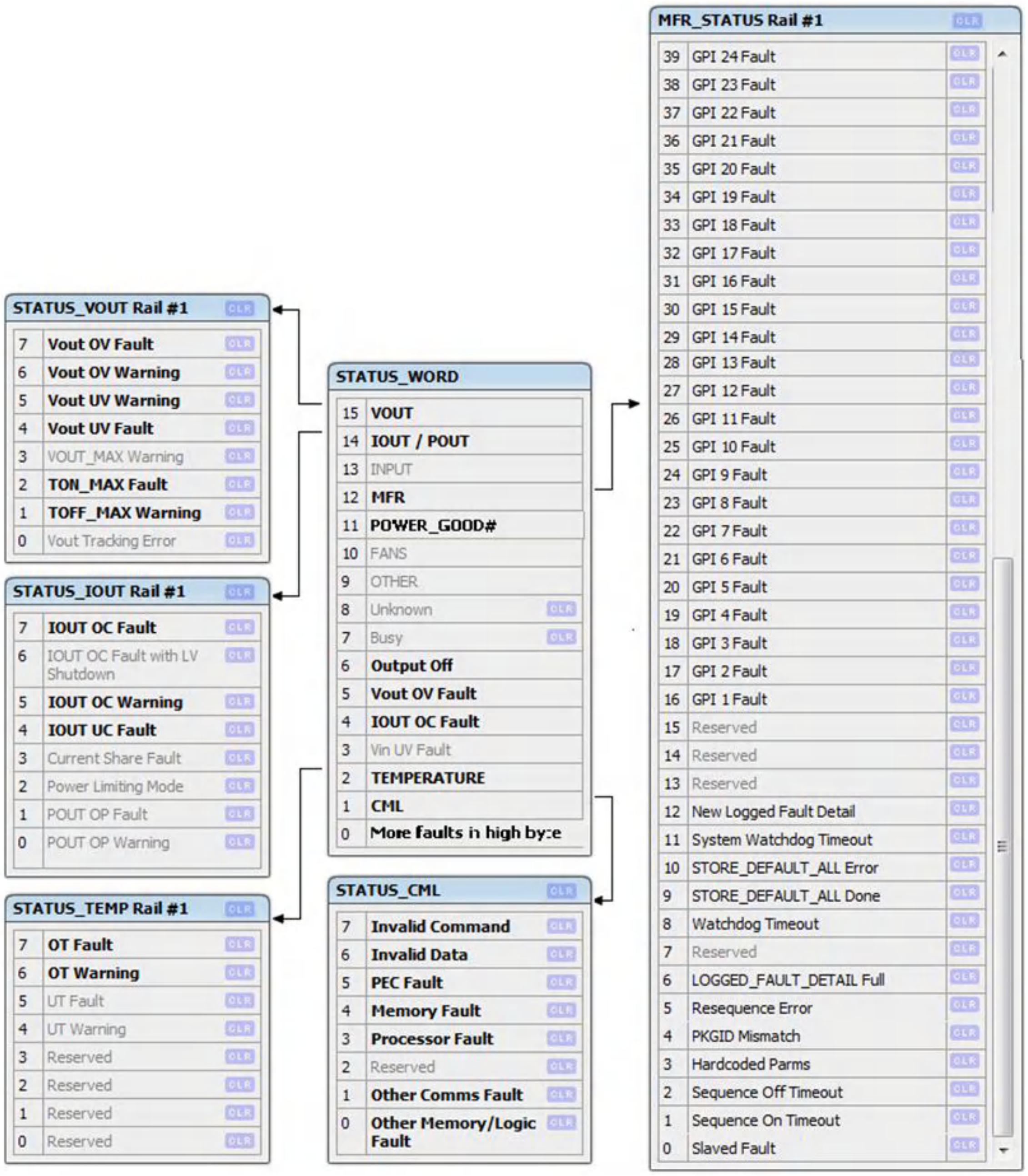
The default state of the System Reset pin is assert. When the System Reset function is configured in-circuit through PMBus commands during normal operation, the pin is briefly asserted by default, even if conditions for de-assert are present. This is because the firmware requires a finite time to examine the de-assert conditions.

See SYSTEM_RESET_CONFIG for more configuration details.

17 PMBus Specification

This document makes frequent mention of the PMBus specification, specifically, the *PMBus Power System Management Protocol Specification Part II – Command Language*, Revision 1.2. The specification is published by the Power Management Bus Implementers Forum and is available from http://pmbus.org.

The types of status supported by the UCD91xxx are shown below. Whenever any of these bits are set, the PMBALERT# line is asserted. Unsupported types are denoted with grayed-out text.



STATUS_VOUT Rail #1	
7	Vout OV Fault
6	Vout OV Warning
5	Vout UV Warning
4	Vout UV Fault
3	VOUT_MAX Warning
2	TON_MAX Fault
1	TOFF_MAX Warning
0	Vout Tracking Error

STATUS_IOUT Rail #1	
7	IOUT OC Fault
6	IOUT OC Fault with LV Shutdown
5	IOUT OC Warning
4	IOUT UC Fault
3	Current Share Fault
2	Power Limiting Mode
1	POUT OP Fault
0	POUT OP Warning

STATUS_TEMP Rail #1	
7	OT Fault
6	OT Warning
5	UT Fault
4	UT Warning
3	Reserved
2	Reserved
1	Reserved
0	Reserved

STATUS_WORD	
15	VOUT
14	IOUT / POUT
13	INPUT
12	MFR
11	POWER_GOOD#
10	FANS
9	OTHER
8	Unknown
7	Busy
6	Output Off
5	Vout OV Fault
4	IOUT OC Fault
3	Vin UV Fault
2	TEMPERATURE
1	CML
0	More faults in high byte

STATUS_CML	
7	Invalid Command
6	Invalid Data
5	PEC Fault
4	Memory Fault
3	Processor Fault
2	Reserved
1	Other Comms Fault
0	Other Memory/Logic Fault

MFR_STATUS Rail #1	
39	GPI 24 Fault
38	GPI 23 Fault
37	GPI 22 Fault
36	GPI 21 Fault
35	GPI 20 Fault
34	GPI 19 Fault
33	GPI 18 Fault
32	GPI 17 Fault
31	GPI 16 Fault
30	GPI 15 Fault
29	GPI 14 Fault
28	GPI 13 Fault
27	GPI 12 Fault
26	GPI 11 Fault
25	GPI 10 Fault
24	GPI 9 Fault
23	GPI 8 Fault
22	GPI 7 Fault
21	GPI 6 Fault
20	GPI 5 Fault
19	GPI 4 Fault
18	GPI 3 Fault
17	GPI 2 Fault
16	GPI 1 Fault
15	Reserved
14	Reserved
13	Reserved
12	New Logged Fault Detail
11	System Watchdog Timeout
10	STORE_DEFAULT_ALL Error
9	STORE_DEFAULT_ALL Done
8	Watchdog Timeout
7	Reserved
6	LOGGED_FAULT_DETAIL Full
5	Resequence Error
4	PKGID Mismatch
3	Hardcoded Parms
2	Sequence Off Timeout
1	Sequence On Timeout
0	Slaved Fault

Figure 17-1. PMBus Status Supported by UCD91160

17.1 Manufacturer Specific Status (STATUS_MFR_SPECIFIC)

The standard STATUS_MFR_SPECIFIC command does not have enough bits to implement UCD91xxx's functionalities. This command has been replaced by the MFR_STATUS command (see Section 26.33).

18 Data Formats

Sections 7 and 8 of the PMBus standard provides for five different data formats: three for parameters related to output voltage and two for all other commands. Each PMBus device is expected to support only one of these formats.

18.1 Data Format for Output Voltage Parameters

For parameters related to output voltage, the UCD91xxx supports the linear format defined in Section 8.3.1 of the PMBus specification. The linear format uses a 16-bit unsigned mantissa for each parameter, along with an exponent that is shared by all the voltage-related parameters. The exponent is reported in the bottom 5 bits of the VOUT_MODE parameter.

The voltage value is calculated using the equation

$$\text{Voltage} = V \times 2^X \tag{2}$$

Where:

Voltage is the parameter of interest, in volts,
V is a 16-bit unsigned binary integer mantissa, and
X is the signed 5-bit twos-complement binary integer exponent from VOUT_MODE.

Exception: The PMBus standard assumes that all output voltages are expressed as positive numbers; therefore, all parameters related to output voltage are unsigned integers, with a few notable exceptions. The VOUT_CAL_OFFSET, and VOUT_CAL_MONITOR values are intended for making fine adjustments to the output voltage and can take on small negative values. As such, these parameters are treated as signed twos-complement binary integers.

18.2 Data Format for Other Parameters

For parameters not directly related to output voltage, the UCD91xxx supports the linear data format described in section 7.1 of the PMBus specification. This linear format is a two-byte value that contains an 11-bit, twos-complement mantissa and a 5-bit, twos-complement exponent.

The relationship between the PMBus parameter and the real-world value is given by the formula:

$$R = Y \times 2^X \tag{3}$$

Where:

R is the real-world value,
Y is an 11-bit, signed twos-complement binary integer mantissa, and
X is the signed 5-bit, `twos-complement binary integer exponent.

This pseudo floating-point notation allows values as large as ≈33E6 down to ≈15E-6 to be sent over the PMBus. The internal variables used by the UCD91xxx firmware are mostly 16 bits wide and do not support such a wide range of values. The resolution of a PMBus setting depends strongly on both the exponent of the PMBus value (larger values have coarser resolution) and the scaling of the internal variables.

18.3 Distinguishing Between Linear Data Formats

The PMBus specification uses the same term, linear, to describe both the 16-bit+exponent format used for the voltage-related parameters as well as the 11-bit+exponent format used for other parameters. In cases where it is necessary to distinguish between these two data formats, this document uses the term LINEAR16 or LINEAR11.

18.4 Translation, Quantization, and Truncation

The internal variables used by the UCD91xxx are often scaled to take optimal advantage of the native units such as ADC or DAC counts rather than volts or amperes. As a result, values that are written and read by PMBus must undergo mathematical translations. These translations, with their inherent quantization, can result in very slight differences between the setting that was written to the UCD91xxx and the value that was later read back from it. This is normal and compliant, described in section 7.4 of the PMBus specification.

In some cases, a value written to the device can cause it to exceed the range of its internal variable. In some cases, the device reports this as an error; in other cases it saturates the variable at a safe value. In all cases, the value read back by the PMBus reflects as accurately as possible the internal variable actually being used by the UCD91xxx.

18.5 8-Bit Time Encoding

To save memory when precision is not needed, several commands encode a time parameter in 8-bits. The 8 bits are separated into two fields. Bits 6 and 7 are the index for the multiplier field and bits 0 to 5 are the mantissa (0x00 to 0x3F). The two are multiplied together to derive the time. A mantissa of 0 results in a time of 0 regardless of the multiplier value.

Table 18-1. 8-Bit Time Encoding

Multiplier Index	Multiplier (ms)	Resulting Range
b'00	1	0 to 63 ms
b'01	8	8 to 504 ms
b'10	64	64 ms to 4.032 s
b'11	512	512 ms to 32.256 s

18.6 16-Bit Time Encoding

The 16 bits are separated into two fields. Bits 14 and 15 are the index for the multiplier field and bits 0 to 13 are the mantissa (0x00 to 0x3FFF). The two are multiplied together to derive the time. A mantissa of 0 results in a time of 0, regardless of the Increment value.

Table 18-2. 16-Bit Time Encoding

Multiplier Index	Multiplier (ms)	Resulting Range
b'00	1	1 to 16384 ms
b'01	8	8 to 131.72 s
b'10	64	64 ms to 1048.576 s
b'11	512	512 ms to 8388.608 s

19 Memory Model

Section 6 of the PMBus specification describes the memory model for PMBus devices. Values used by the PMBus device are loaded into volatile Operating Memory from one or more of the following places:

- Values hard coded into an integrated circuit (IC) design (if any)
- Values programmed from hardware pins (if any)
- A nonvolatile memory called the Default Store
- A nonvolatile memory called the User Store (not supported by the UCD91xxx)
- Communications from PMBus

The UCD91xxx contains RAM that is used as Operating Memory. Embedded Data Flash memory is used to implement the hard-coded values and the Default Store values. Hard-coded values require a new firmware revision. Values in the Default Store can be changed using the STORE_DEFAULT_ALL command described in Section 22.5.

Section 6.1 of the PMBus specification describes the ordering of memory loading and precedence. In general, the hard-coded parameters are loaded into Operating Memory first. Second, any pin-programmable settings take effect. Third, values from the Default Store are loaded. Later, commands issued from the PMBus take effect. In all cases, an operation on a parameter overwrites any prior value that was already in the Operating Memory.

20 Alert Response Address Support

The UCD91xxx supports using the PMBALERT# line to notify the host of warning or fault conditions. It does not support the Alert Response Address protocol.

21 Supported PMBus Commands

Table 21-1 lists the PMBus commands. Commands 00h through CFh are defined in the PMBus specification and are considered to be core commands that are standardized for all manufacturers and products. For details on how core commands are implemented, please refer to the PMBus specification. Commands D0h through FEh are manufacturer specific and can be unique for each manufacturer and product.

The device columns indicate if a command is supported by a given device. The command is supported if a check mark (✓) appears in that column. Commands that are not supported by any device are grayed out.

Most commands support writing and reading. Exceptions are indicated in the Comments column.

The Data Format column indicates the format of the data:

Byte	8-bit binary value. See the PMBus specification for details for each command.
LINEAR16	16-bit linear format used for output voltage parameters. Described in Section 18.1.
LINEAR11	11-bit linear format used for parameters other than output voltage. Described in Section 18.2.
n/a	Command does not have a data field.
String	ASCII string. Described in Section 22.2 of the PMBus specification.
Byte Array	A block of data in binary format.

The Scope column indicates how each command is affected by the PAGE setting.

Common	This command does not depend on the PAGE setting. It is a common variable used by all pages.
PAGE	This command applies to the pages set by the most recent PAGE command. See Section 22.1 for details.

The Page column in Table 21-1 points to additional detail about the command. A number that is not in parenthesis corresponds to the page in the *PMBus Power System Management Protocol Specification Part II – Command Language*, Revision 1.2. The number in parenthesis is the page number in this document.

The Data Flash column indicates if the parameter is stored in the Default Store in Data Flash. If an x appears in that column, it is not stored. See Section 22.5 for more information.

Most commands are used for device configuration which is often only done once with the assistance of the TI provided configuration GUI. In normal operation, only a subset of the commands are used frequently. Those commands are highlighted in **bold** font.

Table 21-1. PMBus Commands

Code (hex)	Command	Transaction Type	Data Format [Units]	Scope	UCD91160	UCD91320	Data Flash	Comments
00	PAGE	R/W Byte	Byte	Common	✓	✓	X	
01	OPERATION	R/W Byte	Byte	PAGE	✓	✓	X	
02	ON_OFF_CONFIG	R/W Byte	Byte	PAGE	✓	✓		
03	CLEAR_FAULTS	Send Byte	n/a	Common	✓	✓		Write Only
04	PHASE							
05-0F	Reserved							
10	WRITE_PROTECT							
11	STORE_DEFAULT_ALL	Send Byte	n/a	Common	✓	✓		Write Only(2)
12	RESTORE_DEFAULT_ALL	Send Byte	n/a	Common				Write Only
13	STORE_DEFAULT_CODE							
14	RESTORE_DEFAULT_CODE							
15	STORE_USER_ALL							
16	RESTORE_USER_ALL							
17	STORE_USER_CODE							
18	RESTORE_USER_CODE							
19	CAPABILITY	Read Byte	Byte	Common	✓	✓		Read Only
1A	QUERY							

Table 21-1. PMBus Commands (continued)

Code (hex)	Command	Transaction Type	Data Format [Units]	Scope	UCD91160	UCD91320	Data Flash	Comments
1B-1F	Reserved							
20	VOUT_MODE	R/W Byte	Byte	PAGE	✓	✓		The mode is fixed at 000 (Linear Mode), but the exponent can be modified. See Section 18.1 and Section 22.7 for details.
21	VOUT_COMMAND	R/W Word	LINEAR16 [V]	PAGE	✓	✓		
22	VOUT_TRIM							
23	VOUT_CAL_OFFSET							
24	VOUT_MAX							
25	VOUT_MARGIN_HIGH	R/W Word	LINEAR16 [V]	PAGE	✓	✓		
26	VOUT_MARGIN_LOW	R/W Word	LINEAR16 [V]	PAGE	✓	✓		
27	VOUT_TRANSITION_RATE							
28	VOUT_DROOP							
29	VOUT_SCALE_LOOP							
2A	VOUT_SCALE_MONITOR	R/W Word	LINEAR11 [V/V]	PAGE	✓	✓		
2B-2F	Reserved							
30	COEFFICIENTS							
31	POUT_MAX							
32	MAX_DUTY							
33	FREQUENCY_SWITCH							
34	Reserved							
35	VIN_ON							
36	VIN_OFF							
37	INTERLEAVE							
38	IOUT_CAL_GAIN	R/W Word	LINEAR11 [mV/A = mΩ]	PAGE				Current Sense Gain
39	IOUT_CAL_OFFSET	R/W Word	LINEAR11 [A]	PAGE				
3A	FAN_CONFIG_1_2							
3B	FAN_COMMAND_1	R/W Word	LINEAR11 [%]	Common				
3C	FAN_COMMAND_2	R/W Word	LINEAR11 [%]	Common				
3D	FAN_CONFIG_3_4							
3E	FAN_COMMAND_3	R/W Word	LINEAR11 [%]	Common				
3F	FAN_COMMAND_4	R/W Word	LINEAR11 [%]	Common				
40	VOUT_OV_FAULT_LIMIT	R/W Word	LINEAR16 [V]	PAGE	✓	✓		
41	VOUT_OV_FAULT_RESPONSE	R/W Byte	Byte	PAGE				See FAULT_RESPONSES command
42	VOUT_OV_WARN_LIMIT	R/W Word	LINEAR16 [V]	PAGE	✓	✓		
43	VOUT_UV_WARN_LIMIT		R/W Word	LINEAR16 [V]	✓	✓	✓	
44	VOUT_UV_FAULT_LIMIT		R/W Word	LINEAR16 [V]	✓	✓	✓	
45	VOUT_UV_FAULT_RESPONSE	R/W Byte	Byte	PAGE				See FAULT_RESPONSES command
46	IOUT_OC_FAULT_LIMIT	R/W Word	LINEAR11 [A]	PAGE				(1)
47	IOUT_OC_FAULT_RESPONSE	R/W Byte	Byte	PAGE				See FAULT_RESPONSES command
48	IOUT_OC_LV_FAULT_LIMIT							
49	IOUT_OC_LV_FAULT_RESPONSE							
4A	IOUT_OC_WARN_LIMIT	R/W Word	LINEAR11 [A]	PAGE				(1)
4B	IOUT_UC_FAULT_LIMIT	R/W Word	LINEAR11 [A]	PAGE				(1)

Table 21-1. PMBus Commands (continued)

Code (hex)	Command	Transaction Type	Data Format [Units]	Scope	UCD91160	UCD91320	Data Flash	Comments
4C	IOUT_UC_FAULT_RESPONSE	R/W Byte	Byte	PAGE				See FAULT_RESPONSES command
4D	Reserved							
4E	Reserved							
4F	OT_FAULT_LIMIT	R/W Word	LINEAR11 [°C]	PAGE	✓	✓		
50	OT_FAULT_RESPONSE	R/W Byte	Byte	PAGE				See FAULT_RESPONSES command
51	OT_WARN_LIMIT		R/W Word	LINEAR11 [°C]	✓	✓		
52	UT_WARN_LIMIT						✓	
53	UT_FAULT_LIMIT							
54	UT_FAULT_RESPONSE							
55	VIN_OV_FAULT_LIMIT							
56	VIN_OV_FAULT_RESPONSE							
57	VIN_OV_WARN_LIMIT							
58	VIN_UV_WARN_LIMIT							
59	VIN_UV_FAULT_LIMIT							
5A	VIN_UV_FAULT_RESPONSE							
5B	IIN_OC_FAULT_LIMIT							
5C	IIN_OC_FAULT_RESPONSE							
5D	IIN_OC_WARN_LIMIT							
5E	POWER_GOOD_ON	R/W Word	LINEAR16 [V]	PAGE	✓	✓		
5F	POWER_GOOD_OFF	R/W Word	LINEAR16 [V]	PAGE	✓	✓		
60	TON_DELAY	R/W Word	LINEAR11 [ms]	PAGE	✓	✓		This does not apply to retries
61	TON_RISE							
62	TON_MAX_FAULT_LIMIT	R/W Word	LINEAR11 [ms]	PAGE	✓	✓		Maximum time to reach POWER_GOOD_ON
63	TON_MAX_FAULT_RESPONSE	R/W Byte	Byte	PAGE				See FAULT_RESPONSES command
64	TOFF_DELAY	R/W Word	LINEAR11 [ms]	PAGE	✓	✓		
65	TOFF_FALL							
66	TOFF_MAX_WARN_LIMIT	R/W Word	LINEAR11 [ms]	PAGE	✓	✓		
67	Reserved							
68	POUT_OP_FAULT_LIMIT							
69	POUT_OP_FAULT_RESPONSE							
6A	POUT_OP_WARN_LIMIT							
6B	PIN_OP_WARN_LIMIT							
6C-77	Reserved							
78	STATUS_BYTE	Read Byte	Byte	Common	✓	✓	X	Read Only
79	STATUS_WORD	Read Word	Word	Common	✓	✓	X	Read Only
7A	STATUS_VOUT	Read Byte	Byte	PAGE	✓	✓	X	Read Only
7B	STATUS_IOUT	Read Byte	Byte	PAGE			X	Read Only
7C	STATUS_INPUT							
7D	STATUS_TEMPERATURE	Read Byte	Byte	PAGE	✓	✓	X	Read Only
7E	STATUS_CML	Read Byte	Byte	Common	✓	✓	X	Read Only
7F	STATUS_OTHER							
80	STATUS_MFR_SPECIFIC							See MFR_STATUS on page 79
81	STATUS_FANS_1_2	Read Byte	Byte	Common				Read Only
82	STATUS_FANS_3_4	Read Byte	Byte	Common				Read Only
83-87	Reserved							

Table 21-1. PMBus Commands (continued)

Code (hex)	Command	Transaction Type	Data Format [Units]	Scope	UCD91160	UCD91320	Data Flash	Comments
88	READ_VIN							
89	READ_IIN							
8A	READ_VCAP							
8B	READ_VOUT	Read Word	LINEAR16 [V]	PAGE	✓	✓	X	Read Only
8C	READ_IOUT	Read Word	LINEAR11 [A]	PAGE			X	Read Only
8D	READ_TEMPERATURE_1	Read Word	LINEAR11 [°C]	Common	✓	✓	X	Read Only
8E	READ_TEMPERATURE_2	R/W Word	LINEAR11 [°C]	PAGE			X	
8F	READ_TEMPERATURE_3							
90	READ_FAN_SPEED_1	Read Word	LINEAR11 [RPM]	Common				Read Only (Only valid when the fan is enabled.)
91	READ_FAN_SPEED_2	Read Word	LINEAR11 [RPM]	Common				Read Only (Only valid when the fan is enabled.)
92	READ_FAN_SPEED_3	Read Word	LINEAR11 [RPM]	Common				Read Only (Only valid when the fan is enabled.)
93	READ_FAN_SPEED_4	Read Word	LINEAR11 [RPM]	Common				Read Only (Only valid when the fan is enabled.)
94	READ_DUTY_CYCLE							
95	READ_FREQUENCY							
96	READ_POUT							
97	READ_PIN							
98	PMBUS_REVISION	Read Byte	Byte	Common	✓	✓		Read Only
99	MFR_ID	R/W Block (18 bytes)	String	Common	✓	✓		
9A	MFR_MODEL	R/W Block (12 bytes)	String	Common	✓	✓		
9B	MFR_REVISION	R/W Block (12 bytes)	String	Common	✓	✓		
9C	MFR_LOCATION	R/W Block (12 bytes)	String	Common	✓	✓		
9D	MFR_DATE	R/W Block (6 bytes)	String	Common	✓	✓		
9E	MFR_SERIAL	R/W Block (12 bytes)	String	Common	✓	✓		
9F	Reserved							
A0	MFR_VIN_MIN							
A1	MFR_VIN_MAX							
A2	MFR_IIN_MAX							
A3	MFR_PIN_MAX							
A4	MFR_VOUT_MIN							
A5	MFR_VOUT_MAX							
A6	MFR_IOUT_MAX							
A7	MFR_POUT_MAX							
A8	MFR_TAMBIENT_MAX							
A9	MFR_TAMBIENT_MIN							
AA-AC	Reserved							
AD	IC_DEVICE_ID	Block Read	String	Common	✓	✓		PMBus 1.2
AE	IC_DEVICE_REV	Block Read	String	Common	✓	✓		PMBus 1.2
AF	Reserved							
B0-B4	MFR_STATUS_0-MFR_STATUS_4	Read Byte	Byte	Common	✓	✓		Command Byte values are to be used with SMBALERT_MASK (0x1B) to write/read MFR_STATUS SMB Alert mask
B5	FIRST_BLACK_BOX_FAULT_INFO(USER_DATA_05)	Block Write/Read (16 Bytes)	Byte Array	Common	✓	✓		

