Figure 8-6 and Figure 8-7 show the write and read operation timing diagrams, respectively. Note that register bytes are sent most significant byte first, followed by the least significant byte.

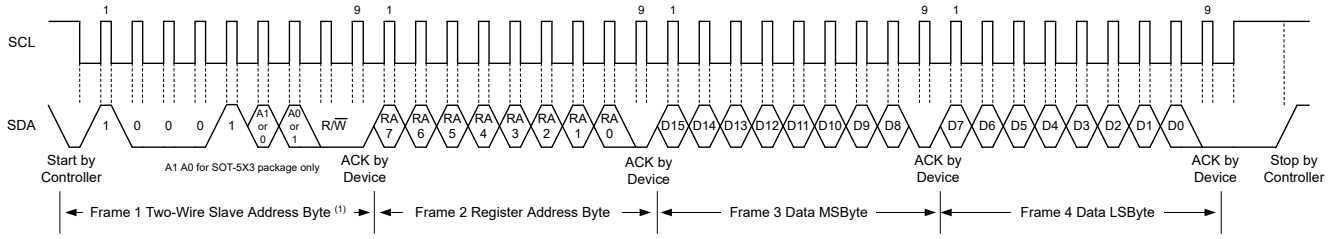
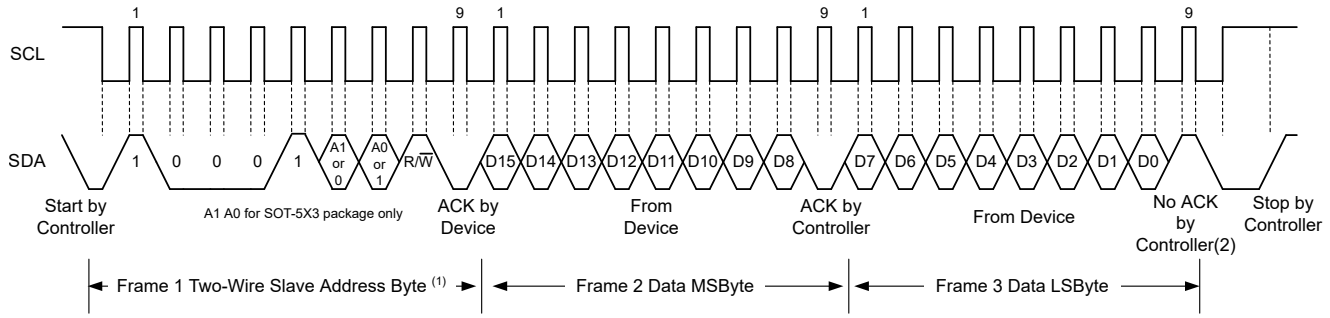



**Figure 8-6. I²C Write Example**




A. An ACK by the controller can also be sent.

**Figure 8-7. I²C Read Example**

#### 8.5.2.1 High-Speed I²C Mode

When the bus is idle, both the SDA and SCL lines are pulled high by the pullup resistors or active pullup devices. The controller generates a start condition followed by a valid serial byte containing the high-speed (HS) controller code 0000 1XXXb. This transmission is made in either standard mode or fast mode (up to 400 kHz). The device does not acknowledge the HS controller code but does recognize the code and switches its internal filters to support a 2.6-MHz operation.

The controller then generates a repeated start condition (a repeated start condition has the same timing as the start condition). After this repeated start condition, the protocol is the same as F/S mode, except that transmission speeds up to 2.6 MHz are allowed. Instead of using a stop condition, use repeated start conditions to secure the bus in HS mode. A stop condition ends the HS mode and switches all internal filters of the ddevice to support the F/S mode.

#### 8.5.2.2 Burst Read Mode

OPT4001 supports I²C burst read mode which helps in minimizing the number of transactions on the bus for efficient data transfer from the device to the controller.

Before considering the burst mode, a regular I²C read transaction involves an I²C write operation to the device read pointer, followed by the actual I²C read operation. If the output registers and FIFO registers which are in continuous locations, are writing the register pointer every 2 bytes, this takes up several clock cycles. With the burst mode enabled, the read pointer address is auto incremented after every register read (2 bytes), eliminating the need write operations to set the pointer for subsequent register reads.

