# TMP139 0.5 °C Accuracy, JEDEC DDR5 Grade B, Digital Temperature Sensor With I²C and I3C Interface

## 1 Features

- Supports JEDEC JESD302-1 DDR5 Grade B temperature sensor
- Exceeds JEDEC temperature accuracy specification:
  - ±0.25 °C typical
  - ±0.5 °C maximum (+75 °C to +95 °C)
  - ±0.75 °C maximum (–40 °C to +125 °C)
- Operating temperature range: –40 °C to +125 °C
- Low power consumption:
  - 4.7-µA typical average quiescent current
  - 0.6-µA typical standby current
- I/O power supply of 1 V
- Core power supply of 1.8 V
- Two wire serial bus interface (I²C and I3C basic operation modes)
- Up to 12.5-MHz data transfer rate in I3C basic mode
- In Band Interrupt (IBI) for alerting host
- Parity error check function for host writes
- Packet error check function for host read and writes
- 11-bit resolution: 0.25 °C (1 LSB)
- Standard 6-ball DSBGA (WCSP) package with 0.5-mm pitch

## 2 Applications

- DDR5 DIMM modules
- Server
- Laptops
- Workstations
- SSDs

## 3 Description

The TMP139 is a high-accuracy temperature sensor with an I²C / I3C compliant digital interface supporting In Band Interrupts (IBI). Supporting the interface requirements of JEDEC JESD302-1 for Grade-B devices, the TMP139 exceeds the temperature accuracy requirements of the specification, enabling higher performance DDR5 memory modules. Available in a compact 6-ball DSBGA package, TMP139 is designed for high-speed, high-accuracy and low-power thermal monitoring applications.

The TMP139 has a typical accuracy of ±0.25 °C over the entire temperature range from –40 °C to +125 °C and offers an on-chip 11-bit analog-to-digital converter (ADC) providing a temperature resolution of 0.25 °C.

The TMP139 is designed to operate from a core power supply of 1.8 V and I/O power supply of 1 V, with a low typical average quiescent current of 4.7 µA when performing conversions every 125 ms.

**Table 3-1. Device Information**

| PART NUMBER | PACKAGE(1) | BODY SIZE (NOM) |
|---|---|---|
| TMP139 | DSBGA (6) | 1.328 mm × 0.828 mm |

(1) For all available packages, see the orderable addendum at the end of the data sheet.

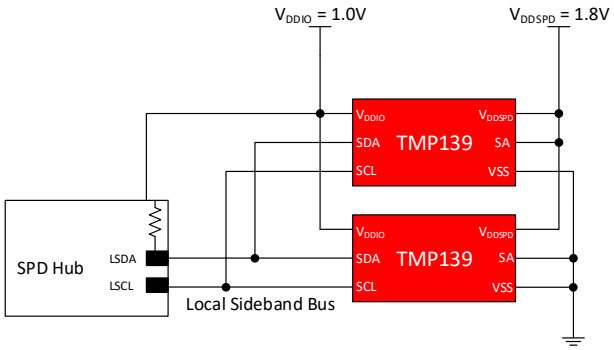



**Figure 3-1. Simplified Schematic**

# Table of Contents



## 4 Revision History

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

| DATE | REVISION | NOTES |
|---|---|---|
| December 2020 | * | Initial release. |

# 5 Pin Configuration and Functions

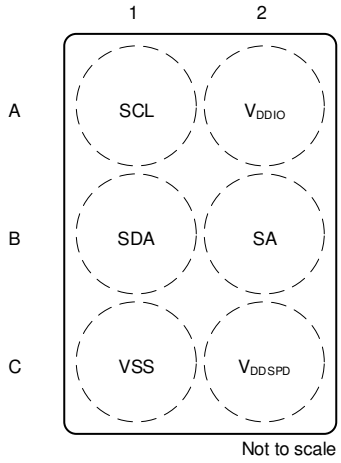



**Figure 5-1. YAH Package 6-Pin DSBGA Top View**

**Table 5-1. Pin Functions**

| PIN | | I/O | DESCRIPTION |
|---|---|---|---|
| NAME | BALL | | |
| SA | B2 | I | Address select. Connected to $V_{DDSPD}$ or GND |
| SCL | A1 | I | Serial clock |
| SDA | B1 | I/O | Serial data input and output. Pin may be open drain or push-pull in I3C mode and open drain in $I^2C$ mode |
| $V_{DDIO}$ | A2 | I | Supply voltage for sensor I/Os |
| $V_{DDSPD}$ | C2 | I | Supply voltage for sensor core |
| VSS | C1 | — | Ground |