Table 21-1. PMBus Commands (continued)

Code (hex)	Command	Transaction Type	Data Format [Units]	Scope	UC D91 160	UC D91 320	Dat a Flas h	Comments
B6	LAST_BLACK_BOX_FAULT_INFO(USER_DATA_06)	Block Read	Byte Array	Common	✓	✓		
B7	Reserved	Block Read	Byte Array	Common				
B8	RAIL_PROFILE(USER_DATA_08)	Block Read	Byte Array	PAGE	✓	✓		
B9	RAIL_STATE(USER_DATA_09)	R Block	Byte Array	PAGE	✓	✓		Provides the current, previous, and pending rail states.
BA-BF	USER_DATA_10 - USER_DATA_15							
C0-CF	Reserved							
D0	FAULT_PIN_CONFIG (MFR_SPECIFIC_00)	R/W Block	Byte Array	Common	✓	✓		Must be configured along with GPI_CONFIG which selects the external pin to use for the input source
D1	VOUT_CAL_MONITOR (MFR_SPECIFIC_01)	R/W Word	LINEAR16 [V]	PAGE	✓	✓	X	Offset calibration value for the sensor used in READ_VOUT. Signed.
D2	SYSTEM_RESET_CONFIG (MFR_SPECIFIC_02)	R/W Block (4, 6, or 9 bytes)	Byte Array	Common	✓	✓	X	Configures the System Reset function
D3	SYSTEM_WATCHDOG_CONFIG (MFR_SPECIFIC_03)	R/W Block (6 bytes)	Byte Array	Common	✓	✓		Configures the System Watchdog function
D4	SYSTEM_WATCHDOG_RESET (MFR_SPECIFIC_04)	Send Byte	n/a	Common	✓	✓		Resets the System Watchdog timeout counter
D5	MONITOR_CONFIG (MFR_SPECIFIC_05)	R/W Block	Byte Array	Common	✓	✓		Configure pins for monitoring (voltage, temperature, and so forth)
D6	NUM_PAGES (MFR_SPECIFIC_06)	Read Byte	Byte	Common	✓	✓		Read Only Returns the number of active pages
D7	RUN_TIME_CLOCK (MFR_SPECIFIC_07)	R/W Block	Byte Array	Common	✓	✓		Time provided in RTC format (Year, Month, Day, Hour, Minute, Second, Milisecond).
D8	RUN_TIME_CLOCK_TRIM (MFR_SPECIFIC_08)	R/W Word	Byte Array (LINEAR11)	Common	✓	✓		PPM adjustment for calibrating the run-time clock
D9	Reserved (MFR_SPECIFIC_09)		n/a					
DA	USER_RAM_00 (MFR_SPECIFIC_10)	R/W Byte	Byte	Common	✓	✓		RAM value that is set to 0 during device reset. By writing a nonzero value to this variable and then monitoring its value, a host can determine that a device reset has occurred.
DB	SOFT_RESET (MFR_SPECIFIC_11)	Send Byte	n/a	Common	✓	✓		Write Only This command restarts the controller firmware
DC	RESET_COUNT (MFR_SPECIFIC_12)	R/W Word (1)	2 Bytes (1)	Common				The number of times that the device has been reset.
DD	PIN_SELECTED_RAIL_STATES (MFR_SPECIFIC_13)	Read Block (18 bytes)	Byte Array	Common	✓	✓		Allows encoding on input pins to decide the state of each of the rails.
DE	RESEQUENCE (MFR_SPECIFIC_14)	Write Word	2 Bytes	Common	✓	✓	X	Commands selected rails to resequence. (Write Only)
DF	CONSTANTS (MFR_SPECIFIC_15)	Read Block (8 bytes)	Byte Array	Common	✓	✓		Fixed information about the device
E0	PWM_SELECT (MFR_SPECIFIC_16)	R/W Byte	Byte	Common	✓	✓	X	Determines which PWM the PWM_CONFIG command applies to
E1	PWM_CONFIG (MFR_SPECIFIC_17)	R/W Block (8 bytes)	Byte Array	Common	✓	✓		Configures a PWM (frequency, duty cycle, and phase)

Table 21-1. PMBus Commands (continued)

Code (hex)	Command	Transaction Type	Data Format [Units]	Scope	UCD91160	UCD91320	Data Flash	Comments
E2	PARM_INFO (MFR_SPECIFIC_18)	R/W Block (5 bytes)	Byte Array	Common	✓	✓	X	Parm Info: <parm base> <parm offset low byte> <parm offset high byte> <parm count> <parm size> This command sets the parameters used by the Parm Value command
E3	PARM_VALUE (MFR_SPECIFIC_19)	R/W Block	Byte Array	Common	✓	✓	X	Value transferred to memory location chosen by the PARM_INFO command
E4	TEMPERATURE_CAL_GAIN (MFR_SPECIFIC_20)	R/W Word	LINEAR11 [°C/V]	PAGE	✓	✓		Gain calibration for the external sensors used by the READ_TEMPERATURE_2 command
E5	TEMPERATURE_CAL_OFFSET (MFR_SPECIFIC_21)	R/W Word	LINEAR11[°C]	PAGE	✓	✓		Offset calibration for the external sensors used by the READ_TEMPERATURE_2 command
E6	SET_BREAKPOINTS (MFR_SPECIFIC_22)	R/W Word	2 Bytes	Page	✓	✓		Used to set breakpoints for given states
E7	DEBUG_CONTINUE (MFR_SPECIFIC_23)	R/W Byte (4 Bytes)	Byte Array	Common	✓	✓		Used to indicate a page should exit the breakpoint state
E8	(MFR_SPECIFIC_24)							
E9	FAULT_RESPONSES (MFR_SPECIFIC_25)	R/W Block (9 Bytes)	Byte Array	PAGE	✓	✓		Defines the response to all supported faults
EA	LOGGED_FAULTS (MFR_SPECIFIC_26)	R/W Block	Byte Array	Common	✓	✓		Flags in Data Flash that are set when each fault type occurs on each page (2)
EB	LOGGED_FAULT_DETAIL_INDEX (MFR_SPECIFIC_27)	R/W Word	2 bytes	Common	✓	✓	X	Number of LOGGED_FAULT_DETAIL entries and a read/write index into those entries
EC	LOGGED_FAULT_DETAIL (MFR_SPECIFIC_28)	Read Block	Byte Array	Common	✓	✓		Detail information about the faults that have occurred
ED	LOGGED_PAGE_PEAKS (MFR_SPECIFIC_29)	R/W Block	Byte Array, Byte [°C], LINEAR16 [V], LINEAR11 [amp]	PAGE				Peak temperature, voltage, and current for a given page, stored in Data Flash (2)
EE	LOGGED_COMMON_PEAKS (MFR_SPECIFIC_30)	R/W Byte	Byte [°C]	Common				Peak internal temperature (2)
EF	LOGGED_FAULT_DETAIL_ENABLES (MFR_SPECIFIC_31)	R/W Block	Byte Array	Common	✓	✓		Selectable fault logging by rail and by fault type
F0	EXECUTE_FLASH (MFR_SPECIFIC_32)	Send Byte	n/a	Common				If in ROM mode, starts the device executing in FLASH mode. If already in FLASH mode, command has no effect.
F1	SECURITY (MFR_SPECIFIC_33)	R/W Block (6 bytes)	Binary Array	Common				Sets the password used to secure the unit against unauthorized modification of its settings
F2	SECURITY_BIT_MASK (MFR_SPECIFIC_34)	R/W Block (32 bytes)	Binary Array	Common				Configures which commands are password protected.
F3	MFR_STATUS (MFR_SPECIFIC_35)	Read Block (2 or 4 bytes)	Byte Array	PAGE	✓	✓	X	Replaces the STATUS_MFR_SPECIFIC command
F4	GPI_FAULT_RESPONSES (MFR_SPECIFIC_36)	R/W Block	Byte Array	PAGE	✓	✓		GPI Fault Responses per page

Table 21-1. PMBus Commands (continued)

Code (hex)	Command	Transaction Type	Data Format [Units]	Scope	UCD91160	UCD91320	Data Flash	Comments
F5	MARGIN_CONFIG (MFR_SPECIFIC_37)	R/W Byte	Byte	PAGE	✓	✓		Selects the margining pin and other margining configuration
F6	SEQ_CONFIG (MFR_SPECIFIC_38)	R/W Block (6-12 bytes)	Byte Array	PAGE	✓	✓		Configures sequencing dependencies and enable pin
F7	GPO_CONFIG_INDEX (MFR_SPECIFIC_39)	R/W Byte	Byte	Common	✓	✓	X	Selects to which GPO the GPO_CONFIG command applies
F8	GPO_CONFIG (MFR_SPECIFIC_40)	R/W Block (20 or 29 bytes)	Byte Array	Common	✓	✓		Configures output pins and their dependencies
F9	GPI_CONFIG (MFR_SPECIFIC_41)	R/W Block (13 bytes)	Byte Array	Common	✓	✓		Configures input pins
FA	GPIO_SELECT (MFR_SPECIFIC_42)	R/W Byte	Byte	Common	✓	✓	X	Determines to which GPIO the GPIO_CONFIG command applies
FB	GPIO_CONFIG (MFR_SPECIFIC_43)	R/W Byte	Byte	Common	✓	✓		Set or get the state of a GPIO
FC	MISC_CONFIG (MFR_SPECIFIC_44)	R/W Block	2 Bytes	Common	✓	✓		Miscellaneous configuration settings
FD	DEVICE_ID (MFR_SPECIFIC_45)	Read Block (up to 32 bytes)	String	Common	✓	✓		Returns ASCII string with hardware and firmware version information of the controller
FE	Mfr_Specific_Extended_Command				✓	✓		
FF	PMBUS_Extended_Command				✓	✓		

(1) These values are only applied when associated voltage is in regulation. If there is not an associated voltage monitor, these values are applied after the rail is enabled and after TON_DELAY.

(2) There is a chance that a write to this command receives a NACK. When the firmware starts a periodic or commanded update of data flash, it can take up to 100 ms to complete. During that time, writes to these commands receive a NACK. If this occurs, wait 100 ms and retry the command. This note only applies if the brownout feature is not enabled (see brownout enable mode in MISC_CONFIG).

22 Implementation Details for PMBus Core Commands

The following PMBus core commands are defined in the PMBus specification. This section describes details that are unique to the UCD91xxx implementation.

22.1 (00h) PAGE

The PAGE command provides the ability to configure, control, and monitor multiple outputs on one unit using a single PMBus physical address. When the PAGE command is sent, all subsequent commands are applied to settings for the rail selected by the PAGE command.

The GUI uses the term Rail to refer to a voltage output. Rails are numbered starting with one, whereas pages are numbered starting at zero. The relationship between the PMBus PAGE value and the Rail number is shown in Table 22-1.

Setting PAGE = 0xFF means that the following write commands are to be applied to all outputs. A page setting of 0xFF is invalid for all read commands, with the exception of the PAGE command.

Table 22-1. Relationship Between PAGE and Rail (For devices with 32 rails)

Page	Output Rail
0	1
1	2
2	3
...	...
31	32
32 – 254	Invalid
255 (0xFF)	All

Section 11.10 of the PMBus specification describes the PAGE command in more detail.

22.2 (01h) OPERATION

This command is used to turn outputs on and off along with input from the CONTROL pin. Section 12.1 of the PMBus specification describes this command in more detail.

The UCD91xxx supports the following modes for the Operation command:

- Immediate Off (No Sequencing)
- Soft Off (With Sequencing)
- On Nominal (No Margining)
- On Margin High (Ignore Faults)
- On Margin High (Act on Fault)
- On Margin Low (Ignore Faults)
- On Margin Low (Act on Fault)

22.3 (0Eh) PASSKEY

The read/write block PASSKEY command works in combination with the ACCESS_CONTROL command described in Section 22.4 to restrict access to updating the ACCESS_CONTROL values by setting the internal lock and unlock state of the device. The passkey is an 8-byte value, that cannot be read out from the device once written.

Table 22-2. PASSKEY Command Format

Byte Number (Write)	Bye Number (Read)	Payload Number	Write Description	Read Description
0			COMMAND BYTE = 0x0E	
1	0		BYTE COUNT = 08h	

Table 22-2. PASSKEY Command Format (continued)

Byte Number (Write)	Bye Number (Read)	Payload Number	Write Description	Read Description
2	1	0	Passkey Byte (LSB)	Passkey Status Byte
3	2	1	Passkey Byte	Reserved
...	...	...	...	...
9	8	7	Passkey Byte (MSB)	Reserved

Writing a non-zero passkey value to the UCD91xxx device when no passkey is set will set the passkey to the written value. When a paskey is set but the device is not locked, writing all 0x00 values to the PASSKEY command will unset the passkey, and allow for setting a new passkey if desired. Writing a set passkey value will lock the device, preventing writes to the ACCESS_CONTROL command from taking effect.

If the PASSKEY is locked, the UCD91xxx allows up to 7 attempts to unlock the device per POR. Failure to unlock the device in those 7 attempts will lock the ACCESS_CONTROL command until a device reset is perfomed. Writing a correct PASSKEY when the number of allowed attempts has been exceeded will not unlock the device. If a set PASSKEY value is stored by a STORE_DEFAULTS_ALL command, upon reset the device will start in a LOCKED state, and no access control values will be updateable until the PASSKEY is unlocked. The user may read the PASSKEY command to determine the current lock/unlock state of the device, and if any unlock attempts remain. The formatting for this information is as shown in Table 22-3.

Table 22-3. Passkey Status Byte Values

Passkey Read Byte Value	Meaning
0x00	Passkey state is unlocked (May be set or unset).
0x1X	Passkey state is locked. X indicates the number of failed unlock attempts since last device reset.
0x1F	Passkey state is locked. The maximum number of unlock attempts has been exceeded.

22.4 (0Fh) ACCESS_CONTROL

The ACCESS_CONTROL command uses the write word command format for writes, and the process call format for reads, as described in the PMBus Standard. This command allows the system integrator to limit read and write access to supported commands at will, as restricted by passkey lock/unlock state (see Section 22.3), or permanently.

Table 22-4 shows the interpretation of each written byte of the ACCESS_CONTROL command. Table 22-5 shows the values and fields available in the ACCESS_CONTROL options byte, and which of those values are supported in the UCD91xxx family of devices and storable. All non-security related UCD91xxx commands are supported and have their own access control bytes and independent settings. The following commands are unsupported:

- (0Eh) PASSKEY
- (0Fh) ACCESS_CONTROL

Table 22-4. ACCESS_CONTROL Write Format

Byte Number (Write)	Payload Number	Description
0		COMMAND_BYTE = 0Fh
1	0	ACCESS_CONTROL Target Command Code
2	1	Access Control Options Byte

Table 22-5. Access Control Byte Interpretation

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	WRITE_ACCESS	READ_ACCESS	AUTHENTICATED_WRITE	NVM_STORE	NVM_RESTORE	WRITE_ONCE	NO_MORE	NEVER_AGAIN
UCD91xx Supported	Yes	Yes	No	No	No	Yes	Yes	Yes
Storable	Yes	Yes	No	No	No	Yes	No	Yes

The behavior of the access control options for the UCD91xx family of devices is as follows:

- WRITE_ACCESS: When this bit is set, writes to the target command code are restricted. Attempts to write these commands will set an INVALID data error in the STATUS_CML register.
- READ_ACCESS: When this bit is set, reads of the target command code are restricted, and attempts to read these commands will set an INVALID data error in the STATUS_CML register.
- WRITE_ONCE: When this bit is set, the target command code will allow only a single write, and will restrict WRITE_ACCESS from that point onward. A POR is required to reset the number of allowed writes. Note that attempting to write a READ_ONLY command will not update the ACCESS_CONTROL byte settings after the write attempt, since writes to these commands are already restricted.
- NO_MORE: When this bit is set, the ACCESS_CONTROL byte settings for the target command code will not allow any more writes until next POR. Even if PASSKEY is unlocked. This bit is not storable.
- NEVER_AGAIN: When this bit is set, it behaves the same as the NO_MORE bit. When it is stored using a STORE_DEFAULT_ALL command, and a POR is performed, the ACCESS_CONTROL byte for the given target command code is locked forever. If it has not been stored, a POR can clear this bit status.

Note

When the WRITE_ONCE feature is enabled, and a single write as been performed for a given command, the WRITE_ACCESS bit will be considered set. Executing a STORE_DEFAULT_ALL command at this point will store the WRITE_ACCESS value and on subsequent PORs writes to the command will be disabled. Users must enusre that the ACCESS_CONTROL value they want to store is correct before executing a STORE_DEFAULT_ALL command.

Table 22-6 shows the format for reading the ACCESS_CONTROL command. The returned value of the ACCESS_CONTROL byte depends on the value of the read options byte as described in Table 22-7.

Table 22-6. ACCESS_CONTROL Command Read Format

Byte Number	**Write/Read**	**Description**
0	Write	COMMAND_BYTE = 0Fh
1	Write	Target Command Code
2	Write	Read Option
Repeated Start + Addr Byte		
3	Read	Target Command Code
4	Read	Access Control Byte

Table 22-7. ACCESS_CONTROL Read Options Byte

Written Read Option	**Access Control Byte Value**
0x00	Current Access Control byte setting
0x02	Supported Access Control byte setting
0x80	Storable Access Control byte settings

22.5 (11h) STORE_DEFAULT_ALL

The STORE_DEFAULT_ALL command saves the PMBus parameters from Operating Memory into the Default Store in Data Flash. The UCD91xxx uses the most recently written set of Default Store values at startup or after a RESTORE_DEFAULT_ALL command. If the Default Store has never been written, values from the hard-coded memory are used.

Note

The device configuration can be corrupted if a power cycle or a reset occurs before this save operation is completed. To ensure a successful save operation, after writing the STORE_DEFAULT_ALL command, wait for the STORE_DEFAULT_ALL_DONE bit to be set (see MFR_STATUS). Once that bit is set, check the STORE_DEFAULT_ALL_ERROR bit. If the STORE_DEFAULT_ALL_DONE bit is set and the STORE_DEFAULT_ALL_ERROR is not set, the operation was successful.

Note

Monitoring and other tasks (including PMBus communication) are not performed during this save operation which can take up to 100 milliseconds.

22.6 (12h) RESTORE_DEFAULT_ALL

The RESTORE_DEFAULT_ALL command restores the PMBus parameters from the Default Store into Operating Memory. If the Default Store has never been written, values from the hard-coded memory are used.

This command is not supported on UCD91xxx.

Note

Resetting the device is a better (more thorough) way to perform this operation.

22.7 (1Bh) SMBALERT_MASK

This command is added in PMBUS specification 1.2 and is described in section 15.38. It is used to block a status bits from causing the SMBALERT# signal to be asserted. Refer to PMBUS specification document for format details. This command has to be used together with status command codes to write or read the mask of a specific status bit. The status command codes that can be used with this command include:

- STATUS_VOUT
- STATUS_IOUT
- STATUS_TEMPERATURE
- STATUS_CML

Another status command code, MFR_STATUS, also needs to work with the SMBALERT_MASK command. However, SMBALERT_MASK can only write/read one byte mask at a time. Since MFR_STATUS contains 48 bits, it is split into 6 sub-commands to work with the SMBALERT_MASK command:

- MFR_STATUS_0: for MFR_STATUS bit 7 to 0
- MFR_STATUS_1: for MFR_STATUS bit 15 to 8
- MFR_STATUS_2: for MFR_STATUS bit 23 to 16
- MFR_STATUS_3: for MFR_STATUS bit 31 to 24
- MFR_STATUS_4: for MFR_STATUS bit 39 to 32
- MFR_STATUS_5: for MFR_STATUS bit 47 to 40

22.8 (20h) VOUT_MODE

This command, described in Sections 8.1 and 8.2 of the PMBus specification, indicates the data format used for all commands related to output voltage (all LINEAR16 commands). The command includes a 3-bit Mode field and a 5-bit Parameter field. In UCD91xxx, the Mode field is a read-only and is fixed to 000b (linear data format, described in Section 18.1). The Parameter field is the exponent value and can be modified.

Note

This exponent field must be updated before modifying any value that is affected by this setting (all LINEAR16 commands).

Table 22-8 shows the full-scale range and the resolution of different exponent values. For example, to monitor a 12-V signal with the finest possible resolution, the preferred exponent is –11.

Table 22-8. Exponent Influence on Voltage Range and Resolution

Exponent	Range (V)		Resolution (mV)
–7	0	255.99219	7.81250
–8	0	127.99609	3.90625
–9	0	63.99805	1.95313
–10	0	31.99902	0.97656
–11	0	15.99951	0.48828
–12	0	7.99976	0.24414
–13	0	3.99988	0.12207
–14	0	1.99994	0.06104
–15	0	0.99997	0.03052

The ADC of UCD91xxx devices supports both internal and external voltage reference. The monitored voltage signals must be scaled proportionally to fit into the ADC window. The nominal voltage value after scaling must be less than 2.5 V, so as to take into account overvoltage scenarios. Table 22-9 shows nominal voltage ranges suited for each exponent value and corresponding scale factors. Table 22-9 assumes the nominal voltages' setpoint is at 1.875 V after scaling, 75% of the 2.5-V ADC full-scale range. The scale factor is programmed by VOUT_SCALE_MONITOR command.

Table 22-9. Example Settings for Voltage Monitoring Ranges

Exponent	High		Low	
	Voltage	Scale	Voltage	Scale
–7	192	0.009766	96	0.019531
–8	96	0.019531	48	0.039063
–9	48	0.039063	24	0.078125
–10	24	0.078125	12	0.15625
–11	12	0.15625	6	0.3125
–12	6	0.3125	3	0.625
–13	3	0.625	1.5	1.25
–14	1.5	1.25	0.75	2.5
–15	0.75	2.5	0.375	5

22.9 (38h) IOUT_CAL_GAIN

This command, described in section 14.8 of the PMBus specification, is used to configure the gain of the current-sense circuit. The units for this command are milliohms (mV/A).

This command and current sensing in general are not currently supported in the UCD91xxx family of devices.

22.10 (41h – 69h) xxx_FAULT_RESPONSE

The following commands are replaced by the FAULT_RESPONSES command (see Section 26.25).

Table 22-10. Replaced Fault Response Commands

Code	Command
41h	VOUT_OV_FAULT_RESPONSE
45h	VOUT_UV_FAULT_RESPONSE
47h	IOUT_OC_FAULT_RESPONSE
4Ch	IOUT_UC_FAULT_RESPONSE
50h	OT_FAULT_RESPONSE
63h	TON_MAX_FAULT_RESPONSE

22.11 (62h) TON_MAX_FAULT_LIMIT

This command, described in Section 16.3 of the PMBus specification, states that the "command sets an upper limit, in ms, on how long the unit can attempt to power up the output without reaching the *output undervoltage fault limit*". For the UCD91xxx, this command is instead "set an upper limit, in milliseconds, on how long the unit can attempt to power up the output without reaching the *POWER_GOOD_ON voltage level*."

When no voltage monitor pin is assigned to a rail, the sequencer cannot monitor the rail voltage and determine its power-good state. Therefore, after the rail is enabled, it is given power-good state after a delay time defined by TON_MAX_FAULT_LIMIT.

22.12 (66h) TOFF_MAX_WARN_LIMIT

When there is no voltage monitor pin assigned to a rail, this time is used to determine when a rail leaves the power-good state after it is disabled. In this case, setting TOFF_MAX_WARN_LIMIT to 0x7FFF (No Limit) results in no delay between disabling the rail and leaving the power-good state.

22.13 (80h) STATUS_MFR_SPECIFIC

The UCD91xxx has replaced the STATUS_MFR_SPECIFIC command with MFR_STATUS (see Section 26.33) to support more than eight status bits.