Burst mode can be enabled by setting the register I2C_BURST. When a STOP command is issued the pointer resets to the original register address before the auto-increments.

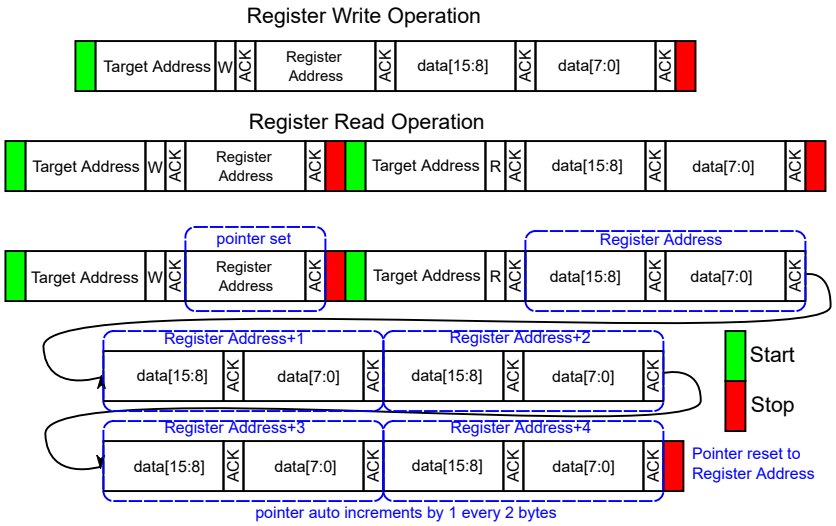



**Figure 8-8. I²C Operations**

#### 8.5.2.3 General-Call Reset Command

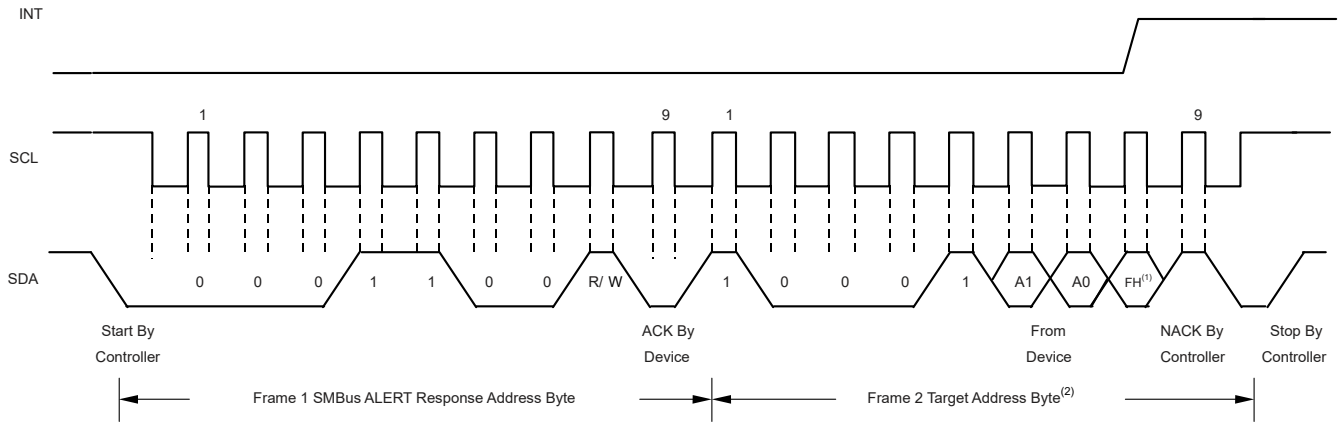
The I²C general-call reset allows the host controller in one command to reset all devices on the bus that respond to the general-call reset command. The general call is initiated by writing to the I²C address 0 (0000 0000b). The reset command is initiated when the subsequent second address byte is 06h (0000 0110b). With this transaction, the device issues an acknowledge bit and sets all registers to the power-on-reset default condition.

#### 8.5.2.4 SMBus Alert Response

The SMBus alert response provides a quick identification for which device issued the interrupt. Without this alert response capability, the processor does not know which device pulled the interrupt line when there are multiple target devices connected.