22.14 (8Dh) READ_TEMPERATURE_1

This read-only command returns the temperature from a sensor embedded inside the UCD91xxx controller.

22.15 (8Eh) READ_TEMPERATURE_2

This paged command returns the temperature from an external temperature sensor located in or near an output power module.

Note

If no temperature monitor pin is assigned to a rail, the internal temperature sensor is used for that rail. See the MONITOR_CONFIG command (Section 26.7).

If the user writes this command, the written value is used as the temperature until another value is written by the user, regardless whether or not a temperature monitor pin is assigned to the rail. When a temperature monitor pin is assigned, once READ_TEMPERATURE_2 is written, the temperature monitor pin is not used again until the device is reset. Similarly, when no temperature monitor pin is assigned, once READ_TEMPERATURE_2 is written, the internal temperature sensor is not used again until the device is reset.

22.16 (ADh) IC_DEVICE_ID

This command is added in PMBUS1.2. It is used to read the part number of the device.

22.17 (AEh) IC_DEVICE_REV

This command is added in PMBUS1.2. It is used to read the revision of the device.

23 Input and Output Pin Configuration

Each of the input pins (GPI, and so forth) and output pins (Enable, PWM, GPO, and so forth) are configured using two bytes. Thee bits are defined as follows:

Bit 10 sets the pin polarity. A pin is asserted when its state is active level.

- 0: Active low
- 1: Active high

Bits 9:8 set the mode for the pin.

- 0: Unused
- 1: Input
- 2: Active-driven output
- 3: Open-drain output

Bits 7:0 select the Pin ID of the desired I/O pin. The pin IDs are numbers of 1 - 88, and correspond to the primary function of a given pin. The Pin ID of 0 is unused, and typically indicates a pin/option is not enabled.

These configuration bytes are used in several of the following commands, such as SEQ_CONFIG, GPO_CONFIG, and GPI_CONFIG, and FAULT_PIN_CONFIG. The commands often dictate the Mode of the pin. For example, an Enable can only be an output. It cannot be configured as an input. But, for consistency, this configuration byte format previously described is always used.

Note

Pin Usage Conflicts

It is possible to issue commands with conflicting pin selections. UCD9xxxx firmware does not attempt to detect and prevent all possible invalid setting combinations. The GUI provides some additional validity checking, but it is the user's responsibility to eliminate conflicting GPIO configurations.

Example: If a command is used to configure a pin for a specific GPIO or sequencing purpose and then issued again with the same pin unassigned, the pin can not revert back to its default usage until after the controller has been reset or power cycled.

Table 23-1. UCD91xxx Pin ID Definitions

Pin ID	UCD 91x Pin Function	Pin Available on Device	
		UCD91320	UCD91160
0	N/A	No	No
1	AMON1	Yes	Yes
2	AMON2	Yes	Yes
3	AMON3	Yes	Yes
4	AMON4	Yes	Yes
5	AMON5	Yes	Yes
6	AMON6	Yes	Yes
7	AMON7	Yes	Yes
8	AMON8	Yes	Yes
9	AMON9	Yes	Yes
10	AMON10	Yes	Yes
11	AMON11	Yes	Yes
12	AMON12	Yes	Yes
13	AMON13	Yes	Yes
14	AMON14	Yes	Yes
15	AMON15	Yes	Yes
16	AMON16	Yes	Yes

Table 23-1. UCD91xxx Pin ID Definitions (continued)

Pin ID	UCD 91x Pin Function	Pin Available on Device	
		UCD91320	UCD91160
17	AMON17	Yes	No
18	AMON18	Yes	No
19	AMON19	Yes	No
20	AMON20	Yes	No
21	AMON21	Yes	No
22	AMON22	Yes	No
23	AMON23	Yes	No
24	AMON24	Yes	No
25	DMON1	Yes	No
26	DMON2	Yes	No
27	DMON3	Yes	No
28	DMON4	Yes	No
29	DMON5	Yes	No
30	DMON6	Yes	No
31	DMON7	Yes	No
32	DMON8	Yes	No
33	EN1	Yes	Yes
34	EN2	Yes	Yes
35	EN3	Yes	Yes
36	EN4	Yes	Yes
37	EN5	Yes	Yes
38	EN6	Yes	Yes
39	EN7	Yes	Yes
40	EN8	Yes	Yes
41	EN9	Yes	Yes
42	EN10	Yes	Yes
43	EN11	Yes	Yes
44	EN12	Yes	Yes
45	EN13	Yes	Yes
46	EN14	Yes	Yes
47	EN15	Yes	Yes
48	EN16	Yes	Yes
49	EN17	Yes	No
50	EN18	Yes	No
51	EN19	Yes	No
52	EN20	Yes	No
53	EN21	Yes	No
54	EN22	Yes	No
55	EN23	Yes	No
56	EN24	Yes	No
57	EN25	Yes	No
58	EN26	Yes	No
59	EN27	Yes	No
60	EN28	Yes	No
61	EN29	Yes	No
62	EN30	Yes	No

Table 23-1. UCD91xxx Pin ID Definitions (continued)

Pin ID	UCD 91x Pin Function	Pin Available on Device	
		UCD91320	UCD91160
63	EN31	Yes	No
64	EN32	Yes	No
65	MAR1	Yes	Yes
66	MAR2	Yes	Yes
67	MAR3	Yes	Yes
68	MAR4	Yes	Yes
69	MAR5	Yes	Yes
70	MAR6	Yes	Yes
71	MAR7	Yes	Yes
72	MAR8	Yes	Yes
73	MAR9	Yes	No
74	MAR10	Yes	No
75	MAR11	Yes	No
76	MAR12	Yes	No
77	MAR13	Yes	No
78	MAR14	Yes	No
79	MAR15	Yes	No
80	MAR16	Yes	No
81	GPIO1	Yes	Yes
82	GPIO2	Yes	Yes
83	GPIO3	No	Yes
84	GPIO4	No	Yes
85	GPIO5	No	Yes
86	GPIO6	No	Yes
87	GPIO7	No	Yes
88	GPIO8	No	Yes

24 PWM Configuration

The MARx pins identified in the device datasheet can be configured as PWM outputs. These can be used for margining or active trimming of a rail. Table 24-1 shows the range of output frequencies supported on the various MARx pins, as well as the source frequency for PWM Generation. Check the device datasheet for the number of available MARx pins per UCD91x device.

Table 24-1. PWM Supported Frequencies

MARx Pin	FCLK Frequency	Output Frequency Range
MARx (1-8)	80MHz	1 kHz to 1 MHz
MARx (9-16)	40MHz	1 kHz to 1 MHz

The PWM_CONFIG command also allows for the setting of frequency for a given PWM as chosen by the PWM_SELECT command. However, the frequencies are not fully independent of each other, and within the groups of MARx pins shown in Table 24-2, the frequencies are the same. This means that setting a frequency for 1 member of a group to 0 will disable the other PWM outputs within that group as well.

Table 24-2. PWM Usage Notes

MARx Group #	MARx Pins (Shared Frequency settings)
1	MAR1, MAR2, MAR3, MAR4
2	MAR5, MAR6
3	MAR7, MAR8
4	MAR9, MAR10
6	MAR11, MAR12
7	MAR13, MAR14, MAR15, MAR16

25 Implementation Details for User Data Commands

25.1 (B5h) FIRST_BLACK_BOX_FAULT_INFO (USER_DATA_05)

The block read-only command returns detailed information of the first fault.

1. Time stamp in RTC Format (see *RUN_TIME_CLOCK Command Format*)
2. Whether or not the fault is page specific
3. The page or GPI that triggered the fault(when applicable)
4. Fault type
5. Status of all rail power good for rails

Clearing the Log: Writing all 0's to this command clears BLACK BOX log entries (first and last). Clearing the black box log will allow to re-log the same fault in both the normal and black box logs.

Table 25-1. FIRST_BLACK_BOX_FAULT_INFO Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description	
0			CMD = B5	
1	0		BYTE_COUNT = 16	
2	1	0	**Bit**	**Description**
			0	paged or non-paged fault
			1	Fault log valid (0)/Not valid Log(1)
			2	Reserved
			7:3	fault_page_num
3	2	1	Seconds & Miliseconds (Low Byte)	
4	3	2	Seconds & Miliseconds (High Byte)	
5	4	3	Days, Hours, & Minutes (Low Byte)	
6	5	4	Days, Hours, & Minutes (High Byte)	
7	6	5	Year & Month (Low byte)	
8	7	6	Year & Month (High byte)	
9	8	7	Fault value (low byte) see Table 26-32	
10	9	8	Fault value (middle low byte) see Table 26-32	
11	10	9	Fault value (middle high byte) see Table 26-32	
12	11	10	Fault value (high byte) see Table 26-32	
13	12	11	**Bit**	**Description**
			0:4	Fault Type (see Table 26-32
			5	Fault Log Full (0) / Not Full (1)
			6:7	Reserved
14	13	12	Rail Power Good Mask (Low Byte)	
15	14	13	Rail Power Good Mask	
16	15	14	Rail Power Good Mask	
17	16	15	Rail Power Good Mask (High Byte)	

25.2 (B6h) LAST_BLACK_BOX_FAULT_INFO Command Format(USER_DATA_06)

The block read-only command returns the black box fault log for a fault written off at device power down. The format is the same as shown in Table 25-1.

25.3 (B8h) RAIL_PROFILE (USER_DATA_08)

This R/W Block paged command selects the index of the rail profile that will be used for later VOUT_COMMAND, VOUT_OV_FAULT_LIMIT, VOUT_OV_WARNING_LIMIT, VOUT_MARGIN_HIGH, POWER_GOOD_ON, VOUT_MARGIN_LOW, POWER_GOOD_OFF, VOUT_UV_WARNING_LIMIT and VOUT_UV_FAULT_LIMIT commands. GPIs must assigned to switch the profile based on it asserted/deasserted state.

Table 25-2. RAIL_PROFILE Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = B8
1	0		BYTE_COUNT = 2
2	1	0	Number Profile
3	2	1	Profile Index
			Bit Description: [7:4] • These bits are read-only. They return the current profile index used by the device. [3:0] • Profile index

25.3.1 Number Profile

In the RAIL_PROFILE write command, number profile is used to perform various operations when combined with profile index.

Table 25-3. Number Profile

Number Profile	Descriptions	Profile Index
0x00	Access Profile	Profile index to access
0x01 - 0x04	Total number of profiles for a given rail	Profile Index to access. It should be less than the total number of profiles.
0x05 - 0xFF	Invalid command	Invalid command

When Number profile is set to 0, the profile index parameter is used to set the right profile index for host to access. Device will set Invalid Data for this command when the profile index is bigger or equal than total available profile for the given rail.

When number profile is set to 1-4, device shall add or delete profiles properly based on the existing number profile and receiving number profile. If the new number profile is bigger than the existing number profile, device shall add the extra new profiles; otherwise device shall delete the extra profile. If the profile to be deleted is the same one under used, device will set Invalid Data for the command. The profile index associated with this command indicates the profile index to be accessed from host. If the profile index associated with the number profile is bigger than or equal the number profile, device will set Invalid Data for this command.

Device will set Invalid Data for this command if number profile is bigger than 4.

When importing a new project, Application must call this command with correct profile number for the given rail. Otherwise device will use the profile information from the existing rail to process.

In the RAIL_PROFILE read command, number profile returns total number of profiles for the given rail.

25.3.2 Profile Index

Each rail (except GPI rail and rail without voltage monitoring) shall have at least one profile and may have up to 4 profiles with total 50 sharing profile pool. Once all 50 profiles have been allocated, no new profile shall be added and Device will set Invalid Data. The profile index shall range from 0 to 3.

25.4 (B9h) RAIL_STATE (USER_DATA_09)

The Read Block Paged Command returns the current, previous, and pending state of the pages.

Table 25-4. RAIL_STATE Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = B9
1	0		BYTE_COUNT = 1
2	1	0	Current Rail State(see Table 25-5)
3	2	1	Previous Rail State
4	3	2	Pending Rail State

The previous rail state is the state which preceeded the current rail state. The pending rail state is the page that the rail will proceed to in the event a DEBUG_CONTINUE command for this page is issued. It is only different from the current rail state when the rail is in the BREAKPOINT state.

Table 25-5. Rail State

Rail State	Value	Description
IDLE	1	On condition is not met, or rail is shut down due to fault, or rail is waiting for the resequence
SEQ_ON	2	Wait the dependency to be met to assert ENABLE signal
START_DELAY	3	TON_DELAY to assert ENABLE signal
RAMP_UP	4	Enable is asserted and rail is on the way to reach power good
REGULATION	5	Once the monitoring voltage is over POWER_GOOD when enable signal is asserted, rails stay at this state even if the voltage is below POWER_GOOD late as long as there is no fault action taken.
SEQ_OFF	6	Wait the dependency to be met to deassert ENABLE signal
STOP_DELAY	7	TOFF_DELAY to deassert ENABLE signal
RAMP_DOWN	8	Enable signal is deasserted and rail is ramping down. This state is only available if TOFF_MAX_WARN_LIMIT is not set to unlimited; or If the turn off is triggered by a fault action, rail must not be under fault retry to show RAMP DOWN state. Otherwise, IDLE state is present.
BREAKPOINT	9	The rail has hit a breakpoint as described in the SET_BREAKPOINTS command. It will remain here until a DEBUG_CONTINUE command is received, at which point it will proceed to the pending state.

26 Implementation Details for Manufacturer-Specific Commands

26.1 Manufacturer-Specific Commands Notice

The tables in the following subsections are based off of the UCD91xxx superset device, which assumes the following:

- The device supports up to 32 rails.
- The device supports up to 32 GPIs.
- The device supports up to 16 LGPOs.

For a given device, this may not be the case. For devices supporting less than 32 GPIs or 32 rails, the above information should be interpreted in the following ways:

- During write operations, bytes encoding extraneous information (i.e. Bytes representing rails 16-31 for 16 rail devices) are considered as don't cares.
- During read operations, the value of 0 is returned for these fixed bytes.

Refer to the device-specific datasheet for more details on exact features supported.

26.2 (D0h) FAULT_PIN_CONFIG (MFR_SPECIFIC_00)

This Read/Write block command configures Fault Pin function for synchronized cascading operation. Fault Pins of different UCD91xxx devices can be connected to the same Fault Bus. The Fault Bus is pulled up to 3.3 V through a resistor. When one Fault Pin pulls the bus low, other Fault Pins on the bus detect the same fault event. As a result, several cascaded UCD91xxx devices respond to the fault event synchronously. The following events can impact the fault pins: RESEQUENCE_ERROR, SEQ_ON_TIMEOUT, SEQ_OFF_TIMEOUT, OT_FAULT, IOUT_UC_FAULT, IOUT_OC_FAULT, VOUT_UV_FAULT, VOUT_OV_FAULT, and TON_MAX_FAULT.

For the SEQ_ON_TIMEOUT, SEQ_OFF_TIMEOUT, and TON_MAX_FAULT conditions, the fault pin will assert until the rail is commanded off and on again. For the other fault types, the fault pin will assert until the condition is otherwise cleared.

Fault pin configuration includes three commands: this command (D0h), (F9h) GPI_CONFIG (MFR_SPECIFIC_41) and (F4h) GPI_FAULT_RESPONSES (MFR_SPECIFIC_36). This command (D0h) configures Fault Pins as fault-influenced outputs. The state of a Fault Pin is determined by any fault of the selected rails. The (F9h) GPI_CONFIG (MFR_SPECIFIC_41) command configures Fault Pins as fault inputs, in which the Fault Enable Flags are set (see Section 26.39.2). The (F4h) GPI_FAULT_RESPONSES (MFR_SPECIFIC_36) command configures fault response. This way, when there is a fault coming from the fault bus, the fault pin behaves as a GPI and triggers GPI fault response; where there is a fault raised from within the device, the fault pin pulls low the fault bus notifying other fault pins on the same bus. When using TI Sequencer Studio to configure the device, the GUI automatically issues the previous three commands to configure the fault pins. When setting up the fault pins for bidirectional faults, the pin configuration for the FAULT_PIN_CONFIG and the GPI_CONFIG should be set to match.

Table 26-1. FAULT_PIN_CONFIG Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = D0
1	0		BYTE_COUNT = 41
2	1	0	Page Mask for Fault Pin 0 (Byte 0 – LSB)
3	2	1	Page Mask for Fault Pin 0 (Byte 1)
4	3	2	Page Mask for Fault Pin 0 (Byte 2)
5	4	3	Page Mask for Fault Pin 0 (Byte 3 – MSB)
6	5	4	Page Mask for Fault Pin 1 (Byte 0 – LSB)
7	6	5	Page Mask for Fault Pin 1 (Byte 1)
8	7	6	Page Mask for Fault Pin 1 (Byte 2)
9	8	7	Page Mask for Fault Pin 1 (Byte 3 – MSB)
10	9	8	Page Mask for Fault Pin 2 (Byte 0 – LSB)
11	10	9	Page Mask for Fault Pin 2 (Byte 1)

Table 26-1. FAULT_PIN_CONFIG Command Format (continued)

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
12	11	10	Page Mask for Fault Pin 2 (Byte 2)
13	12	11	Page Mask for Fault Pin 2 (Byte 3 – MSB)
14	13	12	Page Mask for Fault Pin 3 (Byte 0 – LSB)
15	14	13	Page Mask for Fault Pin 3 (Byte 1)
16	15	14	Page Mask for Fault Pin 3 (Byte 2)
17	16	15	Page Mask for Fault Pin 3 (Byte 3 – MSB)
18	17	16	Pin Config for Fault Pin 1 (Pin ID)
19	18	17	Pin Config for Fault Pin 1 (Other)
20	19	18	Pin Config for Fault Pin 2
21	20	19	Pin Config for Fault Pin 2
22	21	20	Pin Config for Fault Pin 3
23	22	21	Pin Config for Fault Pin 3
24	23	22	Pin Config for Fault Pin 4
25	24	23	Pin Config for Fault Pin 4
26	25	24	GPI Mask for Fault Pin 1 (Byte 0 - LSB)
27	26	25	GPI Mask for Fault Pin 1
28	27	26	GPI Mask for Fault Pin 1
29	28	27	GPI Mask for Fault Pin 1 (Byte 3 - MSB)
30	29	28	GPI Mask for Fault Pin 2 (Byte 0 - LSB)
31	30	29	GPI Mask for Fault Pin 2
32	31	30	GPI Mask for Fault Pin 2
33	32	31	GPI Mask for Fault Pin 2 (Byte 3 - MSB)
34	33	32	GPI Mask for Fault Pin 3 (Byte 0 - LSB)
35	34	33	GPI Mask for Fault Pin 3
36	35	34	GPI Mask for Fault Pin 3
37	36	35	GPI Mask for Fault Pin 3 (Byte 3 - MSB)
38	37	36	GPI Mask for Fault Pin 4 (Byte 0 - LSB)
39	38	37	GPI Mask for Fault Pin 4
40	39	38	GPI Mask for Fault Pin 4
41	40	39	GPI Mask for Fault Pin 4 (Byte 3 - MSB)
42	41	40	Other Mask

26.2.1 Fault Pin Configuration

A maximum of 4 Fault Pins is supported on the UCD91xxx family of devices. See the Input and Output Pin Configuration section for more details on how to configure the pins.

26.2.2 Page Mask

The Page Mask consists of four configuration bytes. For devices supporting less than 32 rails, or pages, the bytes should be considered as don't cares during write commands. For read commands, the bytes will return 0. The bits are defined as follows:

Bit	**31**	**...**	**21**	**20**	**19**	**18**	**17**	**16**
Purpose	PAGE31	...	PAGE21	PAGE20	PAGE19	PAGE18	PAGE17	PAGE16

Bit	15	14	13	12	11	10	9	8
Purpose	PAGE15	PAGE14	PAGE13	PAGE12	PAGE11	PAGE10	PAGE9	PAGE8

Bit	7	6	5	4	3	2	1	0
Purpose	PAGE7	PAGE6	PAGE5	PAGE4	PAGE3	PAGE2	PAGE1	PAGE0

If a bit is set, any fault of the corresponding page (rail) asserts the fault pin. If a bit is cleared, the faults of the corresponding page (rail) has no effect on the fault pin output.

26.2.3 GPI Mask

The GPI Mask consists of four configuration bytes. For devices supporting less than 32 GPIs, the bytes should be considered as don't cares during write commands. For read commands, the bytes will return 0. The bits are defined as follows:

Table 26-2. GPI Mask

Bit	**31**	**...**	**5**	**4**	**3**	**2**	**1**	**0**
Purpose	GPI 32	...	GPI 6	GPI 5	GPI 4	GPI 3	GPI 2	GPI 1

If a bit is set, any fault of the corresponding GPI asserts the fault pin. If a bit is cleared, the faults of the corresponding GPI has no effect on the fault pin output.

26.2.4 Other Mask

This mask determines the connections between fault pins and system watchdog fault (see SYSTEM_WATCHDOG_CONFIG command in Section 26.5) and re-sequence error. If a bit is set, the corresponding connection is enabled, meaning the fault or error asserts the fault pin.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Purpose	Re-Sequence Error	Re-Sequence Error	Re-Sequence Error	Re-Sequence Error	System Watchdog	System Watchdog	System Watchdog	System Watchdog
	Fault Pin3	Fault Pin2	Fault Pin1	Fault Pin0	Fault Pin 3	Fault Pin 2	Fault Pin 1	Fault Pin 0

26.3 (D1h) VOUT_CAL_MONITOR (MFR_SPECIFIC_01)

This Read/Write Word command is used to apply a fixed offset voltage to the output voltage measured by the device and reported by the READ_VOUT command. It is typically used by the PMBus device manufacturer to calibrate a device in the factory.

The VOUT_CAL_MONITOR has two data bytes formatted as a two's-complement binary integer. The effect on this command depends on the settings of the VOUT_MODE command.

26.4 (D2h) SYSTEM_RESET_CONFIG (MFR_SPECIFIC_02)

This Read/Write Block command configures the system reset function. The system reset function allows the device to provide an external reset signal to the system. This signal can be based on time, the power-good state of selected rails, the state of selected GPI pins, or a combination of these things. This ensures that key devices (for example, a CPU) are held in reset until other dependent devices (for example, peripherals) are fully powered. A reset pulse can also be generated as a result of a System Watchdog Timeout.

Table 26-3. SYSTEM_RESET_CONFIG Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = D2
1	0		BYTE_COUNT = 15
2	1	0	Page Flags – Byte 0 (LSB)
3	2	1	Page Flags – Byte 1
4	3	2	Page Flags – Byte 2
5	4	3	Page Flags – Byte 3 (MSB)
6	5	4	GPI Flags – Byte 0 (LSB)
7	6	5	GPI Flags – Byte 1
8	7	6	GPI Flags – Byte 2
9	8	7	GPI Flags – Byte 3 (MSB)
10	9	8	Delay Time
11	10	9	Pulse Time

Table 26-3. SYSTEM_RESET_CONFIG Command Format (continued)

Byte Number (Write)	Byte Number (Read)	Payload Index	Description	
12	11	10	**Bits**	**Description**
			7:3	GPI Number
			2	Deassert When Power-Good
			1	Assert When NOT Power-Good
			0	Assert when Watchdog Timeout
13	12	11	GPI Tracking	
			Bits	**Description**
			7	Enable
			6:5	Reserved
			4	Watchdog Timeout Assertion Uses GPI Tracking Release Delay
			3:0	GPI Tracking Release Delay (100 µs)
14	13	12	GPI Tracking Release Delay (1 ms)	
15	14	13	Reset Pin Configuration	
16	15	14	Reset Pin Configuration [7:3] Reserved [2] Polarity for Reset Pin [1:0] Pin mode for Reset Pin	

26.4.1 GPI Flags

When “Deassert When Power-Good” is selected, the reset pin is deasserted after the GPI pins identified by these bits reach the asserted state and then the Delay Time passes. (see the *GPI_CONFIG*)

When “Assert When NOT Power-Good” is selected, the reset pin is immediately asserted whenever the any of the GPI pins identified by these bits leaves the asserted state.

Bit	**31**	**...**	**5**	**4**	**3**	**2**	**1**	**0**
Purpose	GPI 31	...	GPI 5	GPI 4	GPI 3	GPI 2	GPI 1	GPI 0

26.4.2 Page Flags

When “Assert When NOT Power-Good” is selected, the reset pin is immediately asserted whenever the any of the pages identified by these bits leaves the power-good state.

When “De-assert When Power-Good” is selected, the reset pin is de-asserted after the pages identified by these bits reach the power-good state and then the Delay Time passes.

Bit	**31**	**...**	**13**	**12**	**11**	**10**	**9**	**8**
Purpose	PAGE31	...	PAGE13	PAGE12	PAGE11	PAGE10	PAGE9	PAGE8

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Purpose	PAGE7	PAGE6	PAGE5	PAGE4	PAGE3	PAGE2	PAGE1	PAGE0

26.4.3 Deassert When Power-Good

When this bit is set, the reset pin is de-asserted whenever the pages identified by the Page Flags reach power-good, the GPI pins identified by the GPI Flags are asserted, and then the time identified by the Delay Time expires.

26.4.4 Assert When NOT Power-Good

When this bit is set, the reset pin is immediately asserted whenever any of the pages identified in the Page Flags leaves the power-good state or any of the GPI pins identified by the GPI Flags are deasserted. Whenever the reset pin is asserted because of the “Assert when NOT Power-Good” function, the device attempts to de-asserted the reset pin based on the Delay Time or a combination of the power good state of selected rails, the asserted state of selected GPIs, and the Delay Time. The Delay Time must be nonzero when this is the only enabled feature of the system reset function.

26.4.5 Assert When Watchdog Timeout

When this bit is set, a system watchdog timeout causes the reset pin to be asserted for the time identified by the Delay Time.

If the “Watchdog Timeout Assertion Uses GPI Tracking Release Delay” bit is set, the GPI Tracking Release Delay is used instead of the Delay Time.

26.4.6 Delay Time

These bits define how long the reset pin is asserted. This byte is formatted according the 8-bit time encoding defined in Section 2.5.

26.4.7 Pulse Time

When this byte is nonzero, the pulse feature is enabled and this byte defines how long the reset pin is deasserted after the Delay Time has passed. This byte is formatted according the 8-bit time encoding defined in Section 2.5.

26.4.8 GPI Tracking

This function allows the system reset pin to be more precisely influenced by a specific GPI pin. (The GPI pin can be a reset signal from another device or a push button.) Whenever the GPI deasserts, the system reset immediately asserts. When the GPI asserts, the system reset is held asserted for the GPI Tracking Release Delay time. After this delay time, the system reset is deasserted. The system reset pin tracks the inverse of the GPI to allow the GPI to be used with the "Deassert When Power-Good" function.

Table 26-4. GPI Tracking Configuration

Configuration Parameter	Description
Enable	This bit enables the GPI Tracking function.
GPI Number	These bits identify which one of the GPI pins is tracked (see the GPI_CONFIG).
GPI Tracking Release Delay (100 µs)	These bits are multiplied by 100 µs and used as the higher precision portion of the GPI Tracking Delay.
GPI Tracking Release Delay (1 ms)	This byte is formatted according the 8-bit time encoding defined in Section 2.5.

The total GPI Tracking Release Delay time is the sum of the 100-µs resolution delay time and the 1-ms resolution delay time.

26.4.9 Reset Pin Configuration

This byte configures the reset pin (see Section 23).

26.5 (D3h) SYSTEM_WATCHDOG_CONFIG (MFR_SPECIFIC_03)

This Read/Write Block common command configures the system watchdog function. This function keeps a timeout counter running. That counter is reset when the watchdog input (WDI) pin is toggled or when the SYSTEM_WATCHDOG_RESET command is written. If the counter is not periodically reset within the amount of time identified by the Reset Period byte, the watchdog output (WDO) pin is asserted. The output pin stays asserted until the watchdog input pin is toggled or until the SYSTEM_WATCHDOG_RESET command is written.

Table 26-5. SYSTEM_WATCHDOG_CONFIG Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description	
0			CMD = D3	
1	0		BYTE_COUNT = 6	
2	1	0	**Bit**	**Description**
			7	Enable
			6	Watch System Reset Pin
			5	Reserved
			4	Disable Until System Reset Release
			3:0	Start Time
3	2	1	Input Pin (WDI) Configuration (Pin ID)	
4	3	2	Input Pin (WDI) Configuration (Other)	
5	4	3	Reset Period	
6	5	4	Output Pin (WDO) Configuration (Pin ID)	
7	6	5	Output Pin (WDO) Configuration (Other)	

26.5.1 Enable

This bit enables the system watchdog function.

26.5.2 Watch Reset Pin

When this bit is set, the System Reset pin (see SYSTEM_RESET_CONFIG, Section 26.4) influences the system watchdog timeout behavior. When the system reset pin is asserted, a watchdog timeout no longer occurs until the reset is deasserted. Once it is deasserted, the system watchdog function waits the Start Time before monitoring the Input Pin again.

26.5.3 Disable Until System Reset Release

When this bit is set, the System Watchdog Reset function is temporarily disabled until the System Reset pin is deasserted. This temporarily disabled state only applies when the device comes out of reset or when the System Watchdog Config command is written.

26.5.4 Start Time

These bits identify the time to delay before monitoring the input pin.

Encoding	Start Time (Seconds)
0000	0
0001	0.1
0010	0.2
0011	0.4
0100	0.8
0101	1.6
0110	3.2
0111	6.4
1000	12.8
1001	25.6
1010	51.2
1011	102.4
1100	204.8
1101	409.6
1110	819.2
1111	1638.4

26.5.5 Input Pin (WDI) Configuration

These bytes identify the Input Pin. The format of these bytes is defined in Section 23.

26.5.6 Reset Period

The system watchdog timeout counter must be reset within the period of time defined by this byte, else the output pin is asserted. Either of these actions resets the counter:

1. Toggle the watchdog input pin (WDI).
2. Write to the SYSTEM_WATCHDOG_RESET command.

This byte is formatted according to Table 26-6. Bits 6 and 7 are the index for the multiplier field and Bits 0 to 5 are the mantissa (0x00 to 0x3F). The two are multiplied together to derive the time. A mantissa of 0 results in a time of 0 regardless of the multiplier value.

Table 26-6. Reset Period Configuration

Multiplier Index	Multiplier (ms)	Resulting Range
b’00	1	0 to 63 ms
b’01	16	16 to 1008 ms
b’10	256	256 ms to 16.128 s
b’11	4096	4096 ms to 258048 ms

Note

The system watchdog does not function as expected when the reset period is set to 0.

26.5.7 Output Pin (WDO) Configuration

This configures the output pin (see Section 23). This pin is asserted if the system watchdog's timeout counter does not receive reset signal or command within the time period defined by the Reset Period byte.

26.6 (D4h) SYSTEM_WATCHDOG_RESET (MFR_SPECIFIC_04)

This write-only, Send Byte command resets the system watchdog's timeout counter. This is the equivalent to toggling the system watchdog Input Pin (WDI).

26.7 (D5h) MONITOR_CONFIG (MFR_SPECIFIC_05)

Table 26-7. MONITOR_CONFIG Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = D5
1	0		BYTE_COUNT = N
2	1	0	Monitor 1 Type and Page
3	2	1	Monitor 2 Type and Page
4	3	2	Monitor 3 Type and Page
↓	↓	↓	↓
N+1	N	N	Monitor N Type and Page

The MONITOR_CONFIG write command configures all monitor pins in one command. The byte position in this command determines which monitor pin is being configured. The size of the write command is variable. The maximum size is determined by the number of monitor pins supported by a given device (see). The UCD91xxx will reject the command if the size exceeds this number.

The monitor pins can be configured individually for one of the various types of monitoring listed in Table 26-8. The command format is shown in *MONITOR_CONFIG Command Format*. The Monitor Type and Page byte format is shown in #none#.

The MONITOR_CONFIG read command always returns information for all of the monitors in a given device.

Table 26-8. Monitor Types

Monitor Type	Encoding
No Monitor	0
Analog Voltage Monitor	1
Temperature	2
Current (Unsupported in UCD91x - Reserved for future use)	3
Voltage Comparator (Unsupported in UCD91x- Reserved for future use)	4
Input Voltage (Unsupported in UCD91x- Reserved for future use)	5
Digital Voltage Monitor	6

Table 26-9. Monitor Type and Page Byte

Bits	**7:5**	**4:0**
Definition	Monitor Type (See Monitor Type)	Monitor Page Number

The status of the monitor pin can be read with the appropriate read command for each monitor type (READ_VOUT, READ_TEMPERATURE_2, READ_IOUT), if supported.

If wanting to use a monitor pin in digital mode (high/low behavior only), the Digital Voltage monitor type should be used. In the case of a digital monitor, the value returned by READ_VOUT will be either a 0, or a 1, rather than a LINEAR16 voltage value. If using a monitoring pin that is digital only, and the analog voltage monitor type is written as configuration, the pin will default to digital monitor only behavior.

The number of monitor pins varies based on the device. Table 26-10 shows the availability for each device.

Table 26-10. Monitor Pin Definitions

Monitor ID (MONx Number)	Monitor Type	Monitor Pin ID	
		UCD91160	UCD91320
1	Analog or Digital	1	1
2	Analog or Digital	2	2
3	Analog or Digital	3	3
4	Analog or Digital	4	4
5	Analog or Digital	5	5
6	Analog or Digital	6	6
7	Analog or Digital	7	7
8	Analog or Digital	8	8
9	Analog or Digital	9	9
10	Analog or Digital	10	10
11	Analog or Digital	11	11
12	Analog or Digital	12	12
13	Analog or Digital	13	13
14	Analog or Digital	14	14
15	Analog or Digital	15	15
16	Analog or Digital	16	16
17	Analog or Digital	N/A	17
18	Analog or Digital	N/A	18
19	Analog or Digital	N/A	19
20	Analog or Digital	N/A	20
21	Analog or Digital	N/A	21
22	Analog or Digital	N/A	22
23	Analog or Digital	N/A	23
24	Analog or Digital	N/A	24
25	Digital Only	N/A	25
26	Digital Only	N/A	26
27	Digital Only	N/A	27
28	Digital Only	N/A	28
29	Digital Only	N/A	29
30	Digital Only	N/A	30
31	Digital Only	N/A	31
32	Digital Only	N/A	32

26.8 (D6h) NUM_PAGES (MFR_SPECIFIC_06)

This read-only, byte command returns the number of active pages. This value is determined by the MONITOR_CONFIG and SEQ_CONFIG commands (see Section 26.7 and Section 26.36). It is the higher page number derived from these two evaluations:

- The highest number page that has an enable pin assigned to it. (SEQ_CONFIG)
- The highest number page that has a monitor pin assigned to it. (MONITOR_CONFIG)

26.9 (D7h) RUN_TIME_CLOCK (MFR_SPECIFIC_07)

This command returns the Run-Time Clock value. It is given in a real time format (year, month, day, hour, minute, second, milisecond), and measures . The Run-Time clock may also be written. This allows the clock to be periodically corrected by the host. It also allows the clock to be initialized to the actual, absolute time in years (for example, March 23, 2025). The user must translate the actual time value to the format shown in Table 26-11.

The three usage scenarios for the Run-Time Clock are:

- Time from power-on: The Run-Time Clock starts from its default time (each time the device is powered on).
- Local time: an external processor sets the Run-Time Clock to real-world time each time the device is powered-on.
- After device reset, the most recent fault log is used to reset the time of the run time clock to the most recent value. If no fault log more recent than the default time is present, the device starts from the default time.

The Run-Time clock value is used to time-stamp any faults that are logged (see Section 26.28).

Table 26-11. RUN_TIME_CLOCK Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = D7
1	0		BYTE_COUNT = 8
2	1	0	Seconds & Miliseconds (Low Byte)
3	2	1	Seconds & Miliseconds (High Byte)
4	3	2	Days, Hours, & Minutes (Low Byte)
5	4	3	Days, Hours, & Minutes (High Byte)
6	5	4	Year & Month (Low byte)
7	6	5	Year & Month (High byte)
8	7	6	Reserved
9	8	7	Reserved

The time format for the Run Time Clock is provided as 3, 2-byte values which contain the current device time formatted in calendar mode. Table 26-12, Table 26-13, and Table 26-14 contain the format for each of these 2-byte values.

Table 26-12. Seconds & Miliseconds Format

Bits	Field	Description
15:10	Seconds	Seconds value from RTC (0 to 59)
9:0	Miliseconds	Miliseconds value from RTC (0 to 999)

Table 26-13. Days, Hours, & Minutes Format

Bits	Field	Description
15:11	Day	Day value from RTC (1 to 31)
10:6	Hours	Hours value from RTC (0-23))
5:0	Minutes	Minutes value from RTC (0 to 59)

Table 26-14. Year & Month Format

Bits	Field	Description
15:4	Year	Year value from RTC (0-4096)
3:0	Month	Month value from RTC (1-12)

26.10 (D8h) RUN_TIME_CLOCK_TRIM (MFR_SPECIFIC_08)

This command takes two linear 11 data values, totaling 4 bytes. The values represent the RUN_TIME_CLOCK offset calibration (in PPM), and temperature calibration values (in PPM/°C) to be applied to the RUN_TIME_CLOCK value. Each of these values may not exceed +/-127 individually. If a larger value is written, it will be retained in the device memory and provided on read back, but will no more than +/-127 PPM (or PPM/°C), will be applied.

The temperature calibration is performed using the internal temperature sensor within the UCD91xxx device, and is applied in reference to 25°C. In total, the applied calibration is the sum of the offset calibration and temperature calibration at the current device temperature, and is limited to +/-240ppm in magnitude.

26.11 (DAh) USER_RAM_00 (MFR_SPECIFIC_10)

This Read/Write Byte command allows the user to read/write a byte value into a RAM location of the device. This RAM value is reset to a known value (0) when the device is reset. By monitoring this value, the user is able to tell whether the device has been reset during operation.

Note that this parameter is not stored to nonvolatile Default Store memory when the STORE_DEFAULT_ALL command is issued.

26.12 (DBh) SOFT_RESET (MFR_SPECIFIC_11)

This Write-Only Send Byte command restarts the controller firmware. Any active voltage outputs are turned off before the firmware restarts.

26.13 (DCh) RESET_COUNT (MFR_SPECIFIC_12)

This Read/Write Byte command allows the user to track how many times the device has been reset. In a device that has not been configured, this value is zero and the number of resets is not tracked. To enable this feature, the value must be changed to a non-zero value and the device must be reset. After that, this value is incremented each time the device is restarted (reset or power-on) and completes the start-up sequence.

If the brownout mode feature is not enabled (see brownout enable mode in MISC_CONFIG Section 26.42) and this feature is enabled, the data flash will be written to each time the device is restarted. If the brownout feature is enabled, this lowers the number of times that the flash is written.

Note

This is a Read/Write byte command. If the reset count exceeds the maximum value (255), it is set to zero and began counting again from there. This rollover occurs 255 times. When the count rolls over the 256th time, the count remains zero and this function is disabled.

26.14 (DDh) PIN_SELECTED_RAIL_STATES (MFR_SPECIFIC_13)

This Read Byte Block command allows use to configure up to 3 GPI pins to control the state of all the rails (up to 8 system states). This function can be used to put the system into a low power mode, for example. In each system state, the user defines which rails are on and which rails are off. If a new state is presented on the GPI, and a rail is required to change state, it does so according to its startup or shutdown dependencies – all changes are done with full sequencing functionality.

Note

The Pin Selected Rail States function is implemented by modifying OPERATION command. Therefore, to use this function to control rail states, the related rails must be configured to use OPERATION command in ON_OFF CONFIG.

Whenever the device is reset, these pins are sampled and used to update the system state, if the function is enabled.

Table 26-15. Changes to the OPERATION Command

New State	Soft Off Enable	OPERATION Command
On	n/a	0x80
Off	0	0x00
Off	1	0x40

The first 3 pins configured with the GPI_CONFIG command (see Section 26.39) can be used to select 1 of 8 system states.

Table 26-16. GPI Selection of System States

GPI 2 State	GPI 1 State	GPI 0 State	System State
NOT Asserted	NOT Asserted	NOT Asserted	0
NOT Asserted	NOT Asserted	Asserted	1
NOT Asserted	Asserted	NOT Asserted	2
NOT Asserted	Asserted	Asserted	3
Asserted	NOT Asserted	NOT Asserted	4
Asserted	NOT Asserted	Asserted	5
Asserted	Asserted	NOT Asserted	6

Table 26-16. GPI Selection of System States (continued)

GPI 2 State	GPI 1 State	GPI 0 State	System State
Asserted	Asserted	Asserted	7

Note

When selecting a new System State, changes to the status of the GPI pins must not take longer than 1 microsecond.

Table 26-17. PIN_SELECTED_RAIL_STATES Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = DD
1	0		BYTE_COUNT = 34
2	1	0	System State Enables
3	2	1	Soft Off Enables
4	3	2	System State 0 (Byte 0 – LSB)
5	4	3	System State 0 (Byte 1)
6	5	4	System State 0 (Byte 2)
7	6	5	System State 0 (Byte 3– MSB)
8	7	6	System State 1 (Byte 0 – LSB)
9	8	7	System State 1 (Byte 1)
10	9	8	System State 1 (Byte 2)
11	10	9	System State 1(Byte 3– MSB)
12	11	10	System State 2 (Byte 0 – LSB)
13	1 2	11	System State 2 (Byte 1)
14	1 3	12	System State 2 (Byte 2)
15	14	13	System State 2(Byte 3– MSB)
16	1 5	14	System State 3 (Byte 0 – LSB)
17	1 6	15	System State 3 (Byte 1)
18	1 7	16	System State 3 (Byte 2)
19	18	17	System State 3(Byte 3– MSB)
20	19	18	System State 4 (Byte 0 – LSB)
21	20	19	System State 4 (Byte 1)
22	21	20	System State 4 (Byte 2)
23	22	21	System State 4(Byte 3– MSB)
24	23	22	System State 5 (Byte 0 – LSB)
25	24	23	System State 5 (Byte 1)
26	25	24	System State 5 (Byte 2)
27	26	25	System State 5(Byte 3– MSB)

Table 26-17. PIN_SELECTED_RAIL_STATES Command Format (continued)

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
28	2 7	26	System State 6 (Byte 0 – LSB)
29	28	27	System State 6 (Byte 1)
30	29	28	System State 6 (Byte 2)
31	30	29	System State 6(Byte 3– MSB)
32	31	30	System State 7 (Byte 0 – LSB)
33	32	31	System State 7 (Byte 1)
34	33	32	System State 7 (Byte 2)
35	34	33	System State 7(Byte 3– MSB)

26.14.1 System State Enables

Each bit in the System State Enables byte is used to enable or disable one of the 8 System States.

Bit	7	6	5	4	3	2	1	0
Purpose	System State 7 Enable	System State 6 Enable	System State 5 Enable	System State 4 Enable	System State 3 Enable	System State 2 Enable	System State 1 Enable	System State 0 Enable

26.14.2 Soft Off Enables

Each bit in the Soft Off Enables byte determines how to turn off (1-soft off or 0-immediate) any rails commanded to turn off by the associated System State.

Bit	7	6	5	4	3	2	1	0
Purpose	System State 7 Soft Off Enable	System State 6 Soft Off Enable	System State 5 Soft Off Enable	System State 4 Soft Off Enable	System State 3 Soft Off Enable	System State 2 Soft Off Enable	System State 1 Soft Off Enable	System State 0 Soft Off Enable

26.14.3 System State

The bits in the System State bytes determine whether or not to change the state of a rail (1 = on, 0 = off).

Bit	31	...	13	12	11	10	9	8
Purpose	PAGE31 ON/OFF	...	PAGE13 ON/OFF	PAGE12 ON/OFF	PAGE11 ON/OFF	PAGE10 ON/OFF	PAGE9 ON/OFF	PAGE8 ON/OFF
Bit	7	6	5	4	3	2	1	0
Purpose	PAGE7 ON/OFF	PAGE6 ON/OFF	PAGE5 ON/OFF	PAGE4 ON/OFF	PAGE3 ON/OFF	PAGE2 ON/OFF	PAGE1 ON/OFF	PAGE0 ON/OFF

Note

NOTE: This table assumes that the device supports 32 rails (0 to 31). For a given device, this may not be the case.

If the device supports less than 32 rails, the extra bytes are treated as don't cares.

26.15 (DEh) RESEQUENCE (MFR_SPECIFIC_14)

This Write block common command is used to command specific rails to resequence. If a rail is commanded to resequence, it will first perform a soft off. After all of the rails that have been commanded to resequence have turned off (the voltage has fallen below POWER_GOOD_OFF), there will be a delay before the rails are commanded to turn back on. The delay is defined by the “Time between Resequences” byte in the MISC_CONFIG command (Section 26.42). After the delay, the rails will turn on according to any dependencies and/or TON_DELAY.

The bits in the payload that make up this command determine if a rail resequences or not (1 = yes, 0 = no).

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = DE
1	0		BYTE_COUNT = 4
2	1	0	Resequence page mask (byte 0 - LSB)
3	2	1	Resequence page mask (byte 1)
4	3	2	Resequence page mask (byte 2)
5	4	3	Resequence page mask (byte 3- MSB)

Bit	31	...	13	12	11	10	9	8
Purpose	PAGE31	...	PAGE13	PAGE12	PAGE11	PAGE10	PAGE9	PAGE8
Bit	7	6	5	4	3	2	1	0
Purpose	PAGE7	PAGE6	PAGE5	PAGE4	PAGE3	PAGE2	PAGE1	PAGE0

26.16 (DFh) CONSTANTS (MFR_SPECIFIC_15)

This Read Byte Block command provides fixed information about the device. Refer to the device specific datasheet for more details.

Table 26-18. CONSTANTS Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = DF
1	0		BYTE_COUNT = 8

Table 26-18. CONSTANTS Command Format (continued)

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
2	1	0	Maximum Number of Digital Comparators
3	2	1	Maximum Number of General-Purpose Outputs (GPOs)
4	3	2	Maximum Number of General-Purpose Inputs (GPIs)
5	4	3	Maximum Number of Pages
6	5	4	Maximum Number of Fans
7	6	5	Maximum Number of Monitors
8	7	6	Maximum Number of Entries in the Logged Fault Detail (see LOGGED_FAULT_DETAIL)
9	8	7	Maximum Number of PWM outputs

26.17 (E0h) PWM_SELECT (MFR_SPECIFIC_16)

This Read/Write Byte command determines which PWM that the PWM_CONFIG command applies to. The value given here must be a valid Pin ID. Refer to Table 23-1 for the Pin ID values of the available PWM pins.

26.18 (E1h) PWM_CONFIG (MFR_SPECIFIC_17)

This Read/Write Byte command configures the PWM identified by the PWM_SELECT command. The duty cycle is in LINEAR11 format and valid values are between 0 and 100 (%). Note that this means that valid values are between 0 and 1, with any value above 1 causing the device to set Invalid Data. Frequency is an unformatted, 32-bit value. See Section 24 for more information about configuring the PWM frequency. To unconfigure a PWM, set the frequency to zero, issue a STORE_DEFAULT_ALL command, and reset the device.

Table 26-19. PWM_CONFIG Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = E1
1	0		BYTE_COUNT = 8
2	1	0	Duty Cycle (Low Byte)
3	2	1	Duty Cycle (High Byte)
4	3	2	Frequency (Low Byte)
5	4	3	Frequency
6	5	4	Frequency
7	6	5	Frequency (High Byte)
8	7	6	Reserved
9	8	7	Reserved

26.19 (E2h) PARM_INFO (MFR_SPECIFIC_18)

This Read/Write Block command is used to configure the parameters used by the PARM_VALUE command.

The PARM_INFO command updates four variables that are needed to issue a generic read/write of RAM or hardware registers, and so forth. The four variables are parm_index, parm_offset, parm_count, and parm_size.

Their descriptions follow:

	Parm_index – Index for base address	
		0 = RAM
		1 = Hardware Peripherals
		2 = Constants in Data Flash (Read Only, unless unlocked)
		3 = Constants in Program Flash (Read Only)
		4 = Data Flash Control Registers
		5 = EEPROM (Read only, with Parm_size = 4 only)

	Parm_offset – offset from the base address selected by parm_base. Parm_count – number of elements to read or write Parm_size – the size of each element in bytes. (Valid values are 1, 2, or 4).

PARM_INFO and PARM_VALUE are combined to provide a method for reading or writing to any RAM address or hardware register. A map file specific to the firmware release may be required to determine the offset for a particular RAM variable, because variables may be in different locations for each release.

Table 26-20. PARM_INFO Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = E2
1	0		BYTE_COUNT = 6
2	1	0	Index
3	2	1	Offset low byte
4	3	2	Offset middle byte
5	4	3	Offset high byte
6	5	4	Count
7	6	5	Size

26.20 (E3h) PARM_VALUE (MFR_SPECIFIC_19)

This Read/Write Block command is used to read and write to RAM addresses or hardware peripheral registers. This command assumes that the PARM_INFO command has be previously run to set up the parm_base, parm_offset, parm_count, and parm_size variables as needed.