OPT4001 is designed to respond to the SMBus alert response address, when in the latched window-style comparison mode. The OPT4001 does not respond to the SMBus alert response when in transparent mode.

The response behavior of the device to the SMBus alert response is shown in Figure 8-9. When the interrupt line to the processor is pulled to active, the controller can broadcast the alert response target address. Following this alert response, any target devices that generated an alert identify themselves by acknowledging the alert response and sending respective I²C address on the bus. The alert response can activate several different target devices simultaneously. If more than one target attempts to respond, bus arbitration rules apply. The device with the lowest address wins the arbitration. If the OPT4001 loses the arbitration, the device does not acknowledge the I²C transaction and the INT pin remains in an active state, prompting the I²C controller processor to issue a subsequent SMBus alert response. When the OPT4001 wins the arbitration, the device acknowledges the transaction and sets the INT pin to inactive. The controller can issue that same command again, as many times as necessary to clear the INT pin. See Section 8.4.2 for additional details of how the flags and INT pin are controlled. The controller can obtain information about the source of the OPT4001 interrupt from the address broadcast in the above process. The FLAG_H value is sent as the final LSB of the address to provide the controller additional information about the cause of the OPT4001 interrupt. If the controller requires additional information, the result register or the configuration register can be queried. The FLAG_H and FLAG_L fields are not cleared upon an SMBus alert response.




A. FH is the FLAG_H register

B. A1 and A0 are determined by the ADDR pin (only on SOT-5X3 version)

**Figure 8-9. Timing Diagram for SMBus Alert Response**

## 8.6 Register Maps

**Figure 8-10. ALL Register Map**

| ADD | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00h | EXPONENT | | | | RESULT_MSB | | | | | | | | | | | |
| 01h | RESULT_LSB | | | | | | | | COUNTER | | | | CRC | | | |
| 02h | EXPONENT_FIFO0 | | | | RESULT_MSB_FIFO0 | | | | | | | | | | | |
| 03h | RESULT_LSB_FIFO0 | | | | | | | | COUNTER_FIFO0 | | | | CRC_FIFO0 | | | |
| 04h | EXPONENT_FIFO1 | | | | RESULT_MSB_FIFO1 | | | | | | | | | | | |
| 05h | RESULT_LSB_FIFO1 | | | | | | | | COUNTER_FIFO1 | | | | CRC_FIFO1 | | | |
| 06h | EXPONENT_FIFO2 | | | | RESULT_MSB_FIFO2 | | | | | | | | | | | |
| 07h | RESULT_LSB_FIFO2 | | | | | | | | COUNTER_FIFO2 | | | | CRC_FIFO2 | | | |
| 08h | THRESHOLD_L_EXPONENT | | | | THRESHOLD_L_RESULT | | | | | | | | | | | |
| 09h | THRESHOLD_H_EXPONENT | | | | THRESHOLD_H_RESULT | | | | | | | | | | | |
| 0Ah | QWAKE | 0 | RANGE | | | | CONVERSION_TIME | | | | OPERATING_MODE | | LATCH | INT_POL | FAULT_COUNT | |
| 0Bh | 1024 | | | | | | | | | | | INT_DIR | INT_CFG | | 0 | I2C_BURST |
| 0Ch | 0 | | | | | | | | | | | | OVERLOAD_FLAG | CONVERSION_READY_FLAG | FLAG_H | FLAG_L |
| 11h | 0 | | DIDL | | DIDH | | | | | | | | | | | |

### 8.6.1 ALL Register Map

#### 8.6.1.1 Register 0h (offset = 0h) [reset = 0h]

**Figure 8-11. Register 0h**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| EXPONENT | | | | RESULT_MSB | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT_MSB | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; W = Write only; -n = value after reset

**Figure 8-12. Register 00 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-12 | EXPONENT | R | 0h | EXPONENT output. Determines the full-scale range of the light measurement. Used as a scaling factor for lux calculation |
| 11-0 | RESULT_MSB | R | 0h | Result register MSB (Most significant bits). Used to calculate the MANTISSA representing light level within a given EXPONENT or full-scale range |

#### 8.6.1.2 Register 1h (offset = 1h) [reset = 0h]

**Figure 8-13. Register 1h**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT_LSB | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COUNTER | | | | CRC | | | |
| R-0h | | | | R-0h | | | |

LEGEND: R/W = Read/Write; W = Write only; -n = value after reset

**Figure 8-14. Register 01 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | RESULT_LSB | R | 0h | Result register LSB(Least significant bits). Used to calculate MANTISSA representing light level within a given EXPONENT or full-scale range |
| 7-4 | COUNTER | R | 0h | Sample counter. Rolling counter which increments for every conversion |
| 3-0 | CRC | R | 0h | CRC bits.<br>R[19:0]=MANTISSA=((RESULT_MSB<<8)+ RESULT_LSB<br>X[0]=XOR(E[3:0],R[19:0],C[3:0]) XOR of all bits<br>X[1]=XOR(C[1],C[3],R[1],R[3],R[5],R[7],R[9],R[11],R[13],R[15],R[17],R[19],E[1],E[3])<br>X[2]=XOR(C[3],R[3],R[7],R[11],R[15],R[19],E[3])<br>X[3]=XOR(R[3],R[11],R[19]) |

#### 8.6.1.3 Register 2h (offset = 2h) [reset = 0h]

**Figure 8-15. Register 2h**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|

**Figure 8-15. Register 2h (continued)**

| EXPONENT_FIFO0 | | | | RESULT_MSB_FIFO0 | | | |
|---|---|---|---|---|---|---|---|
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT_MSB_FIFO0 | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; W = Write only; -n = value after reset

**Figure 8-16. Register 02 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-12 | EXPONENT_FIFO0 | R | 0h | EXPONENT register from FIFO 0 |
| 11-0 | RESULT_MSB_FIFO0 | R | 0h | RESULT_MSB Register from FIFO 0 |

### 8.6.1.4 Register 3h (offset = 3h) [reset = 0h]

**Figure 8-17. Register 3h**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT_LSB_FIFO0 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COUNTER_FIFO0 | | | | CRC_FIFO0 | | | |
| R-0h | | | | R-0h | | | |

LEGEND: R/W = Read/Write; W = Write only; -n = value after reset

**Figure 8-18. Register 03 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | RESULT_LSB_FIFO0 | R | 0h | RESULT_LSB Register from FIFO 0 |
| 7-4 | COUNTER_FIFO0 | R | 0h | COUNTER Register from FIFO 0 |
| 3-0 | CRC_FIFO0 | R | 0h | CRC Register from FIFO 0 |

### 8.6.1.5 Register 4h (offset = 4h) [reset = 0h]

**Figure 8-19. Register 4h**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| EXPONENT_FIFO1 | | | | RESULT_MSB_FIFO1 | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT_MSB_FIFO1 | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; W = Write only; -n = value after reset

**Figure 8-20. Register 04 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-12 | EXPONENT_FIFO1 | R | 0h | EXPONENT register from FIFO 1 |

**Figure 8-20. Register 04 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11-0 | RESULT_MSB_FIFO1 | R | 0h | RESULT_MSB Register from FIFO 1 |

### 8.6.1.6 Register 5h (offset = 5h) [reset = 0h]

**Figure 8-21. Register 5h**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT_LSB_FIFO1 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COUNTER_FIFO1 | | | | CRC_FIFO1 | | | |
| R-0h | | | | R-0h | | | |

LEGEND: R/W = Read/Write; W = Write only; -n = value after reset

**Figure 8-22. Register 05 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | RESULT_LSB_FIFO1 | R | 0h | RESULT_LSB Register from FIFO 1 |
| 7-4 | COUNTER_FIFO1 | R | 0h | COUNTER Register from FIFO 1 |
| 3-0 | CRC_FIFO1 | R | 0h | CRC Register from FIFO 1 |

### 8.6.1.7 Register 6h (offset = 6h) [reset = 0h]

**Figure 8-23. Register 6h**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| EXPONENT_FIFO2 | | | | RESULT_MSB_FIFO2 | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT_MSB_FIFO2 | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; W = Write only; -n = value after reset

**Figure 8-24. Register 06 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-12 | EXPONENT_FIFO2 | R | 0h | EXPONENT register from FIFO 2 |
| 11-0 | RESULT_MSB_FIFO2 | R | 0h | RESULT_MSB Register from FIFO 2 |