26.21 (E4h) TEMPERATURE_CAL_GAIN (MFR_SPECIFIC_20)

This Read/Write Word command sets the gain calibration for the external sensors used by the READ_TEMPERATURE_2 command. Each external temperature sensor (typically one per power output rail) has its own calibration setting.

This command has two data bytes formatted in the Linear11 Data format. The units are Celsius degrees per volt.

26.22 (E5h) TEMPERATURE_CAL_OFFSET (MFR_SPECIFIC_21)

This Read/Write Word command sets the offset calibration for the external sensors used by the READ_TEMPERATURE_2 command. Each external temperature sensor (typically one per power output rail) has its own calibration setting.

This command has two data bytes formatted in the Linear11 Data format. The units are degrees Celsius.

26.23 (E6h) SET_BREAKPOINTS (MFR_SPECIFIC 22)

This paged read//write word command sets breakpoints in the UDC91xx device such that a given page can be made to stop before proceeding to the next rail state (See the RAIL_STATE command for more infomration on device available rail states). While halted, the page will return the BREAKPOINT state upon read of the RAIL_STATE command.

Note

This command is intended to be used for debug use only, not as part of a regular device startup sequence.

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
1	0	0	Stop-on states byte (See Table 26-21)
2	1	1	Stop-from states byte (See Table 26-21)

The two bytes sent in this command reflect the given state transitions to break at. In order to stop at a given state, a pair of conditions from both the "Stop-on" byte, and the "Stop-off" byte must be met. For example, If it is desired to stop before begining the START_DELAY function after a rail's dependencies are met, then both the START_DELAY bit in the "Stop-on" states byte, and the SEQ_ON bit in the "Stop-from" states byte must be set. If only one of these is set, the rail will not enter the breakpoint state, and will instead enter the START_DELAY state as normal. The page will remain in the BREAKPOINT state until a DEBUG_CONTINUE command (see Section 26.24) is received which indicates the page should continue transitioning.

Bits in these bytes are not exclusive, and may be simultaneously set to create more than a single condition where the given page will state transitions.

Table 26-21. Breakpoint State Byte Format

Bit	7	6	5	4	3	2	1	0
Purpose	RAMP_DOWN	STOP_DELAY	SEQ_OFF	REGULATION	RAMP_UP	START_DELAY	SEQ_ON	IDLE

26.24 (E7h) DEBUG_CONTINUE (MFR_SPECIFIC_23)

This common read/write block command instructs a given page to continue from the BREAKPOINT state. If a page is not currently in the BREAKPOINT state, then writing this command has no effect. The value returned by the command will indicate the flag for the given page is set, but if the condition for the page described in Section 26.23 is hit, the flag will be cleared and the page will halt as instructed by that command. It will not continue until the DEBUG_CONTINUE command is issued again.

The bits included in this command indicate whether a given page continues or not (1=yes, 0= no)

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = E7h
1	0		BYTE_COUNT = 4
2	1	0	Debug continue page mask (Byte 0 -LSB)
3	2	1	Debug continue page mask (Byte 1)
4	3	2	Debug continue page mask (Byte 2)
5	4	3	Debug continue page mask (Byte 3)

Bit	31	...	5	4	3	2	1	0
Purpose	Page 31	...	Page 5	Page 4	Page 3	Page 2	Page 1	Page 0

26.25 (E9h) FAULT_RESPONSES (MFR_SPECIFIC_25)

This paged, read/write block command sets the response to each fault condition. This command is used instead of the following standard PMBus commands:

- VOUT_OV_FAULT_RESPONSE
- VOUT_UV_FAULT_RESPONSE

- IOUT_OC_FAULT_RESPONSE
- IOUT_UC_FAULT_RESPONSE
- OT_FAULT_RESPONSE
- TON_MAX_FAULT_RESPONSE

As with the original PMBus fault response commands, whenever a fault occurs:

- The corresponding fault bit is the status register is set.
- The PMBus ALERT pin is asserted.
- The fault bit, once set, is cleared only in accordance to Section 10.2.3 in the PMBus specification and not when the fault condition is removed or corrected.

The command format is shown in Table 26-22.

Table 26-22. FAULT_RESPONSES Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = E9
1	0		BYTE_COUNT = 9
2	1	0	VOUT_OV Fault Response
3	2	1	VOUT_UV Fault Response
4	3	2	IOUT_OC Fault Response
5	4	3	IOUT_UC Fault Response
6	5	4	OT Fault Response
7	6	5	TON_MAX Fault Response
8	7	6	Time between retries
9	8	7	Maximum glitch time for voltage faults
10	9	8	Maximum glitch time for non-voltage faults

26.25.1 Fault Response Bytes

The Fault Response bytes (the first eight bytes) are formatted as shown in Table 26-23.

Table 26-23. Fault Response Byte Format

Bits	Description	Value	Meaning
7	Operation	n/a	When this bit is set to one, the device shuts down (disables the output) and responds according to the retry setting in bits [3:0]. The function associated with the following Glitch Filter bit, can delay or prevent the shutdown.
			When set to zero, the PMBus device continues operation without interruption.
6	Glitch Filter	n/a	When this bit is set to one, the device continues operation for a delay time specified by the “Maximum glitch time for voltage faults” or the “Maximum glitch time for non-voltage faults” byte. If the fault condition is removed during the delay time, the timer resets, and the fault is ignored. If the fault condition is present for longer than the delay time, the device responds to the fault as programmed in the Retry Setting (bits[3:0]).
5	Soft Stop	n/a	If this bit is set to 1, the rail comes to a soft stop (using TOFF_DELAY). If it is set to 0, the rail shuts down immediately.
4	Re-sequence	n/a	When this bit is set to 1 and the retries have been exhausted, the rail and its Fault Slaves are shutdown in a manner based on the Soft Stop bit. All of those rails are resequenced after a delay time defined by the “Time between Resequences” byte in the MISC_CONFIG command (see Section 26.42)

Table 26-23. Fault Response Byte Format (continued)

Bits	Description	Value	Meaning
3:0	Retry Setting	0000	A 0 value for the Retry Setting means that the device does not attempt to restart the rail. The rail remains off until a turn-off and then turn-on command (by OPERATION command or CONTROL pin or Pin-Selected State) are received.
		0001–1110	The device attempts to restart the rail for the number of times set by these bits. The minimum number is 1 and the maximum number is 14 when the soft stop bit is not set. When the soft stop bit is enabled, one less retry will be attempted. If the rail fails to restart successfully within the allowed number of retries, the device disables the rail and remains off unless Resequence bit is set. The rail can be also turned back on according to the conditions described in Section 10.7 of PMBus specification. The time between the start of each restart attempt is set by the ”Time between retries” byte.
			Note: This retry count is reset whenever the rail stays in regulation for a TON_MAX_FAULT_LIMIT amount of time without having a glitch. (If TON_MAX_FAULT_LIMIT is set to 0, 4 seconds are used for the time.) Glitches on faults where the Operation bit is set to zero are ignored.
		1111	The device attempts to restart the rail continuously, without limitation, until it is commanded OFF (by the CONTROL pin or OPERATION command or both), bias power is removed, or another fault condition causes the rail to shut down.

26.25.2 Re-Sequence

The option to re-sequence is offered as a response to each type of fault. The re-sequence operation is not performed until the retries are exhausted for a given fault. During the re-sequence operation:

- The faulty rail and its Fault Shutdown Slaves (FSS) rails are immediately disabled (Stop Immediately) or disabled according to the configured shutdown sequence (Stop With Delay)
- After the faulted rail and its FSS rails' Enable pins are deasserted and all of these rails have turned off (the voltage has fallen below POWER_GOOD_OFF), the UCD91xxx waits for a programmable delay time before starting the resequence. The delay time is configured by the "Time between Resequences" byte in the MISC_CONFIG command, (see Section 26.42). While waiting for the rails to turn off, if any rail voltage does not fall below 12.5% of nominal output voltage within a time period defined by TOFF_MAX_WARN_LIMIT, a TOFF_MAX_WARN warning occurs. Software can configure whether to abort the resequence when the TOFF_MAX_)WARN warning occurs. For more information, see the MISC_CONFIG command (Section 26.42). It is configurable to ignore the POWER_GOOD_OFF and TOFF_WARNING status of the rail when resequencing, (see Section 26.42). This is mainly for those rails whose EN signals are not controlled by UCD.
- After the resequence delay time, the faulty rail and its FSS rails are sequenced on according to the start-up sequence configuration. The resequence attempt can repeat 1-4 times or unlimited times to bring the faulty rail and its FSS rails to regulation. The maximum resequence times is configured in the Section 26.42. If a resequence operation is successful – that is, all of the rails that were resequenced maintain in normal operation (power-good and no retries) for 1 second – the resequence counter is reset. (If a resequenced rail is not configured to turn on during the resequence, the device does not wait for that rail to come on before declaring the resequence operation is successful.) If the rails are resequenced the maximum number times and they fail to reach normal operation, a device reset is required to reset the resequence counter. Rails can be commanded off and then on in an attempt to get them back to normal operation.

This resequence operation is straight-forward if there is only one set of faulty rail and FSS rails. If two or more sets of rails require resequence at the same time, the device behavior is always conservative. For example, if a set of rails is already on its second resequence, and the device is configured to resequence three times, and another set of rails enters the resequence state, the second set of rails are only resequenced once so that the total resequence time is 3. Another example – if one set of rails is awaiting all of its rails to shutdown so that it can resequence, and another set of rails enters the resequence state, the device now waits for all rails from both sets to shutdown before resequencing.

If any rails at the resequence state are caused by GPI fault response, the whole resequence is suspended until the GPI fault is physically clear.

26.25.3 Time Between Retries

When configured to retry, this value determines the amount of time that the device waits before it tries to re-enable a rail after it was shutdown due to a fault.

This byte is formatted according to Section 18.5. The minimum value is 1 ms.

26.25.4 Maximum Glitch Time for Voltage Faults

The value in this byte is multiplied by 400 µs to get the "Maximum glitch time for voltage faults". This applies to the following faults: VOUT_OV, VOUT_UV.

26.25.5 Maximum Glitch Time for Non-Voltage Faults

The value in this byte is multiplied by 100 ms to get the "Maximum glitch time for non-voltage faults". This applies to the following faults: IOUT_OC, IOUT_UC, OT.

Note

The TON_MAX Fault Response does not support glitch filtering.

26.26 (EAh) LOGGED_FAULTS (MFR_SPECIFIC_26)

This read/write block command reports a history of all faults that have ever been reported and logged into nonvolatile memory.

Clearing the Log: Writing a block resets all logged entries to 0. This also clears the LOGGED_FAULT_DETAIL entries (see Section 26.28). Clearing the log does not clear the non-volatile memory if brownout function is enabled. For the applications where brownout feature is required, the user must disable the brownout function before issuing a command to clear the log, and re-enable the brownout function after the log is cleared.

26.26.1 Command Format

This command returns a binary array in the order shown in Table 26-24. Note that the command sends information of all pages, even when some pages are not active. Log entries of inactive pages are reported as 0x00.

Table 26-24. LOGGED_FAULTS Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = EA
1	0		BYTE_COUNT = 37
2	1	0	Non-Paged Faults
3	2	1	GPI Faults (Byte 0 - LSB)
4	3	2	GPI Faults (Byte 1)
5	4	3	GPI Faults (Byte 2)
6	5	4	GPI Faults (Byte 3 - MSB)
7	6	5	Page 0 Faults
. . .	. . .	. . .	Page 1 Faults
. . .	. . .	. . .	. . .
38	37	36	Page 31Faults

26.26.2 Non-Paged Faults

The bit definitions for common faults are shown in Table 26-25.

By examining this single bit (bit 0 of the Non-Paged Faults byte), a host can determine if any page-dependent or GPI faults have occurred. A value of 0 in this bit indicates that all of the page-dependent and GPI fault log entries are 0 and need not be read. A value of 1 in this bit indicates that one or more of the page-dependent, or GPI faults has occurred. In that case, the host must examine all of the log entries to determine which ones have log information.

Table 26-25. Non-Paged Fault Log Bit Definitions

Bit	Description	Can Generate a LOGGED_FAULT_DETAIL Entry?
0	LOG_NOT_EMPTY	No
1	System Watchdog Timeout	Yes
2	Resequence Error	No
3	Watchdog Timeout	Yes
4	Reserved	No
5	Reserved	No
6	Reserved	No
7	Reserved	No

26.26.3 GPI Faults

The bit definitions for GPI faults are shown in Table 26-26. These bits are set whenever the associated GPI is deasserted.

Table 26-26. GPI Fault Log Bit Definitions

Bit	Description	Can Generate a LOGGED_FAULT_DETAIL Entry?
		Byte 0 (LSB)
0	GPI_0	Yes
1	GPI_1	Yes
2	GPI_2	Yes
3	GPI_3	Yes
4	GPI_4	Yes
5	GPI_5	Yes
6	GPI_6	Yes
7	GPI_7	Yes
		Byte 1
0	GPI_8	Yes
1	GPI_9	Yes
2	GPI_10	Yes
3	GPI_11	Yes
4	GPI_12	Yes
5	GPI_13	Yes
6	GPI_14	Yes
7	GPI_15	Yes
		Byte 2
0	GPI_16	Yes
1	GPI_17	Yes
2	GPI_18	Yes
3	GPI_19	Yes
4	GPI_20	Yes
5	GPI_21	Yes
6	GPI_22	Yes
7	GPI_23	Yes
		Byte 3
0	GPI_24	Yes
...	...	...

Table 26-26. GPI Fault Log Bit Definitions (continued)

Bit	Description	Can Generate a LOGGED_FAULT_DETAIL Entry?
7	GPI_31	Yes

26.26.4 Page-Dependent Faults

The bit definitions for page-dependent faults are shown in Table 26-27.

Table 26-27. Page-Dependent Fault Log Bit Definitions

Bit	Description	Can Generate a LOGGED_FAULT_DETAIL Entry?
0	VOUT_OV Fault	Yes
1	VOUT_UV Fault	Yes
2	TON_MAX Fault	Yes
3	IOUT_OC Fault	Yes
4	IOUT_UC Fault	Yes
5	TEMPERATURE_OT Fault	Yes
6	Sequence On Timeout	Yes
7	Sequence Off Timeout	Yes

26.27 (EBh) LOGGED_FAULT_DETAIL_INDEX (MFR_SPECIFIC_27)

The read form of this command reports the total number of LOGGED_FAULT_DETAIL entries and the current value of the index into those entries (see Section 26.28). The Fault index byte can be written.

Table 26-28. LOGGED_FAULT_DETAIL_INDEX Command Format

Byte Number	Description
1	Fault Index(R/W)
2	Total Number of LOGGED_FAULT_DETAIL entries (Read Only))

26.28 (ECh) LOGGED_FAULT_DETAIL (MFR_SPECIFIC_28)

This read-only command returns detail information about a given fault:

1. The time that it happened in calendar formatted time (see Section 26.9)
2. Whether or not the fault is page specific
3. The page that the fault occurred on (when applicable)
4. The type of fault (see Section 26.26)

Table 26-29. LOGGED_FAULT_DETAIL Command Format

Byte Number (Read)	Payload Index	Description
		CMD = EC
0		BYTE_COUNT = 12
1	0	Seconds & Miliseconds (Low Byte)
2	1	Seconds & Miliseconds (High Byte)
3	2	Days, Hours, & Minutes (Low Byte)
4	3	Days, Hours, & Minutes (High Byte)
5	4	Year & Month (Low Byte)
6	5	Year & Month (High Byte)
7	6	Page Number
8	7	Fault ID
9	8	Fault Value (low byte)
10	9	Fault Value (middle byte)
11	10	Fault Value (middle high byte)(1)

Table 26-29. LOGGED_FAULT_DETAIL Command Format (continued)

Byte Number (Read)	Payload Index	Description
12	11	Fault Value (high byte)(1)

(1) This byte is valid only if the current fault type is either Sequence On Timeout or Sequence Off Timeout.

New LOGGED_FAULT_DETAIL entries for a given fault are logged once and then not logged again until one of the following events occurs:

1. Firmware is restarted.
2. The rail stays in regulation for the amount of time determined by TON_MAX_FAULT_LIMIT. If TON_MAX_FAULT_LIMIT is set to 0, 4 seconds are used for the time. (This applies to paged faults only)
3. The rail is turned off and then turned back on. (This applies to paged faults only.)
4. The CLEAR_FAULTS command is written.
5. The LOGGED_FAULTS command is written.

For temperature only rails, meaning rails with only a temperature monitor and no Enable pin assigned, an overtemperature fault will be logged once only, unless operating from a stored configuration.

The GPI fault is logged each time when the GPI state is changed to deasserted.

The maximum number of entries per device is show in Table 26-30. Once the maximum entries have been logged, no more detail is logged until the fault log is cleared (see Section 26.26). Unless, the “Enable Log FIFO” bit in the MISC_CONFIG command is set (see Table 26-46).

Table 26-30. Number of Log Entries in Each Device Type

Device	Entries
UCD91160	100
UCD91320	100

The fault identification byte indicates the type of fault, and whether or not that fault is page specific, see Table 26-31.

Table 26-31. Fault Identification

Bits	Description
7	Page Specific (1 – yes, 0 – no)
6:0	Fault Type

Information on the Fault Value (units and formatting) is shown inTable 26-32.

Table 26-32. Fault Value

Fault Type Number	Paged?	Description	Fault Value Units	Data Format
0	No	Reserved		
1	No	System Watchdog Timeout	Not Valid	n/a
2	No	Re-sequence Error(3)	Not Valid	n/a
3	No	Watchdog Timeout	Not Valid	n/a
4	No	Single Event Upset (SEU)		n/a
5	No	Reserved		n/a
6	No	Reserved		n/a
7	No	Reserved		n/a
8	No	Reserved		n/a
9	No	GPI Fault(1)	Not Valid	n/a
0	Yes	VOUT_OV Fault	Voltage	LINEAR16

Table 26-32. Fault Value (continued)

Fault Type Number	Paged?	Description	Fault Value Units	Data Format
1	Yes	VOUT_UV Fault	Voltage	LINEAR16
2	Yes	TON_MAX Fault	Voltage	LINEAR16
3	Yes	Reserved		n/a
4	Yes	Reserved		n/a
5	Yes	TEMPERATURE_OT Fault	Temperature	LINEAR11
6	Yes	Sequence On Timeout	N/A	Bit Mask(2)
7	Yes	Sequence Off Timeout	N/A	Bit Mask(2)

(1) The Page Number is used to encode which GPI that the fault information applies to.
(2) Any bit set to 1 corresponds to a page dependency that is not met. The GPI dependencies that are not met are ORed into the 32-bits. So, for example, if page dependencies 2 and 5 are not met and GPI W (bit 4) dependency is not met, the fault value is 0x00000034.
(3) A fault value is reserved for resequencing errors, but they are not logged.

26.29 (EFh) LOG_FAULT_DETAIL_ENABLES (MFR_SPECIFIC_31)

This Read/Write Block command selects what fault detail (see Section 26.28) is logging by rail and by fault type. The command, the bytes in the command, and the bits in those bytes are formatted the same as the LOGGED_FAULTS command, see Section 26.26.1. For this LOGGED_FAULT_DETAIL_ENABLES command, the bits select if a fault is logged (1) or not (0).

Table 26-33. LOGGED_FAULT_DETAIL_ENABLES Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = EF
1	0		BYTE_COUNT = 37
2	1	0	Non-paged Enable Flags
3	2	1	GPI Enable Flags - Byte 0 (LSB)
4	3	2	GPI Enable Flags - Byte 1
5	4	3	GPI Enable Flags - Byte 2
6	5	4	GPI Enable Flags - Byte 3 (MSB)
7	6	5	Page Enable Flags – Page 0
8	7	6	Page Enable Flags – Page 1
9	8	7	Page Enable Flags – Page 2
10	9	8	Page Enable Flags – Page 3
11	10	9	Page Enable Flags – Page 4
12	11	10	Page Enable Flags – Page 5
13	12	11	Page Enable Flags – Page 6
14	13	12	Page Enable Flags – Page 7
15	14	13	Page Enable Flags – Page 8
16	15	14	Page Enable Flags – Page 9
17	16	15	Page Enable Flags – Page 10
18	17	16	Page Enable Flags – Page 11
19	18	17	Page Enable Flags – Page 12
20	19	18	Page Enable Flags – Page 13
21	20	19	Page Enable Flags – Page 14
22	21	20	Page Enable Flags – Page 15

Table 26-33. LOGGED_FAULT_DETAIL_ENABLES Command Format (continued)

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
23	22	21	Page Enable Flags – Page 16
24	23	22	Page Enable Flags – Page 17
25	24	23	Page Enable Flags – Page 18
26	25	24	Page Enable Flags – Page 19
27	26	25	Page Enable Flags – Page 20
28	27	26	Page Enable Flags – Page 21
29	28	27	Page Enable Flags – Page 22
30	29	28	Page Enable Flags – Page 23
...	...	...	
38	37	36	Page Enable Flags – Page 31

26.30 (F0h) EXECUTE_FLASH (MFR_SPECIFIC_32)

This command is not supported on UCD91xxx.

26.31 (F1h) SECURITY (MFR_SPECIFIC_33)

This command is not supported on UCD91xxx. Similar functionality is supported in the ACCESS_CONTROL command.

26.32 F2h) SECURITY_BIT_MASK (MFR_SPECIFIC_34)

This command is not supported on UCD91xxx.

26.33 (F3h) MFR_STATUS (MFR_SPECIFIC_35)

The Manufacturer Status bits are defined in Table 26-34. With the exception of the STORE_DEFAULT_ALL_DONE bit, whenever any of these bits are set, the PMBALERT# line is asserted.

Table 26-34. Manufacturer Specific Status

Byte Number (write)	Byte Number (Read)	Payload Index	Bit in Each Payload Byte	Name	Descriptions
0					CMD = F3
1	0				BYTE_COUNT = 6
2	1	0	0	GPI25 Fault	Set whenever the GPI is de-asserted
...	...	...	...	...	...
2	1	0	7	GPI32 Fault	
3	2	1	0	GPI17 Fault	
...	...	...	...	...	
3	2	1	7	GPI24 Fault	
4	3	2	0	GPI9 Fault	
...	...	...	...	...	
4	3	2	7	GPI16 Fault	
5	4	3	0	GPI1 Fault	
...	...	...	...		

Table 26-34. Manufacturer Specific Status (continued)

Byte Number (write)	Byte Number (Read)	Payload Index	Bit in Each Payload Byte	Name	Descriptions
5	4	3	7	GPI8 Fault	Set whenever the GPI is de-asserted
6	5	4	0	WATCHDOG_TIMEOUT	A UCD91xxx internal watchdog timeout reset has occurred
			1	STORE_DEFAULT_ALL_DONE	The STORE_DEFAULT_ALL operation has completed
			2	STORE_DEFAULT_ALL_ERROR	An error occurred during STORE_DEFAULT_ALL
			3	SYSTEM_WATCHDOG_TIMEOUT	A system watchdog timeout reset has occurred (see Section 26.5)
			4	NEW_ LOGGED_FAULT_DETAIL	This bit is set whenever a new fault detail entry is logged. It is cleared whenever the LOGGED_FAULT_DETAIL command is read.
			5	STORE_PARM_VALUE_ERROR	An error occurred during downloading the configuration
			6	SINGLE_EVENT_UPSET	Single event upset occurred
			7	LFCLK_FAILURE	A failure of the low frequency clock used for the RTC has occured. This is only relevant when the LFCLK enable bit is set in the MISC_CONFIG command.
7	6	5	0	SLAVED_FAULT	The rail was shutdown because it is dependent on another rail that had a fault (paged) [See Section 26.36.9]
			1	SEQ_ON_TIMEOUT	Sequence on timeout (paged) [See Section 26.36.4]
			2	SEQ_OFF_TIMEOUT	Sequence off timeout (paged) [See Section 26.36.4]
			3	HARDCODED_PARMS	PMBus hard-coded configuration values have been loaded into RAM. The data flash image on the device is empty or corrupt. This state is referred to as a NOBOARD configuration. This default configuration only minimally configures the device (for example, no rails are defined). Executing a STORE_DEFAULT_ALL command updates the data flash on the device. After a STORE_DEFAULT_ALL and reset, this HARDCODED_PARMS status bit is cleared. It also can be cleared with the CLEAR_FAULTS command.
			4	PKGID_MISMATCH	Hardware Package ID does not match firmware
			5	RESEQUENCE_ERROR	An error occurred during the re-sequencing operation. This occurs when one of the rails that are part of the resequence operation does not turn off before its TOFF_MAX_WARN_LIMIT.
			6	LOGGED_FAULT_DETAIL_FULL	The LOGGED_FAULT_DETAIL buffer is full [See Section 26.26)
			7	INVALID_LOG	A checksum match failed for a log during non-volatile memory check.