### 8.6.1.8 Register 7h (offset = 7h) [reset = 0h]

**Figure 8-25. Register 7h**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT_LSB_FIFO2 | | | | | | | |
| R-0h | | | | | | | |

**Figure 8-25. Register 7h (continued)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| COUNTER_FIFO2 | | | | CRC_FIFO2 | | | |
| R-0h | | | | R-0h | | | |

LEGEND: R/W = Read/Write; W = Write only; -n = value after reset

**Figure 8-26. Register 07 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | RESULT_LSB_FIFO2 | R | 0h | RESULT_LSB Register from FIFO 2 |
| 7-4 | COUNTER_FIFO2 | R | 0h | COUNTER Register from FIFO 2 |
| 3-0 | CRC_FIFO2 | R | 0h | CRC Register from FIFO 2 |

### 8.6.1.9 Register 8h (offset = 8h) [reset = 0h]

**Figure 8-27. Register 8h**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| THRESHOLD_L_EXPONENT | | | | THRESHOLD_L_RESULT | | | |
| R/W-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| THRESHOLD_L_RESULT | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; W = Write only; -n = value after reset

**Figure 8-28. Register 08 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-12 | THRESHOLD_L_EXPONENT | R/W | 0h | Threshold low register exponent |
| 11-0 | THRESHOLD_L_RESULT | R/W | 0h | Threshold low register result |

### 8.6.1.10 Register 9h (offset = 9h) [reset = BFFFh]

**Figure 8-29. Register 9h**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| THRESHOLD_H_EXPONENT | | | | THRESHOLD_H_RESULT | | | |
| R/W-Bh | | | | R/W-Fh | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| THRESHOLD_H_RESULT | | | | | | | |
| R/W-FFh | | | | | | | |

LEGEND: R/W = Read/Write; W = Write only; -n = value after reset

**Figure 8-30. Register 09 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-12 | THRESHOLD_H_EXPONENT | R/W | Bh | Threshold high register exponent |
| 11-0 | THRESHOLD_H_RESULT | R/W | FFFh | Threshold high register result |

### 8.6.1.11 Register Ah (offset = Ah) [reset = 3208h]

**Figure 8-31. Register Ah**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| QWAKE | 0 | RANGE | | | | CONVERSION_TIME | |
| R/W-0h | R/W-0h | R/W-Ch | | | | R/W-2h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CONVERSION_TIME | | OPERATING_MODE | | LATCH | INT_POL | FAULT_COUNT | |
| R/W-0h | | R/W-0h | | R/W-1h | R/W-0h | R/W-0h | |

LEGEND: R/W = Read/Write; W = Write only; -n = value after reset

**Figure 8-32. Register 0A Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-15 | QWAKE | R/W | 0h | Quick Wake-up from Standby in one shot mode by not powering down all circuits. Applicable only in One-shot mode and helps get out of standby mode faster with penalty in power consumption compared to full standby mode. |
| 14-14 | 0 | R/W | 0h | Must read or write 0 |
| 13-10 | RANGE | R/W | Ch | Controls the full-scale light level range of the device. The format of this register is same as the EXPONENT register for all values from 0 to 8. PicoStar™ variant:<br>0 : 328lux<br>1 : 655lux<br>2 : 1.3klux<br>3 : 2.6klux<br>4 : 5.2klux<br>5 : 10.5klux<br>6 : 21klux<br>7 : 42klux<br>8 : 83klux<br>12 : Auto-Range<br>SOT-5X3 variant:<br>0 : 459lux<br>1 : 918lux<br>2 : 1.8klux<br>3 : 3.7klux<br>4 : 7.3klux<br>5 : 14.7klux<br>6 : 29.4klux<br>7 : 58.7klux<br>8 : 117.4klux<br>12 : Auto-range |
| 9-6 | CONVERSION_TIME | R/W | 8h | Controls the device conversion time<br>0 : 600us<br>1 : 1ms<br>2 : 1.8ms<br>3 : 3.4ms<br>4 : 6.5ms<br>5 : 12.7ms<br>6 : 25ms<br>7 : 50ms<br>8 : 100ms<br>9 : 200ms<br>10 : 400ms<br>11 : 800ms |
| 5-4 | OPERATING_MODE | R/W | 0h | Controls device mode of operation<br>0 : Power-down<br>1 : Forced auto-range One-shot<br>2 : One-shot<br>3 : Continuous |

**Figure 8-32. Register 0A Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3-3 | LATCH | R/W | 1h | Controls the functionality of the interrupt reporting mechanisms for INT pin for the threshold detection logic. |
| 2-2 | INT_POL | R/W | 0h | Controls the polarity or active state of the INT pin.<br>0 : Active Low<br>1 : Active High |
| 1-0 | FAULT_COUNT | R/W | 0h | Fault count register instructs the device as to how many consecutive fault events are required to trigger the threshold mechanisms: the flag high (FLAG_H) and the flag low (FLAG_L) registers.<br>0 : One fault Count<br>1 : Two Fault Counts<br>2 : Four Fault Counts<br>3 : Eight Fault Counts |

### 8.6.1.12 Register Bh (offset = Bh) [reset = 8011h]

**Figure 8-33. Register Bh**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | INT_DIR | INT_CFG | | 0 | I2C_BURST |
| R/W-0h | R/W-0h | R/W-0h | R/W-1h | R/W-0h | | R/W-0h | R/W-1h |

LEGEND: R/W = Read/Write; W = Write only; -n = value after reset

**Figure 8-34. Register 0B Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-5 | 1024 | R/W | 400h | Must read or write 1024 |
| 4-4 | INT_DIR | R/W | 1h | Determines the direction of the INT pin.<br>0 : Input<br>1 : Output |
| 3-2 | INT_CFG | R/W | 0h | Controls the output interrupt mechanism after end of conversion<br>0 : SMBUS Alert<br>1 : INT Pin asserted after every conversion<br>2: Invalid<br>3: INT pin asserted after every 4 conversions (FIFO full) |
| 1-1 | 0 | R/W | 0h | Must read or write 0 |
| 0-0 | I2C_BURST | R/W | 1h | When set, enables I2C burst mode minimizing I2C read cycles by auto incrementing read register pointer by 1 after every register read |

### 8.6.1.13 Register Ch (offset = Ch) [reset = 0h]

**Figure 8-35. Register Ch**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | OVERLOAD_FLAG | CONVERSION_READY_FLAG | FLAG_H | FLAG_L |

**Figure 8-35. Register Ch (continued)**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R-0h | R-0h | R-0h |

LEGEND: R/W = Read/Write; W = Write only; -n = value after reset

**Figure 8-36. Register 0C Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-4 | 0 | R/W | 0h | Must read or write 0 |
| 3-3 | OVERLOAD_FLAG | R | 0h | Indicates when an overflow condition occurs in the data conversion process, typically because the light illuminating the device exceeds the full-scale range. |
| 2-2 | CONVERSION_READY_FLAG | R | 0h | Conversion ready flag indicates when a conversion completes. The flag is set to 1 at the end of a conversion and is cleared (set to 0) when register address 0xC is either read or written with any non-zero value<br>0 : Conversion in progress<br>1 : Conversion is complete |
| 1-1 | FLAG_H | R | 0h | Flag high register identifies that the result of a conversion is measurement than a specified level of interest. FLAG_H is set to 1 when the result is larger than the level in the THRESHOLD_H_EXPONENT and THRESHOLD_H_RESULT registers for a consecutive number of measurements defined by the FAULT_COUNT register. |
| 0-0 | FLAG_L | R | 0h | Flag low register identifies that the result of a measurement is smaller than a specified level of interest. FL is set to 1 when the result is smaller than the level in the THRESHOLD_LOW_EXPONENT and THRESHOLD_L_RESULT registers for a consecutive number of measurements defined by the FAULT_COUNT register. |

### 8.6.1.14 Register 11h (offset = 11h) [reset = 121h]

**Figure 8-37. Register 11h**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | DIDL | | DIDH | | | |
| R/W-0h | R/W-0h | R-0h | | R-1h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIDH | | | | | | | |
| R-21h | | | | | | | |

LEGEND: R/W = Read/Write; W = Write only; -n = value after reset

**Figure 8-38. Register 11 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-14 | 0 | R/W | 0h | Must read or write 0 |
| 13-12 | DIDL | R | 0h | Device ID L |
| 11-0 | DIDH | R | 121h | Device ID H |