26.34 (F4h) GPI_FAULT_RESPONSES (MFR_SPECIFIC_36)

This paged Read/Write Block command configures the rail response to GPI faults. The fault flag must be set for the corresponding GPI via GPI_CONFIG(F9h) command, or this command is ignored. GPI is treated as fault only if it is transited from assertion to deassertion. See Section 26.39.2 for more details.

Table 26-35. GPI_FAULT_RESPONSES Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = F4
1	0		BYTE_COUNT = 39
2	1	0	GPI Fault Responses – GPI 0

Table 26-35. GPI_FAULT_RESPONSES Command Format (continued)

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
3	2	1	GPI Fault Responses – GPI 1
4	3	2	GPI Fault Responses – GPI 2
5	4	3	GPI Fault Responses – GPI 3
6	5	4	GPI Fault Responses – GPI 4
7	6	5	GPI Fault Responses – GPI 5
8	7	6	GPI Fault Responses – GPI 6
9	8	7	GPI Fault Responses – GPI 7
10	9	8	GPI Fault Responses – GPI 8
11	10	9	GPI Fault Responses – GPI 9
12	11	10	GPI Fault Responses – GPI 10
13	12	11	GPI Fault Responses – GPI 11
14	13	12	GPI Fault Responses – GPI 12
15	14	13	GPI Fault Responses – GPI 13
16	15	14	GPI Fault Responses – GPI 14
17	16	15	GPI Fault Responses – GPI 15
18	17	16	GPI Fault Responses – GPI 16
19	18	17	GPI Fault Responses – GPI 17
20	19	18	GPI Fault Responses – GPI 18
21	20	19	GPI Fault Responses – GPI 19
22	21	20	GPI Fault Responses – GPI 20
23	22	21	GPI Fault Responses – GPI 21
24	23	22	GPI Fault Responses – GPI 22
25	24	23	GPI Fault Responses – GPI 23
26	25	24	GPI Fault Responses – GPI 24
27	26	25	GPI Fault Responses – GPI 25
28	27	26	GPI Fault Responses – GPI 26
29	28	27	GPI Fault Responses – GPI 27
30	29	28	GPI Fault Responses – GPI 28
31	30	29	GPI Fault Responses – GPI 29
32	31	30	GPI Fault Responses – GPI 30
33	32	31	GPI Fault Responses – GPI 31
34	33	32	Time between retries
35	34	33	Maximum glitch time for GPI(low byte)
36	35	34	Maximum glitch time for GPI(high byte)
37	36	35	GPI number Rail Profile Selection 1
38	37	36	GPI number Rail Profile Selection 2
39	38	37	Block Out period for profile (low byte)
40	39	38	Block Out period for profile (high byte)

26.34.1 Fault Responses Byte

Refer to Section 26.25.1 for the byte definition of Fault Responses Byte, retry action is not supported by GPI fault response.

26.34.2 Time Between Retries

When configured to do so, this value determines the delay time before the device tries to re-enable a rail after it was shutdown due to a fault. The byte is formatted according to Section 18.5. This value has to be 1 ms or bigger.

Note

This value should be written to match the desired time between retries for the FAULT_RESPONSES command if used. The UCD91xxx family of devices supports only a single setting for time between retries, and therefore writing different values for these commands will overwrite the previous one.

26.34.3 Maximum Glitch Time for GPI

The value in this byte is multiplied by 300 μs to get the Maximum glitch of GPI input.

26.34.4 GPI Number Rail Profile Pin Selection

Rail Profile Selection can be enabled with the input pin identified by this byte. The function is ignored if the value of this byte is 0 and the corresponding profile selection state is default at de-asserted. When this function is enabled, the assigned GPI pin is used to select the predefine Rail Profile based on the asserted status. See RAIL_PROFILE_INDEX for more details on the pre-defined rail profile. Moreover, the assigned GPI pin cannot be used for other functions, such as Fault response, GPO output. It is application’s responsibility to make sure the setting are correct, Device does not perform any checks about these. Each Rail could have up to 2 GPI as rail profile selection.

Table 26-36. Rail Profile Pin Selection

Rail Profile Index	GPI Number Rail Profile Selection 2	GPI Number Rail Profile Selection 1	Rail has 1 Profile	Rail has 2 Profile	Rail has 3 Profile	Rail has 4 Profile
0	De-asserted	De-asserted	0	0	0	0
1	De-asserted	Asserted	0	1	1	1
2	Asserted	De-asserted	0	0	2	2
3	Asserted	Asserted	0	1	2	3

26.34.5 Block Out Period for Profile

The value in these two bytes multiplied by 400 μs defines the block out period which blocks all voltage related fault on the giving rail when profile is changed. Block out period is only reloaded when there is profile switched.

26.35 (F5h) MARGIN_CONFIG (MFR_SPECIFIC_37)

This paged Read/Write Byte command configures if a given rail can be margined, and if so, how. Rails using digital monitoring do not support margining.

Table 26-37. MARGIN_CONFIG Command Format

Bits	Name	Description
		Byte 0

Table 26-37. MARGIN_CONFIG Command Format (continued)

Bits	Name	Description	
7:0	PWM Pin	Selects the PWM pin. This value is a MARx Pin ID. Refer to the device datasheet for more details.	
		The frequency for the PWM is configured with the PWM_CONFIG command.	
		Byte 1	
7:6	Mode[1]	b'00:	DISABLE - Margining is disabled
		b'01:	ENABLE_HIGH_IMPEDANCE - When not margining, the PWM pin is put in a high-impedance state
		b'10:	ENABLE_ACTIVE_TRIM - When not margining, the PWM duty cycle is continuously adjusted to keep the voltage at VOUT_COMMAND
		b'11:	ENABLE_FIXED_DUTY_CYCLE - When not margining, the PWM duty cycle is set to a fixed duty cycle
5	Ignore Faults	When margining is enabled with a pin, this bit determines if faults (over-voltage and under-voltage) are ignored or not. This can overwrite, or be overwritten by, the equivalent setting in the OPERATION command.	
4	Duty Cycle	This bit determines the relationship between the duty cycle and the output voltage. The output voltage increases when the duty cycle:	
			1 – increases 0 – decreases
3	Nominal Duty Cycle Mode	When set, the nominal duty cycle mode is enabled, and the PWM pins will start from, and return to, the DUTY_CYCLE value set in PWM_CONFIG when enabled/disabled. See Margining for more details.	
2:0	Reserved	Reserved	

(1) If the Mode is not DISABLE, the associated PWM_CONFIG command must be written before the MARGIN_CONFIG command is written.

26.36 (F6h) SEQ_CONFIG (MFR_SPECIFIC_38)

This Read/Write Block command configures the sequencing dependencies and enable pin for a given rail.

Features include:

1. Sequencing – Configures interdependency between how voltage rails are enabled/disabled
2. Fault Slaves – Configure slave pages which also shut down when a fault occurs
3. Enable Pin – Identifies the enable pin, and its operating characteristics.

Note

All configurations done with the GPI_CONFIG command must be done before writing this command.

When this command is written, the enable pin for the rail is deasserted. Then the state of the rail is re-evaluated. If it is determined that the rail should be on, the enable pin is then asserted.

Table 26-38. SEQ_CONFIG Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = F6
1	0		BYTE_COUNT = 29
2	1	0	Enable Pin Configuration (Pin ID)
3	2	1	Enable Pin Configuration (Other)
4	3	2	GPI Sequence On Dependency Mask (Byte 0 - LSB)
5	4	3	GPI Sequence On Dependency Mask (Byte 1)
6	5	4	GPI Sequence On Dependency Mask (Byte 2)
7	6	5	GPI Sequence On Dependency Mask (Byte 3 - MSB)
8	7	6	GPI Sequence Off Dependency Mask (Byte 0 - LSB)

Table 26-38. SEQ_CONFIG Command Format (continued)

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
9	8	7	GPI Sequence Off Dependency Mask (Byte 1)
10	9	8	GPI Sequence Off Dependency Mask (Byte 2)
11	10	9	GPI Sequence Off Dependency Mask (Byte 3- LSB)
12	11	10	Sequencing Timeout Configuration
13	12	11	Sequencing On Timeout
14	13	12	Sequencing Off Timeout
15	14	13	Page Sequence On Dependency Mask (Byte 0 - LSB)
16	15	14	Page Sequence On Dependency Mask (Byte 1)
17	16	15	Page Sequence On Dependency Mask (Byte 2)
18	17	16	Page Sequence On Dependency Mask (Byte 3- MSB)
19	18	17	Page Sequence Off Dependency Mask (Byte 0 - LSB)
20	19	18	Page Sequence Off Dependency Mask (Byte 1)
21	20	19	Page Sequence Off Dependency Mask (Byte 2)
22	21	20	Page Sequence Off Dependency Mask (Byte 3- MSB)
23	22	21	Fault Slaves Mask (Byte 0 - LSB)
24	23	22	Fault Slaves Mask (Byte 1)
25	24	23	Fault Slaves Mask (Byte 2)
26	25	24	Fault Slaves Mask (Byte 3- LSB)
27	26	25	GPO Sequence On Dependency Mask (Byte 0)
28	27	26	GPO Sequence On Dependency Mask (Byte 1)
29	28	27	GPO Sequence Off Dependency Mask (Byte 0)
30	29	28	GPO Sequence Off Dependency Mask (Byte 1)

The turn on condition for each page can depend on the state of several other pages or input pins. The same pages and pins also may be used to control multiple pages. The GPI and Page dependencies (Sequence On Dependencies) define a set of conditions which allow a page to turn on when met. Note that the logical AND of all conditions must be met before a page can be turned on. Once the page is on, these dependencies have no further effect on the operating status of the page. Specifically, they do not cause a page to turn off.

Sequence On Dependencies work in parallel with the PMBus defined mechanisms used to enable an output. That is, both the Sequence On Dependencies and the PMBus mechanism must be satisfied. For example, if the page responds to an OPERATION command and the OPERATION command specifies OFF, the page remains off even if all of the sequencing Sequence On Dependencies are met. Further, the order in which they are met is irrelevant; if the OPERATION command is issued first, the page waits for the sequencing requirements to be met before it can be turned on; or if the sequencing requirements are met first, the page waits for the OPERATION command before it can be turned on.

In the ON_OFF_CONFIG command, a “None” setting is still subject to the specified Sequence On Dependencies.

Turn on delay (TON_DELAY) is applied after the page has been commanded on and all Sequence On Dependencies are met.

Unless the rail is instructed to turn off immediately, the previous descriptions also apply to the Sequence Off Dependencies – they must be met before the rail is turned off, the turn off delay (TOFF_DELAY) is applied after the dependencies are met, and so forth.

After a fault, the PMBus specification requires that an OFF/ON sequence occurs before the rail is allowed to restart. The toggle of sequencing pin (such as CONTROL pin and Pin-Selected State GPI) is interpreted to meet this requirement. For example, consider a page that responds to the CONTROL pin and was shut down due to a fault. A toggle low then high on the CONTROL pin is sufficient to restart the page.

26.36.1 Enable Pin Configuration

See Section 23 for byte format and details.

Enable pins are configured just like output pins. They are active once conditions dictate that a page must be enabled and inactive until then. The conditions are defined by the ***GPI Dependency Mask*** and the ***Page Dependency Mask*** in this command and the value given by the TON_DELAY command.

Note

The “input” mode is invalid for enable pins. Attempting to set the mode bits to 1 sets Invalid Data.

26.36.2 GPI Sequence On Dependency Mask

Each page has its own GPI Sequence On Dependency Mask, whose bits are defined as follows:

Bit	**31**	**...**	**5**	**4**	**3**	**2**	**1**	**0**
Purpose	GPI 32	...	GPI 6	GPI 5	GPI 4	GPI 3	GPI 2	GPI 1

The enable pin for the page is not asserted until all of the input pins selected by the GPI bits are asserted. Once the enable pin is asserted, the state of these GPI pins has no effect on the state of the enable pin.

Each page can depend on the state of up to 32 input pins. The same pins can be used to control multiple pages.

26.36.3 GPI Sequence Off Dependency Mask

Each page has its own GPI Sequence Off Dependency Mask, whose bits are defined as follows:

Bit	**31**	**...**	**5**	**4**	**3**	**2**	**1**	**0**
Purpose	GPI 32	...	GPI 6	GPI 5	GPI 4	GPI 3	GPI 2	GPI 1

The enable pin for the page is not deasserted until all of the input pins selected by the GPI bits are deasserted. Once the enable pin is deasserted, the state of these GPI pins has no effect on the state of the enable pin.

Each page can depend on the state of up to 32 input pins. The same pins can be used to control multiple pages.

26.36.4 Sequencing Timeout Configuration

The configuration bits in Table 26-39 defines how the device responds to sequence timeouts. Whenever a sequencing timeout occurs, the associated MFR Status and fault log information are updated.

Table 26-39. Sequencing Timeout Configuration Byte

Bits	Name	Description
7:4	Reserved	

Table 26-39. Sequencing Timeout Configuration Byte (continued)

Bits	Name	Description
3:2	Sequence-Off Timeout Action	This bits determine what action to take after a sequence-off timeout occurs: b'00 – The device continues to wait indefinitely for the sequence-off dependencies to be met. b'01 – The device stops waiting for the sequence-off dependencies to be met and continues the process of disabling the rail. b'10 – (same as b'00 action) b'11 – (same as b'00 action)
1:0	Sequence-On Timeout Action	This bits determine what action to take after a sequence-on timeout occurs: b'00 – The device continues to waits indefinitely for the sequence-on dependencies to be met. b'01 – The device stops waiting for the sequence-on dependencies to be met and continues the process of enabling the rail. b'10 – The device resequences this rail and all fault shutdown slaves associated with this rail. This operation happens according to the “Time between Resequences” byte and the “Maximum Resequences” field in the MISC_CONFIG command (see Section 26.42). b'11 – (same as b'00 action)

26.36.5 Sequencing On Timeout

This timeout value is used to check that rail sequences on within a certain period of time. In other words, this timeout is used to detect if one of the rail's sequence-on dependencies is never met. When this occurs, a status bit is set.

The timeout periods start to count when a rail receives a turn-on command as defined in ON_OFF_CONFIG. A timeout value of 0 disables the timeout monitoring function. This byte is formatted according to Section 18.5.

26.36.6 Sequencing Off Timeout

This timeout value is used to check that rail sequences off within a certain period of time. In other words, this timeout is used to detect if one of the rail's sequence-off dependencies is never met. When this occurs, a status bit is set. This byte is formatted according to Section 18.5.

26.36.7 Page Sequence On Dependency Mask

Each page has its own Page Sequence On Dependency Mask, whose bits are defined as follows:

Bit	31	...	13	12	11	10	9	8
Purpose	PAGE31	...	PAGE13	PAGE12	PAGE11	PAGE10	PAGE9	PAGE8

Bit	7	6	5	4	3	2	1	0
Purpose	PAGE7	PAGE6	PAGE5	PAGE4	PAGE3	PAGE2	PAGE1	PAGE0

The enable pin for the page is not asserted until all of the rails selected by these bits have reached their power-good state.

Each page can depend on the state of several other pages. The same pages can be used to control other pages as well.

26.36.8 Page Sequence Off Dependency Mask

Each page has its own Page Sequence Off Dependency Mask, whose bits are defined as follows:

Bit	31	...	13	12	11	10	9	8
Purpose	PAGE31	...	PAGE13	PAGE12	PAGE11	PAGE10	PAGE9	PAGE8

Bit	7	6	5	4	3	2	1	0
Purpose	PAGE7	PAGE6	PAGE5	PAGE4	PAGE3	PAGE2	PAGE1	PAGE0

The enable pin for the page is not deasserted until all of the rails selected by these bits have left their power-good state.

Each page can depend on the state of several other pages. The same pages can be used to control other pages as well.

26.36.9 Fault Slaves Mask

Each page has its own Fault Slaves Mask, whose bits are defined as follows:

Bit	31	...	13	12	11	10	9	8
Purpose	PAGE31 fault slave	...	PAGE13 fault slave	PAGE12 fault slave	PAGE11 fault slave	PAGE10 fault slave	PAGE9 fault slave	PAGE8 fault slave
Bit	7	6	5	4	3	2	1	0
Purpose	PAGE7 fault slave	PAGE6 fault slave	PAGE5 fault slave	PAGE4 fault slave	PAGE3 fault slave	PAGE2 fault slave	PAGE1 fault slave	PAGE0 fault slave

Each page can have multiple fault slave pages. When a fault occurs on the master page, if its response is to shut down, all slave pages are also shut down. If retries are specified for the master page, the slave pages remain running until all retries are exhausted. The slave pages are shut down using sequence off dependencies and TOFF_DELAY. The slave pages do not perform any retries during the fault slave shutdown.

After being shut down, slave rails are latched off as if they had experienced the fault. To re-enable their outputs, an off command must be received before another on command is accepted. A status bit is set in their MFR_STATUS word indicating the reason they are latched off.

26.36.10 GPO Sequence On Dependency Mask

Each page has its own GPO Sequence On Dependency Mask, whose bits are defined in the following table:

Bit	**15**	**...**	**5**	**4**	**3**	**2**	**1**	**0**
Purpose	GPO16	...	GPO6	GPO5	GPO4	GPO3	GPO2	GPO1

It is logical state selected by the GPO not the actual pin output of the GPO used as dependency.

The enable pin for the page is not deasserted until all of the logical states selected by the GPO bits are FALSE. Once the enable pin is deasserted, the logical state of these GPO has no effect on the state of the enable pin.

Each page can depend on the state of up to 16 GPO logical states. The same states can be used to control multiple pages (see Section 26.38).

26.36.11 GPO Sequence Off Dependency Mask

Each page has its own GPO Sequence Off Dependency Mask, whose bits are defined in the following table:

Bit	**15**	**...**	**5**	**4**	**3**	**2**	**1**	**0**
Purpose	GPO16	...	GPO6	GPO5	GPO4	GPO3	GPO2	GPO1

It is logical state selected by the GPO not the actual pin output of the GPO used as dependency.

Each page can depend on the state of up to 16 GPO logical states. The same states can be used to control multiple pages (see Section 26.38).

26.37 (F7h) GPO_CONFIG_INDEX (MFR_SPECIFIC_39)

This R/W Byte command selects the index of the GPO to use for subsequent GPO_CONFIG commands.

Table 26-40. Relationship Between GPO_CONFIG_INDEX and the Actual GPO Number

GPO_CONFIG_INDEX	GPO
0	1

Table 26-40. Relationship Between GPO_CONFIG_INDEX and the Actual GPO Number (continued)

GPO_CONFIG_INDEX	GPO
1	2
. . .	. . .
15	16

26.38 (F8h) GPO_CONFIG (MFR_SPECIFIC_40)

This Read/Write Block command configures the functionality of a given output pin. This paged command allows pins to be configured as status/GPI-influenced outputs. The state of the output pin is determined by a selection of GPIs and statuses of rails, other GPOs (with or without assignment of an actual pin), and other function blocks such as watchdog. The input states are processed through some combinational logic with optional inversion steps. Upon device power up, the GPO status has an initial evaluation according to its configuration. Thereafter, the evaluation is triggered when any of the associated status changes.

Note

All configurations done with the MONITOR_CONFIG, GPI_CONFIG, and SEQ_CONFIG commands must be done before writing this command.

Figure 26-1 provide an overview of how the state of the GPO is determined.

Note

This document refers to the AND paths starting with "AND Path 0".

Note

The information in Figure 26-1 implies that the device supports 12 rails (0 to 11). For a given device, this may not be the case. The interpretation of this information must be adjusted for the correct number of rails.

The effect of State Machine Mode is not represented in Figure 26-1. When that mode is enabled, only one AND patch is active at a time.

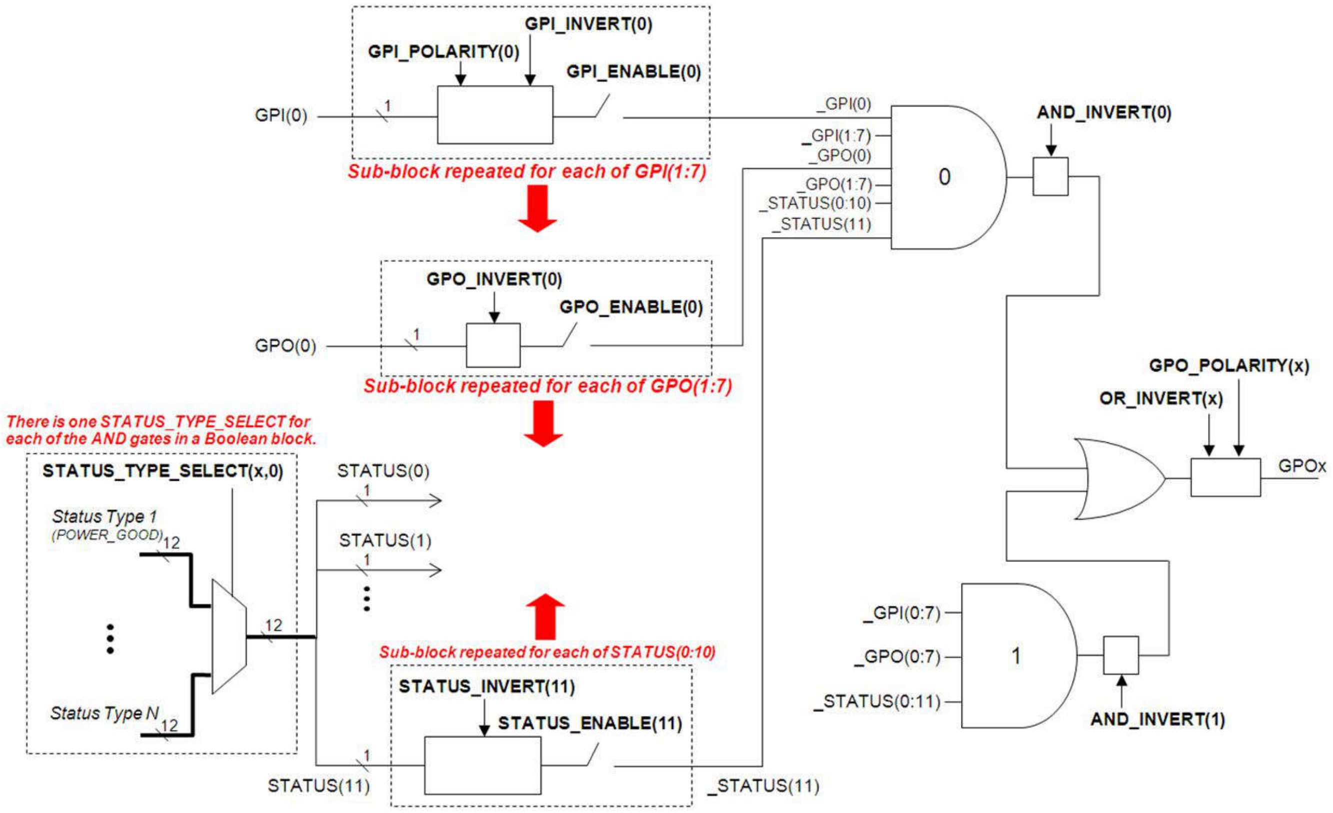


Figure 26-1. Factors Determining the State of a GPO

Note

For the sake of code efficiency, all configured/active GPOs must be packed in the lower GPO indexes. This means, if there is only one configured GPO, it must be associated with GPO index 0. If there are two, they must be associated GPO indexes 0 and 1.

Table 26-41. GPO_CONFIG Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description	
0			CMD = F8	
1	0		BYTE_COUNT = 46	
2	1	0	Output Pin Configuration (Pin ID)	
3	2	1	Output Pin Configuration (Other)	
4	3	2	**Bit**	**Description**
			7	Assert Delay Enable
			6	Deassert Delay Enable
			5	Invert OR Output
			4	Ignore Inputs During Delay
			3:0	High Resolution Delay Count
5	4	3	Millisecond Delay	

Table 26-41. GPO_CONFIG Command Format (continued)

Byte Number (Write)	Byte Number (Read)	Payload Index	Description	
6	5	4	AND Path 0 Configuration	
			Bit	**Description**
			7	Invert AND Output
			6	State Machine Mode Enable
			5:0	Status Type
7	6	5	AND Path 1 Configuration	
			Bit	**Description**
			7	Invert AND Output
			6	reserved
			5:0	Status Type
			AND Path 0	
8	7	6	Status Mask (Byte 0 - LSB)	
9	8	7	Status Mask	
10	9	8	Status Mask	
11	10	9	Status Mask (Byte 3 - MSB)	
12	11	10	Status Inversion Mask (Byte 0 - LSB)	
13	12	11	Status Inversion Mask	
14	13	12	Status Inversion Mask	
15	14	13	Status Inversion Mask (Byte 3 - MSB)	
16	15	14	GPI Mask (Byte 0 - LSB)	
17	16	15	GPI Mask	
18	17	16	GPI Mask	
19	18	17	GPI Mask (Byte 3 - MSB)	
20	19	18	GPI Inversion Mask (Byte 0 - LSB)	
21	20	19	GPI Inversion Mask	
22	21	20	**GPI Inversion Mask**	
23	22	21	GPI Inversion Mask (Byte 3 - MSB)	
24	23	22	GPO Mask (Byte 0 - LSB)	
25	24	23	GPO Mask (Byte 1 - MSB)	
26	25	24	GPO Inversion Mask (Byte 0 - LSB)	
27	26	25	GPO Inversion Mask (Byte 1 - MSB)	
			AND Path 1	
28 -47	27 - 46	26 - 45	Same Configuration Options as AND Path 0 (20 bytes)	

26.38.1 Output Pin Configuration

This byte configures which pin is used for this GPO, its polarity, and if it is actively driven or not. For details, see Input and Output Pin Configuration. A mode of “Input” causes Invalid Data to be set.

26.38.2 Assert Delay Enable

When this bit is set there is a delay (High Resolution or Millisecond) before the GPO is asserted.

26.38.3 Deassert Delay Enable

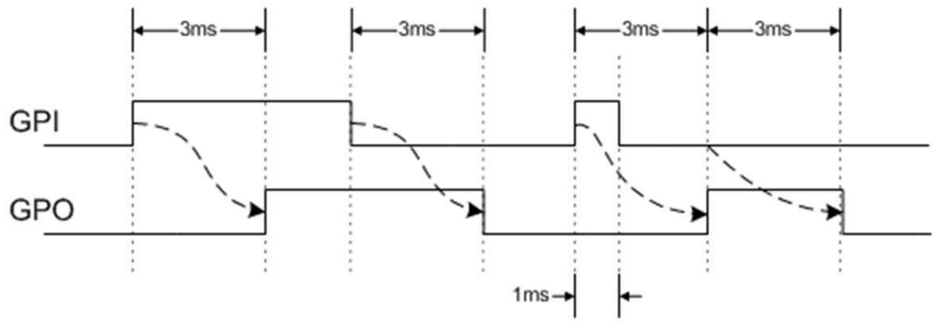
When this bit is set, there is a delay (High Resolution or Millisecond) before the GPO is deasserted.

26.38.4 Invert OR Output

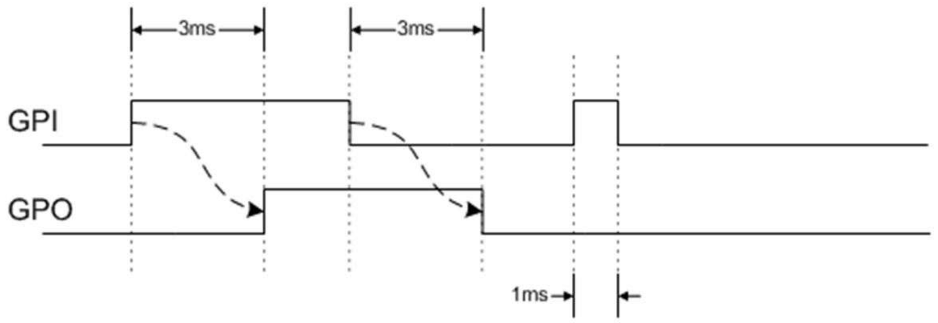
When this bit is set, the result of the OR operation is inverted. As an example, this function can be used to change the OR operation into an AND operation, (a OR b) = (a’ AND b’)’.

26.38.5 Ignore Inputs During Delay

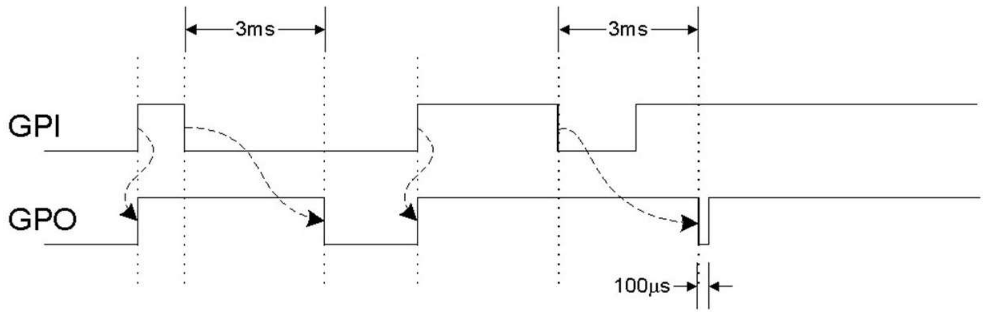
When this bit is set, changes to the inputs affecting the state of the GPO are ignored while the update to the GPO is delayed. In other words, the GPO changes state upon the expiration of the delay time regardless the subsequent input state changes during the delay time. For example, if a GPO was simply following the state of a GPI with a 3 millisecond delay on assertion and deassertion, a timing diagram follows.



If this bit is not set, the GPO state is re-evaluated upon the expiration of the delay time. If the input state returned to the original state at the re-evaluation point, the GPO state does not change. An example timing diagram follows.



When this bit is set, and there is a 3-ms delay on deassertion, but no delay on assertion, an example timing diagram follows. Note that the second deassert-to-assert pulse is only the sampling period of the GPI pins, which happens to be 100 μs.



26.38.6 Invert AND Output

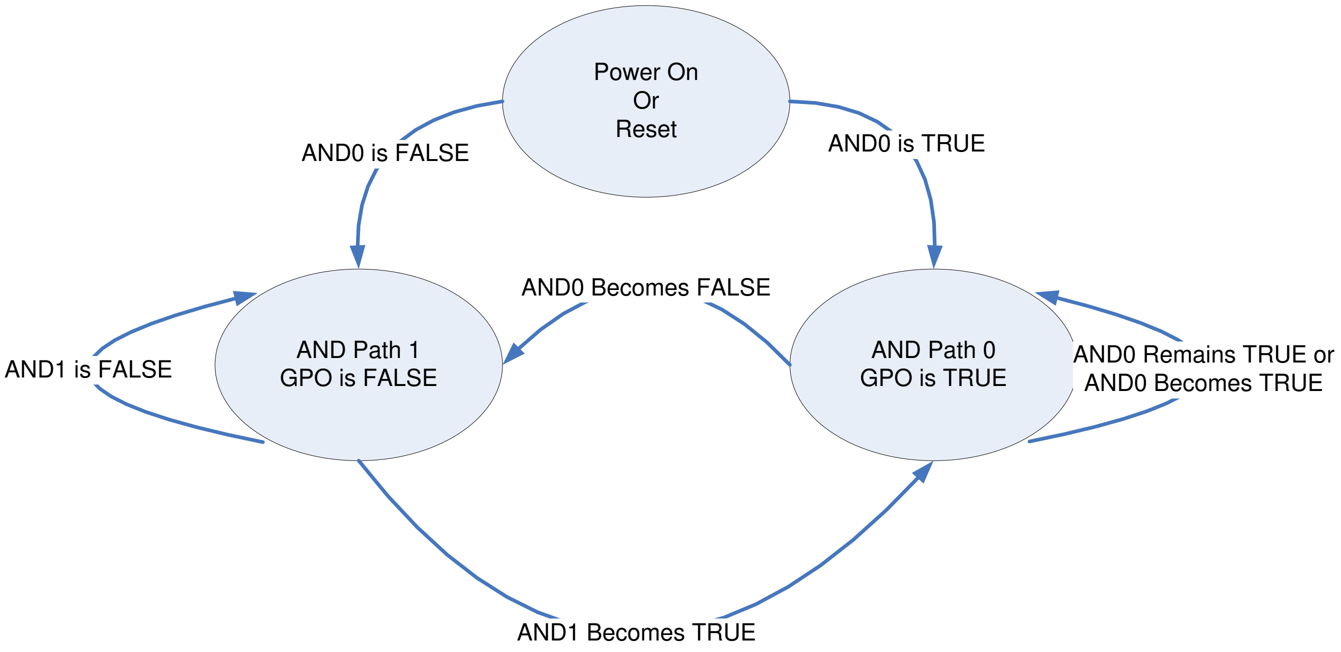
When this bit is set, the associated AND path result is inverted before it is fed into the OR gate. As an example, this function can be used to change the AND operation into an OR operation, (a' AND b')' = (a OR b).

26.38.7 State Machine Mode Enable

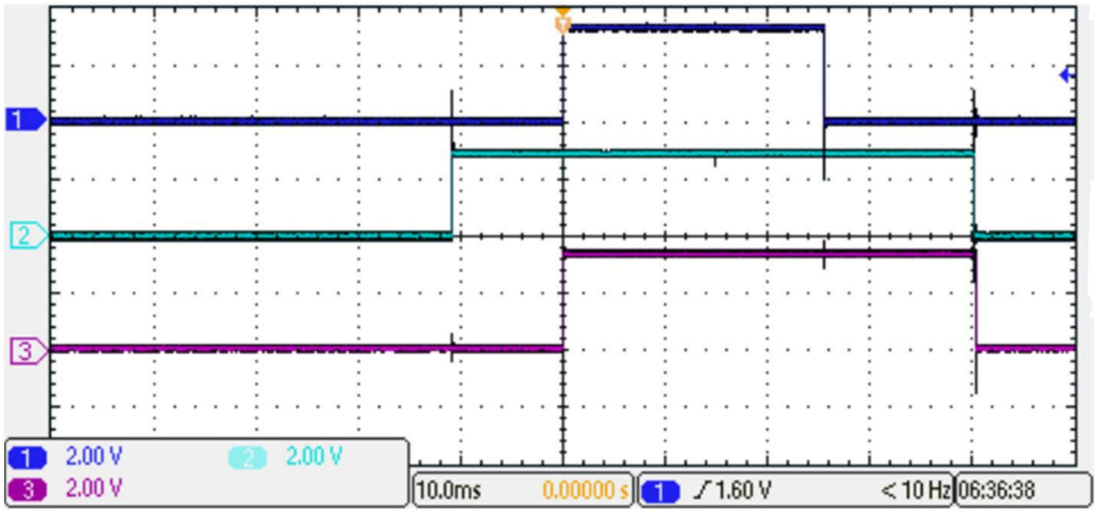
When this bit is set, only one of the AND paths is used at a given time. When the GPO logic result is currently TRUE, AND path 0 is used until the result becomes FALSE. When the GPO logic result is currently FALSE, AND path 1 is used until the result becomes TRUE. This provides a very simple state machine and allows for more complex logical combinations.

Note

The device initially evaluates AND path 0. If it is TRUE, it continues to evaluate AND path 0. If it is FALSE, it begins evaluating AND path 1 in the next evaluation cycle. Evaluation of a GPO is only triggered when its input states change; therefore, the state machine cannot be configured as a self-sustaining oscillator.



For example, to configure a GPO in such a way that it is asserted when two GPI pins are both asserted and stay asserted until both GPIs are deasserted, we need to apply an AND operation when the GPO is deasserted and apply an OR operation when the GPO is asserted. This is shown in the following scope image, where waveform 1 is a GPI, waveform 2 is a GPI, and waveform 3 is a GPO.



This behavior is configured by setting the "State Machine Mode Enable", configuring AND path 0 as (GPI1' AND GPI2')', and configuring AND path 1 as (GPI1 AND GPI2).

26.38.8 High Resolution Delay Count

This value is multiplied by 100 μs to define the high resolution delay.

26.38.9 Millisecond Delay

This value is used to delay the update of a GPO. This byte is formatted according the 8-bit time encoding defined in Section 18.5.

26.38.10 Status Mask

The Status Mask selects which pages are to be used in a given AND path.

Bit	**31**	**...**	**13**	**12**	**11**	**10**	**9**	**8**
Purpose	PAGE31	...	PAGE13	PAGE12	PAGE11	PAGE10	PAGE9	PAGE8

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Purpose	PAGE7	PAGE6	PAGE5	PAGE4	PAGE3	PAGE2	PAGE1	PAGE0

If a bit is set to 1, the corresponding page status is used in the AND path. If a bit is set to 0, the corresponding page status is not used by the AND path.

26.38.11 Status Inversion Mask

The status of each page may be inverted before being used in a given AND path. If a bit is set to 1, the corresponding page status is inverted. Setting the bit to 0 has no effect.

Bit	**31**	**...**	**13**	**12**	**11**	**10**	**9**	**8**
Purpose	PAGE31	...	PAGE13	PAGE12	PAGE11	PAGE10	PAGE9	PAGE8

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Purpose	PAGE7	PAGE6	PAGE5	PAGE4	PAGE3	PAGE2	PAGE1	PAGE0

26.38.12 GPI Mask

This mask determines which GPIs are used for the given AND path. If a bit is set to 1, the corresponding GPI state is used by the AND path. If a bit is set to 0, the corresponding GPI state is not used by the AND path. See the GPI_CONFIG command.

Bit	31	...	5	4	3	2	1	0
Purpose	GPI 31	...	GPI 5	GPI 4	GPI 3	GPI 2	GPI 1	GPI 0

26.38.13 GPI Inversion Mask

This mask determines if the GPI state is inverted before being used for the given AND path. Setting the bit to 1 inverts the corresponding GPI state. Setting the bit to 0 has no effect.

Bit	31	...	5	4	3	2	1	0
Purpose	GPI 31	...	GPI 5	GPI 4	GPI 3	GPI 2	GPI 1	GPI 0

26.38.14 GPO Mask

This mask determines which of the first 8 GPOs are used for a given AND path. If a bit is set to 1, the corresponding GPO state is used by the AND path. If a bit is set to 0, the corresponding GPO state is not used by the AND path.

Note

The pin polarity setting for a GPO does not affect the influence of that GPO on another GPO.

Bit	15	...	5	4	3	2	1	0
Purpose	GPO 15	...	GPO 5	GPO 4	GPO 3	GPO 2	GPO 1	GPO 0

26.38.15 GPO Inversion Mask

This mask determines if the GPO state is inverted before being used for the given AND path. If a bit is set to 1, the corresponding GPO state is inverted. Setting the bit to 0 has no effect.

Bit	15	...	5	4	3	2	1	0
Purpose	GPO 15	...	GPO 5	GPO 4	GPO 3	GPO 2	GPO 1	GPO 0

26.38.16 Status Type Select

These bits select the status type to be applied to the Status Mask. Each AND path can select a different status type.

The _LATCH version of these status flags is latched and can only be cleared by a command (CLEAR_FAULTS) or a GPI (see Section 26.39.3). The version without _LATCH is the actual current state of the condition.

Note

1. The MARGIN_EN and MRG_LOW_nHIGH statuses are updated based on GPI pins (see GPI_CONFIG, see Section 26.39) or the OPERATION command. Whenever the OPERATION command disables margining, the MRG_LOW_nHIGH status is set to true.
2. After the following status types are set, they cannot be cleared until the rail is turned off and then back on: TON_MAX_FAULT, TOFF_MAX_WARN, SEQ_ON_TIMEOUT, SEQ_OFF_TIMEOUT. Note that a resequence operation clears these status types, because it turns the rail off and then turns the rail on.
3. When the _LATCH version of a status flag is cleared, it stays FALSE even if the status is still TRUE. After that, it only becomes TRUE again after the status goes to the FALSE state and back to the TRUE state.

Table 26-42. Status Types

Encoding	Status Type
0	POWER_GOOD
1	MARGIN_EN
2	MRG_LOW_nHIGH
3	VOUT_OV_FAULT
4	VOUT_OV_WARN
5	VOUT_UV_WARN
6	VOUT_UV_FAULT
7	TON_MAX_FAULT
8	TOFF_MAX_WARN
9	IOUT_OC_FAULT
10	IOUT_OC_WARN
11	IOUT_UC_FAULT
12	TEMP_OT_FAULT
13	TEMP_OT_WARN
14	SEQ_ON_TIMEOUT
15	SEQ_OFF_TIMEOUT
16	SYSTEM_WATCHDOG_TIMEOUT
17	VOUT_OV_FAULT_LATCH
18	VOUT_OV_WARN_LATCH
19	VOUT_UV_WARN_LATCH
20	VOUT_UV_FAULT_LATCH
21	TON_MAX_FAULT_LATCH
22	TOFF_MAX_WARN_LATCH
23	IOUT_OC_FAULT_LATCH
24	IOUT_OC_WARN_LATCH
25	IOUT_UC_FAULT_LATCH
26	TEMP_OT_FAULT_LATCH
27	TEMP_OT_WARN_LATCH
28	SEQ_ON_TIMEOUT_LATCH
29	SEQ_OFF_TIMEOUT_LATCH
30	SYSTEM_WATCHDOG_TIMEOUT_LATCH
31	SINGLE_EVENT_UPSET

26.38.17 GPO Configuration Examples

Example 1: GPO = POWER_GOOD(0) *AND* POWER_GOOD(2) *AND* POWER_GOOD(5) *AND* POWER_GOOD(7) *AND* POWER_GOOD(8)

Status Mask 0	=	0x01A3
Status Inversion Mask 0	=	0x0000
GPI Mask 0	=	0x00
Status Type Select 0	=	0
Status Mask 1	=	0x0000
GPI Mask 1	=	0x00
Status Mask 2	=	0x0000
GPI Mask 2	=	0x00
Status Mask 3	=	0x0000
GPI Mask 3	=	0x00

Example 2: GPO = (*NOT* POWER_GOOD[0]) *OR* (*NOT* POWER_GOOD[2]) *OR* (*NOT* POWER_GOOD[5]) *OR* (*NOT* POWER_GOOD[7])

Status Mask 0	=	0x0001
Status Inversion Mask 0	=	0x0001
GPI Mask 0	=	0x00
Status Type Select 0	=	0
Status Mask 1	=	0x0004
Status Inversion Mask 1	=	0x0004
GPI Mask 1	=	0x00
Status Type Select 1	=	0
Status Mask 2	=	0x0020
Status Inversion Mask 2	=	0x0020
GPI Mask 2	=	0x00
Status Type Select 2	=	0
Status Mask 3	=	0x0080
Status Inversion Mask 3	=	0x0080
GPI Mask 3	=	0x00
Status Type Select 3	=	0

Example 3: GPO = ((*NOT* POWER_GOOD[0]) *AND* (*NOT* GPI[3])) *OR* ((*NOT* POWER_GOOD[1]) *AND* (*NOT* GPI[3])) *OR* (VOUT_OV_WARN[2] *AND* (*NOT* GPI[3]))

Status Mask 0	=	0x0001
Status Inversion Mask 0	=	0x0001
GPI Mask 0	=	0x04
GPI Inversion Mask 0	=	0x04
Status Type Select 0	=	0
Status Mask 1	=	0x0002
Status Inversion Mask 1	=	0x0002
GPI Mask 1	=	0x04
GPI Inversion Mask 1	=	0x04
Status Type Select 1	=	0
Status Mask 2	=	0x0004
Status Inversion Mask 2	=	0x0004
GPI Mask 2	=	0x04
GPI Inversion Mask 2	=	0x04
Status Type Select 2	=	3
Status Mask 3	=	0x0000
GPI Mask 3	=	0x00

Example 4: GPO = (GPI[0] *AND* GPI[2]) *OR* (GPI[0] *AND* (*NOT* GPI[4]) *AND* (*NOT* GPI[7]) *OR* GPI[3] *OR* GPI[7]

Status Mask 0	=	0x0000
GPI Mask 0	=	0x05
GPI Inversion Mask 0	=	0x00
Status Mask 1	=	0x0000
GPI Mask 1	=	0x91
GPI Inversion Mask 1	=	0x90
Status Mask 2	=	0x0000
GPI Mask 2	=	0x04

	GPI Inversion Mask 2	=	0x00
	Status Mask 3	=	0x0000
	GPI Mask 3	=	0x80
	GPI Inversion Mask 3	=	0x00

Example 5: GPO = VOUT_OV_WARN[1] *AND* IOUT_OC_WARN[1]

Cannot implement this directly. Apply the relationship (a AND b) = (a' OR b')' = >

GPO = *NOT* ((*NOT* VOUT_OV_WARN[1]) *OR* (*NOT* IOUT_OC_WARN[1]))

Status Mask 0	=	0x0002
Status Inversion Mask 0	=	0x0002
GPI Mask 0	=	0x00
Status Type Select 0	=	4
Status Mask 1	=	0x0002
Status Inversion Mask 1	=	0x0002
GPI Mask 1	=	0x00
Status Type Select 1	=	16
Status Mask 2	=	0x0000
GPI Mask 2	=	0x00
Status Mask 3	=	0x0000
GPI Mask 3	=	0x00

Note UCD90120, UCD90124, and UCD90910): The first NOT, the one that applies to the entire equation, is actually handled by configuring the GPO's polarity backwards.

26.39 (F9h) GPI_CONFIG (MFR_SPECIFIC_41)

This Read/Write Block command configures the functionality for the input pins (GPI).

Table 26-43. GPI_CONFIG Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = F9
1	0		BYTE_COUNT =
2	1	0	GPI_1 Pin Configuration (Pin ID)
3	2	1	GPI_1 Pin COnfiguration [7:3] Reserved [2] Polarity [1:0] Mode
4-5	3-4	2-3	GPI_2 Pin Configuration
6-7	5-6	4-5	GPI_3 Pin Configuration
8-9	7-8	6-7	GPI_4 Pin Configuration
10	9	8	GPI_5 Pin Configuration
12	11	10	GPI_6 Pin Configuration
14	13	12	GPI_7 Pin Configuration
16	15	14	GPI_8 Pin Configuration
18	17	16	GPI_9 Pin Configuration
20	19	18	GPI_10 Pin Configuration
22	21	20	GPI_11 Pin Configuration

Table 26-43. GPI_CONFIG Command Format (continued)

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
24	23	22	GPI_12 Pin Configuration
26	25	24	GPI_13 Pin Configuration
28	27	26	GPI_14 Pin Configuration
30	29	28	GPI_15 Pin Configuration
32	31	30	GPI_16 Pin Configuration
34	33	32	GPI_17 Pin Configuration
36	35	34	GPI_18 Pin Configuration
38	37	36	GPI_19 Pin Configuration
40	39	38	GPI_20 Pin Configuration
42	41	40	GPI_21 Pin Configuration
44	43	42	GPI_22 Pin Configuration
46	45	44	GPI_23 Pin Configuration
48	47	46	GPI_24 Pin Configuration
50	49	48	GPI_25 Pin Configuration
52	51	50	GPI_26 Pin Configuration
54	53	52	GPI_27 Pin Configuration
56	55	54	GPI_28 Pin Configuration
58	57	56	GPI_29 Pin Configuration
60	59	58	GPI_30 Pin Configuration
62	61	60	GPI_31 Pin Configuration
64	63	62	GPI_32 Pin Configuration
66	65	64	Fault Enable Flags – Byte 0 (LSB)
67	66	65	Fault Enable Flags – Byte 1
68	67	66	Fault Enable Flags – Byte 2
69	68	67	Fault Enable Flags – Byte 3 (MSB)
70	69	68	Latched Statuses Clear Pin Selection
71	70	69	“Margin Enable” (MRG_EN) Pin Selection
72	71	70	“Margin Low/Not-High” (MRG_LOW_nHIGH) Pin Selection
73	72	71	Reserved
74	73	72	Debug Mode Pin Selection

26.39.1 GPI Pin Configuration

These bytes configure which pin is used for each GPI and its polarity. For details, see Section 23. A mode other than “Unused” or “Input” causes this command to be rejected, and the device to set Invalid Data.

26.39.2 Fault Enable Flags

When a bit is set, the deassertion of the corresponding GPI will trigger the configured fault response. This fault is noted in the MFR_STATUS command (see Section 26.33) and is logged if configured to do so (see LOGGED_FAULT_DETAIL_ENABLES). When a GPI fault occurs, the PMBALERT# pin is asserted.

GPI_FAULT_RESPONSE(0xF4) works only if the corresponding GPI bit is set here.

Table 26-44. Fault Enable Bits

Bit	31	...	5	4	3	2	1	0
Purpose	GPI 32	...	GPI 6	GPI 5	GPI 4	GPI 3	GPI 2	GPI 1

26.39.3 Latched Statuses Clear Pin Selection

The latched status types (_LATCH) in the GPO_CONFIG command can be cleared by a pin. That Input Pin is selected by this byte (see *GPI Selection (UCD90120 and UCD90124 Only)*). The pin is ignored if the value of this byte is 0. This pin is edge-sensitive.

26.39.4 MRG_EN Pin Selection

Margining can be enabled with the Input Pin selected by this byte (see *GPI Selection (UCD90120 and UCD90124 Only)*). The pin is ignored if the value of this byte is 0.

When this pin is asserted, all rails with margining enabled (see MARGIN_CONFIG, Section 26.35) are put in a margined state (low or high).

Note

If a pin is selected in this byte, another pin must be selected in the MRG_LOW_nHIGH byte as well.

26.39.5 MRG_LOW_nHIGH Pin Selection

The margining level (low or high) can be selected with the Input Pin selected by this byte (see *GPI Selection (UCD90120 and UCD90124 Only)*). The pin is ignored if the value of this byte is 0.

When this pin is asserted, and the MRG_EN pin is asserted, all rails with margining enabled (see *GPI Selection (UCD90120 and UCD90124 Only)*) are put in the margin-low state.

When this pin is not asserted and the MRG_EN pin is asserted, all rails with margining enabled are put in the margin-high state.

26.39.6 Maximum Glitch Time for Fault Pin

This glitch time has the exact same function and format as Maximum Glitch for GPI defined in Section 26.38.11.

26.39.7 GPI Debug Mode Pin Selection

GPI debug Function can be enabled with the input pin identified by this byte. The pin is ignored if the value of the byte is 0. Under Debug Mode, device shall not active PMBus alert pin for any faults/warnings, not response for any fault responses and not log any faults. This function is mainly designed for debug purpose and it is not recommended in the final production.

The following faults and warnings are impacted by impacted by debug mode:

VOUT_OV_FAULT	TON_MAX	IOUT_UC*	SYSTEM_WATCHDOG_TIMEOUT
VOUT_OV_WARNING	TOFF_MAX Warning	OT_FAULT*	RESEQUENCE_ERROR
VOUT_UV_FAULT	IOUT_OC_FAULT*	OT_WARNING*	SLAVE_FAULT
VOUT_UV_WARNING	IOUT_OC_WARNING*	SEQ_ON_TIMEOUT	SEQ_OFF_TIMEOUT
All GPI deasserted			
* These are only available if the devices support temperature or current.			

When the debug mode is on, the rail sequence on/off dependency conditions are ignored; as soon as the sequence on/off timeout is expired, the rails are sequenced on or off accordingly regardless of the timeout action, if the sequence on/off timeout value is set to 0, the rails are sequenced on or off immediately. The fault pin does not pull the fault bus low when debug mode is on. Moreover LGPO affected by these events return to original states when debug mode is on.

26.40 (FAh) GPIO_SELECT (MFR_SPECIFIC_42)

This read/write byte command determines to which GPIO that the GPIO_CONFIG command applies. Refer to Table 23-1 for a list of valid Pin IDs.

26.41 (FBh) GPIO_CONFIG (MFR_SPECIFIC_43)

This Read/Write Byte Command configures the GPIO identified by the GPIO_SELECT command. The Status bit is read-only and gives the current state of the pin. The Out_Enable bit determines if the pin is an output (1 – actively driven) or an input (0 – high impedance). The Out_Value bit determines the state of the pin when it is configured as an output. The Enable bit is a flag indicating whether or not to process this command. When the Enable bit is cleared, this command is ignored. To temporarily change to an output pin state, write this command twice. In the first write, set the Enable bit to 1 so that the changes can be applied. In the second write, set the Enable bit to 0. This does not change the pin state; and because the Enable bit is 0, STORE_DEFAULT_ALL command ignores this command when storing configurations from RAM to flash. This way, the new temporary configuration does not overwrite the default configuration.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Purpose	Reserved	Reserved	Reserved	Reserved	Status	Out_Value	Out_Enable	Enable

This configuration is stored to nonvolatile memory with the STORE_DEFAULT_ALL command. If the pin is configured as an output, it can take a very short time after a reset or power-cycle for the pin to reach the configured state.

Note

Configuring a pin that is also being used by another function (enable and so forth) can likely result in unexpected and unwanted behavior.

26.42 (FCh) MISC_CONFIG (MFR_SPECIFIC_44)

This Read/Write Block Command configures features not covered by other commands.

Table 26-45. MISC_CONFIG Command Format

Byte Number (Write)	Byte Number (Read)	Payload Index	Description
0			CMD = FC
1	0		BYTE_COUNT = 8
2	1	0	Miscellaneous Configuration Byte
3	2	1	Time between Re-Sequences
4	3	2	External Reference Voltage(low byte)
5	4	3	External Reference Voltage(high byte)
6	5	4	Resequence Rail Mask (Byte 0 - LSB)
7	6	5	Resequence Rail Mask (Byte 1)
8	7	6	Resequence Rail Mask (Byte 2)
9	8	7	Resequence Rail Mask (Byte 3- MSB)

26.42.1 Miscellaneous Configuration Byte

The bit definitions for the Miscellaneous Configuration Byte are shown in Table 26-46.

Table 26-46. Miscellaneous Configuration Byte

Bits	Name	Description
7	Resequence Continuously	When this bit is set, there is no limit to the number of times that the device attempts to resequence. The “Max Resequences” value does not apply.
6	Resequence Abort	If a TOFF_MAX_WARN warning occurs during resequencing, stop the resequencing operation
5:4	Max Re-Sequences	The maximum number of times to attempt to re-sequence. b'00 – 1 time b'01 – 2 times b'10 – 3 times b'11 – 4 times See Section 26.25.2 for more information.
3	Slave	When this bit is set, the device is a slave to take external sync clock. This bit is only valid if it is under multi-chip user case.
2	Enable Log FIFO	When this bit is set, all or part of the LOGGED_FAULT_DETAIL is treated as a FIFO, depending on the "FIFO Entire Log" bit.
1	External ADC Reference	When this bit is set, the external ADC reference (2.4v-3.0v) is used for ADC. A device reset is required after this bit is changed.
0	External Crystal Enable	When this bit is set and stored, the UCD91xxx device will attempt to use the Low Frequency (32.768 kHz) crystal on next restart.

26.42.2 Time Between Resequences

This byte is formatted according to Section 18.5.

26.42.3 External Reference Voltage

This field defines the external ADC reference voltage and it follows LINEAR16 format defined in Section 18.2. The reference voltage must be between 2.4 V and 3 V with 0.01-V resolution.

Note

If the external reference voltage is to be enabled, it should be done before the MONITOR_CONFIG command is written, otherwise the monitoring channels will default to using the internal reference voltage.

26.42.4 Resequence_rails_mask

The page mask is made up of four bytes whose bits are defined as follows:

Bit	31	...	5	4	3	2	1	0
Purpose	PAGE31	...	PAGE5	PAGE4	PAGE3	PAGE2	PAGE1	PAGE0

Note

This table assumes that the device supports 32 rails.

Each bit selects which pages are not checked for POWER_GOOD_OFF and TOFF_WARN status when performing re-sequence. When the corresponding page bit is set to 1, the re-sequence engine does not check its POWER_GOOD_OFF and TOFF_WARN status. When the corresponding page bit is set to 0, the resequence engine checks its POWER_GOOD_OFF and TOFF_WARN status.

26.43 (FDh) DEVICE_ID (MFR_SPECIFIC_45)

This Read-only Block Read command returns an ASCII string up to 32 characters in length. It is divided into three or four sections, separated by the vertical bar character (‘|’). The format within each section can change in future releases; therefore, support tools must not rely on specific byte alignment. Instead they must identify the sections and sub-sections using the vertical bar and the periods that separate them.

1. The first section is the hardware device ID (for example, ‘UCD91xxx ”).
2. The second section contains the firmware version information.

   Its format is “A.BB.C.DDDD”, where:

   A = Major Release Level (1 character)
   BB = Minor Release Level (2 characters)
   C = Sub-Release (1 character)
   DDDD = Build Number (4 characters)

   The major and minor release numbers are incremented immediately after each official firmware release.

   The sub-release field allows for branching off the main development path to build updates based on older versions.

   The build number is automatically updated every time firmware is compiled. The value does not reset to 0 when the release level is updated. Several prerelease versions of firmware can have the same major, minor, and sub-release numbers. These different prerelease versions can be distinguished by the build number.
3. The third section contains the firmware compilation date.
   The date is reported in YYMMDD format, similar to the MFR_DATE command.
4. The optional fourth section can contain device-specific information.

Example: A typical DEVICE_ID string is “UCD91xxx|2.04.0.0069|070509”. In this example,

Hardware device	= UCD91xxx
Firmware Major Release	= 2
Firmware Minor Release	= 04
Firmware Sub-Release	= 0
Firmware Build Number	= 69
Firmware Build Date	= May 9, 2007

27 Range Checking and Limits

Table 27-1 shows the numerical limits for all the supported PMBus commands.

Table 27-1. Range and Limits for PMBus Commands

Code (hex)	Command	Minimum	Maximum	Hardcoded Default	Comments
0	PAGE	0	13 or 255	0	
1	OPERATION	See comments	See comments	0	The seven meaningful values for this parameter are 0x00, 0x40, 0x80, 0x94, 0x98, 0x98, 0xA4 and 0xA8.
2	ON_OFF_CONFIG	See comments	See comments	0	>0x20 invalid, all others accepted
3	CLEAR_FAULTS	n/a	n/a	n/a	Write Only
4	PHASE				Not Supported
05-0F	Reserved				
10	WRITE_PROTECT				Not Supported
11	STORE_DEFAULT_ALL	n/a	n/a	n/a	Write Only
12	RESTORE_DEFAULT_ALL	n/a	n/a	n/a	Write Only
13	STORE_DEFAULT_CODE				Not Supported
14	RESTORE_DEFAULT_CODE				Not Supported
15	STORE_USER_ALL				Not Supported
16	RESTORE_USER_ALL				Not Supported
17	STORE_USER_CODE				Not Supported
18	RESTORE_USER_CODE				Not Supported
19	CAPABILITY	n/a	n/a	0xB0	Read Only
1A	QUERY				Not Supported
1B-1F	Reserved				
20	VOUT_MODE	–16	15	0	Five-bit, two's complement exponent
21	VOUT_COMMAND	0	See comment	0	Depends on VOUT_MODE
22	VOUT_TRIM				Not Supported
23	VOUT_CAL_OFFSET				Not Supported
24	VOUT_MAX				Not Supported
25	VOUT_MARGIN_HIGH	0	See comment	0	Depends on VOUT_MODE
26	VOUT_MARGIN_LOW	0	See comment	0	Depends on VOUT_MODE
27	VOUT_TRANSITION_RATE				Not Supported
28	VOUT_DROOP				Not Supported
29	VOUT_SCALE_LOOP				Not Supported
2A	VOUT_SCALE_MONITOR	0	See comment	0	Depends on VOUT_MODE
2B-2F	Reserved				
30	COEFFICIENTS				Not Supported
31	POUT_MAX				Not Supported
32	MAX_DUTY				Not Supported
33	FREQUENCY_SWITCH				Not Supported
34	Reserved				
35	VIN_ON				Not Supported
36	VIN_OFF				Not Supported
37	INTERLEAVE				Not Supported
38	IOUT_CAL_GAIN	0.6113	20000	0	A number from 20000 to 40031 results in 20000 because of internal resolution.
39	IOUT_CAL_OFFSET	–511.5	511.5	0	
3A	FAN_CONFIG_1_2				Not Supported
3B	FAN_COMMAND_1				Not Supported
3C	FAN_COMMAND_2				Not Supported
3D	FAN_CONFIG_3_4				Not Supported
3E	FAN_COMMAND_3				Not Supported
3F	FAN_COMMAND_4				Not Supported

Table 27-1. Range and Limits for PMBus Commands (continued)

Code (hex)	Command	Minimum	Maximum	Hardcoded Default	Comments
40	VOUT_OV_FAULT_LIMIT	0	See comment	0	Depends on VOUT_MODE
41	VOUT_OV_FAULT_RESPONSE				See FAULT_RESPONSES command
42	VOUT_OV_WARN_LIMIT	0	See comment	0	Depends on VOUT_MODE
43	VOUT_UV_WARN_LIMIT	0	See comment	0	Depends on VOUT_MODE
44	VOUT_UV_FAULT_LIMIT	0	See comment	0	Depends on VOUT_MODE
45	VOUT_UV_FAULT_RESPONSE				See FAULT_RESPONSES command
46	IOUT_OC_FAULT_LIMIT	–511.5	511.5	0	
47	IOUT_OC_FAULT_RESPONSE				See FAULT_RESPONSES command
48	IOUT_OC_LV_FAULT_LIMIT				Not Supported
49	IOUT_OC_LV_FAULT_RESPONSE				Not Supported
4A	IOUT_OC_WARN_LIMIT	–511.5	511.5	0	
4B	IOUT_UC_FAULT_LIMIT	–511.5	511.5	0	
4C	IOUT_UC_FAULT_RESPONSE				See FAULT_RESPONSES command
4D	Reserved				
4E	Reserved				
4F	OT_FAULT_LIMIT	–255.75	255.75	0	
50	OT_FAULT_RESPONSE				See FAULT_RESPONSES command
51	OT_WARN_LIMIT	–255.75	255.75	0	
52	UT_WARN_LIMIT				Not Supported
53	UT_FAULT_LIMIT				Not Supported
54	UT_FAULT_RESPONSE				Not Supported
55	VIN_OV_FAULT_LIMIT				Not Supported
56	VIN_OV_FAULT_RESPONSE				Not Supported
57	VIN_OV_WARN_LIMIT				Not Supported
58	VIN_UV_WARN_LIMIT				Not Supported
59	VIN_UV_FAULT_LIMIT				Not Supported
5A	VIN_UV_FAULT_RESPONSE				Not Supported
5B	IIN_OC_FAULT_LIMIT				Not Supported
5C	IIN_OC_FAULT_RESPONSE				Not Supported
5D	IIN_OC_WARN_LIMIT				Not Supported
5E	POWER_GOOD_ON	0	See comment	0	Depends on VOUT_MODE
5F	POWER_GOOD_OFF	0	See comment	0	Depends on VOUT_MODE
60	TON_DELAY	0	3276	0	
61	TON_RISE				Not Supported
62	TON_MAX_FAULT_LIMIT	0	3276	0	
63	TON_MAX_FAULT_RESPONSE				See FAULT_RESPONSES command
64	TOFF_DELAY	0	3276	0	
65	TOFF_FALL				Not Supported
66	TOFF_MAX_WARN_LIMIT	0	3276 or 0x7FFF	0	0x7FFF is a special value meaning there is no limit. See section 16.7 of the PMBus Specification.
67	Reserved				
68	POUT_OP_FAULT_LIMIT				Not Supported
69	POUT_OP_FAULT_RESPONSE				Not Supported
6A	POUT_OP_WARN_LIMIT				Not Supported
6B	PIN_OP_WARN_LIMIT				Not Supported
6C-77	Reserved				
78	STATUS_BYTE				Read Only
79	STATUS_WORD				Read Only
7A	STATUS_VOUT				Read Only
7B	STATUS_IOUT				Read Only
7C	STATUS_INPUT				Not Supported

Table 27-1. Range and Limits for PMBus Commands (continued)

Code (hex)	Command	Minimum	Maximum	Hardcoded Default	Comments
7D	STATUS_TEMPERATURE				Read Only
7E	STATUS_CML				Read Only
7F	STATUS_OTHER				Not Supported
80	STATUS_MFR_SPECIFIC				Not Supported
81	STATUS_FANS_1_2				Not Supported
82	STATUS_FANS_3_4				Not Supported
83-87	Reserved				
88	READ_VIN				Not Supported
89	READ_IIN				Not Supported
8A	READ_VCAP				Not Supported
8B	READ_VOUT				Read Only
8C	READ_IOUT				Read Only
8D	READ_TEMPERATURE_1				Read Only
8E	READ_TEMPERATURE_2				Read Only
8F	READ_TEMPERATURE_3				Not Supported
90	READ_FAN_SPEED_1				Not Supported
91	READ_FAN_SPEED_2				Not Supported
92	READ_FAN_SPEED_3				Not Supported
93	READ_FAN_SPEED_4				Not Supported
94	READ_DUTY_CYCLE				Not Supported
95	READ_FREQUENCY				Not Supported
96	READ_POUT				Not Supported
97	READ_PIN				Not Supported
98	PMBUS_REVISION				Read Only
99	MFR_ID	n/a	n/a	See comment	The default is an empty string, all zeros
9A	MFR_MODEL	n/a	n/a	See comment	The default is an empty string, all zeros
9B	MFR_REVISION	n/a	n/a	See comment	The default is an empty string, all zeros
9C	MFR_LOCATION	n/a	n/a	See comment	The default is an empty string, all zeros
9D	MFR_DATE	n/a	n/a	See comment	The default is an empty string, all zeros
9E	MFR_SERIAL	n/a	n/a	See comment	The default is an empty string, all zeros
9F	Reserved				
A0	MFR_VIN_MIN				Not Supported
A1	MFR_VIN_MAX				Not Supported
A2	MFR_IIN_MAX				Not Supported
A3	MFR_PIN_MAX				Not Supported
A4	MFR_VOUT_MIN				Not Supported
A5	MFR_VOUT_MAX				Not Supported
A6	MFR_IOUT_MAX				Not Supported
A7	MFR_POUT_MAX				Not Supported
A8	MFR_TAMBIENT_MAX				Not Supported
A9	MFR_TAMBIENT_MIN				Not Supported
AA-AF	Reserved				
B0-BF	USER_DATA_00 -USER_DATA_15				Not Supported
C0-CF	Reserved				
D0	SEQ_TIMEOUT(MFR_SPECIFIC_00)	0	3276	0	
D1	VOUT_CAL_MONITOR (MFR_SPECIFIC_01)	See comment	See comment	0	Depends on VOUT_MODE (Note this parameter is treated as a SIGNED variable)

Table 27-1. Range and Limits for PMBus Commands (continued)

Code (hex)	Command	Minimum	Maximum	Hardcoded Default	Comments
D2	SYSTEM_RESET_CONFIG (MFR_SPECIFIC_02)	n/a	n/a	0	
D3	SYSTEM_WATCHDOG_CONFIG (MFR_SPECIFIC_03)	n/a	n/a	0	
D4	SYSTEM_WATCHDOG_RESET (MFR_SPECIFIC_04)	n/a	n/a	0	
D5	MONITOR_CONFIG (MFR_SPECIFIC_05)	n/a	n/a	0	
D6	NUM_PAGES (MFR_SPECIFIC_06)	0	Device dependent	0	Read Only
D7	RUN_TIME_CLOCK (MFR_SPECIFIC_07)	n/a	n/a	0	
D8	RUN_TIME_CLOCK_TRIM (MFR_SPECIFIC_08)	n/a	n/a	0	
D9	ROM_MODE (MFR_SPECIFIC_09)	n/a	n/a	n/a	Write Only
DA	USER_RAM_00 (MFR_SPECIFIC_10)	0	255	0	
DB	SOFT_RESET (MFR_SPECIFIC_11)	n/a	n/a	n/a	Write Only
DC	RESET_COUNT (MFR_SPECIFIC_12)	0	65535	0	
DD	PIN_SELECTED_RAIL_STATES (MFR_SPECIFIC_13)	n/a	n/a	0	
DE	RESEQUENCE (MFR_SPECIFIC_14)	0	0xFFFF	n/a	Write Only
DF	CONSTANTS (MFR_SPECIFIC_15)	n/a	n/a	n/a	Read Only
E0	PWM_SELECT (MFR_SPECIFIC_16)	0	12	0	
E1	PWM_CONFIG (MFR_SPECIFIC_17)	n/a	n/a	0	
E2	PARM_INFO (MFR_SPECIFIC_18)	n/a	n/a	0	Index is checked to verify that it points to a valid base address
E3	PARM_VALUE (MFR_SPECIFIC_19)	n/a	n/a	0	
E4	TEMPERATURE_CAL_GAIN (MFR_SPECIFIC_20)	–1638	1638	0	
E5	TEMPERATURE_CAL_OFFSET (MFR_SPECIFIC_21)	–255.75	255.75	0	
E6	SET_BREAKPOINTS (MFR_SPECIFIC_22)	n/a	n/a	0	
E7	DEBUG_CONTINUE (MFR_SPECIFIC_23)	n/a	n/a	0	
E8	(MFR_SPECIFIC_24)	n/a	n/a	0	
E9	FAULT_RESPONSES (MFR_SPECIFIC_25)	n/a	n/a	0	
EA	LOGGED_FAULTS (MFR_SPECIFIC_26)	n/a	n/a	n/a	Only valid write is all zeroes.
EB	LOGGED_FAULT_DETAIL_INDEX (MFR_SPECIFIC_27)	0	Device dependent	0	
EC	LOGGED_FAULT_DETAIL (MFR_SPECIFIC_28)	n/a	n/a	0	Read Only
ED	LOGGED_PAGE_PEAKS (MFR_SPECIFIC_29)	n/a	n/a	0	Only valid write is all zeroes.
EE	LOGGED_COMMON_PEAKS (MFR_SPECIFIC_30)	n/a	n/a	0	Only valid write is all zeroes.
EF	LOGGED_FAULT_DETAIL_ENABLES (MFR_SPECIFIC_31)	n/a	n/a	See comment	All logging is enabled by default
F0	EXECUTE_FLASH (MFR_SPECIFIC_32)	n/a	n/a	0	Write Only
F3	MFR_STATUS (MFR_SPECIFIC_35)	n/a	n/a	0	
F4	GPI_FAULT_RESPONSES (MFR_SPECIFIC_36)	n/a	n/a	0	
F5	MARGIN_CONFIG (MFR_SPECIFIC_37)	n/a	n/a	0	
F6	SEQ_CONFIG (MFR_SPECIFIC_38)	n/a	n/a	0	
F7	GPO_CONFIG_INDEX (MFR_SPECIFIC_39)	0	12	0	
F8	GPO_CONFIG (MFR_SPECIFIC_40)	n/a	n/a	0	

Table 27-1. Range and Limits for PMBus Commands (continued)

Code (hex)	Command	Minimum	Maximum	Hardcoded Default	Comments
F9	GPI_CONFIG (MFR_SPECIFIC_41)	n/a	n/a	0	
FA	GPIO_SELECT (MFR_SPECIFIC_42)	0	n/a	0	
FB	GPIO_CONFIG (MFR_SPECIFIC_43)	n/a	n/a	0	
FC	MISC_CONFIG (MFR_SPECIFIC_44)	n/a	n/a	0	
FD	DEVICE_ID (MFR_SPECIFIC_45)	n/a	n/a	Device dependent	
FE	Mfr_Specific_Extended_Command				Not Supported
FF	PMBUS_Extended_Command				Not Supported

28 Glossary

ACK:	Acknowledge – Indicates that the PMBus has received the message correctly.
ADC:	Analog-to-digital converter – Converts analog voltages to digital counts that can be used for monitoring or control.
DAC:	Digital-to-analog converter
DFlash:	Data Flash memory – Nonvolatile memory used for storing PMBus settings. The values in DFlash are automatically copied to RAM during wake up.
FPWM:	Fast pulse width modulation pin. These pins are capable of a higher frequency than the other PWM pins.
GPI:	General-purpose input
GPIO:	General-purpose input/output
GPO:	General-purpose output
NACK:	Non-acknowledge – An error has occurred in the PMBus message transfer.
PFlash:	Program Flash memory – Nonvolatile memory used for the UCD91xxx main firmware.
PMBus:	Power Management Bus – An open-standard protocol that defines a means of communicating with power conversion devices using an I^2C physical interface.
PWM:	Pulse width modulation or pulse width modulator
RAM:	Random access memory – Volatile memory used to hold PMBus settings and internal variables. PMBus settings are lost after a reset unless they are stored to Data Flash.
ROM:	Read-only memory – Nonvolatile memory used for the UCD91xxx boot algorithms and some common data tables.

29 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

DATE	REVISION	NOTES
December 2025	*	Initial Release