# 6 Specifications

## 6.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)(1)

| | MIN | MAX | UNIT |
|---|---|---|---|
| Power supply, $V_{DDIO}$ | –0.5 | 2.1 | V |
| Power supply, $V_{DDSPD}$ | –0.5 | 2.1 | V |
| Input voltage SA | –0.5 | 2.1 | V |
| Input voltage SCL, SDA | –0.5 | $V_{DDIO}$ + 0.3 | V |
| Output sink current SDA | | ±15 | mA |
| Junction temperature, $T_J$ | –55 | 150 | °C |
| Storage temperature, $T_{stg}$ | –65 | 150 | °C |

(1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

## 6.2 ESD Ratings

| | | | VALUE | UNIT |
|---|---|---|---|---|
| $V_{(ESD)}$ | Electrostatic discharge | Human body model (HBM), per ANSI/ESDA/JEDEC JS-001(1) | ±2000 | V |
| | | Charged device model (CDM), per JEDEC specification JESD22-C101(2) | ±1000 | |

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
(2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

## 6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

| | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|
| Supply voltage | $V_{DDIO}$ | 0.95 | 1.0 | 1.05 | V |
| | $V_{DDSPD}$ | 1.7 | 1.8 | 1.98 | V |
| I/O Voltage | SA | 0 | | $V_{DDSPD}$ + 0.3 | V |
| | SCL, SDA | 0 | | $V_{DDIO}$ + 0.3 | V |
| Operating free-air temperature, $T_A$ | | –40 | | 125 | °C |

## 6.4 Thermal Information

| THERMAL METRIC(1) | | TMP139 YAH (WCSP) 6 PINS | UNIT |
|---|---|---|---|
| $R_{\theta JA}$ | Junction-to-ambient thermal resistance | 116.6 | °C/W |
| $R_{\theta JC(top)}$ | Junction-to-case (top) thermal resistance | 1.0 | °C/W |
| $R_{\theta JC(bottom)}$ | Junction-to-case (bottom) thermal resistance | NA | °C/W |
| $R_{\theta JB}$ | Junction-to-board thermal resistance | 33.6 | °C/W |
| $\Psi_{JT}$ | Junction-to-top characterization parameter | 0.4 | °C/W |
| $\Psi_{JB}$ | Junction-to-board characterization parameter | 33.6 | °C/W |

(1) For more information about traditional and new thermal metrics, see the Semiconductor and IC Package Thermal Metrics application report.

## 6.5 Electrical Characteristics

at $T_A$ = −40 °C to +125 °C, $V_{DDIO}$ = 0.95 V to 1.05 V and $V_{DDSPD}$ = 1.7 V to 1.98 V (unless noted); typical specification are at $T_A$ = 25 °C, $V_{DDIO}$ = 1 V and $V_{DDSPD}$ =1.8 V

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| **TEMPERATURE INPUT** | | | | | | |
| $T_{ERR}$ | Temperature Accuracy | +75 °C to +95 °C | | ±0.25 | ±0.5 | °C |
| | | –40 °C to +125 °C | | ±0.25 | ±0.75 | °C |
| $T_{RES}$ | Resolution | 1 LSB (11-bit) | | 0.25 | | °C |
| $T_{REPEAT}$ | Repeatability(1) | | | 1 | | LSB |
| $t_{ACT}$ | Active conversion time | | | 5.5 | | ms |
| $t_{CONV}$ | Conversion interval | | | 125 | | ms |
| $T_{HYST}$ | Temperature Hysteresis | | 1 | | | °C |
| **DIGITAL INPUT/OUTPUT** | | | | | | |
| $C_{IN}$ | Input capacitance(2) | Input capacitance (SCL and SDA) | | | 4 | pF |
| $R_{ON}$ | Output pullup and pulldown driver impedance | SDA pin | 20 | | 100 | Ω |
| $I_{LI}$ | Leakage input current | | -1 | 0 | 1 | µA |
| $I_{LO}$ | Leakage output current | | -1 | 0 | 1 | µA |
| $V_{IL}$ | Low-level input logic | | –0.3 | | 0.3 | V |
| $V_{IH}$ | High-level input logic | | 0.7 | | 1.35 | V |
| $V_{HYS}$ | Input voltage hysteresis | SCL and SDA pins | 60 | 100 | | mV |
| $V_{OL}$ | Low-level output logic | SDA pin, $I_{OL}$ = –3 mA | 0 | | 0.3 | V |
| $V_{OH}$ | High-level output logic | SDA pin, $I_{OH}$ = 3 mA | 0.75 | | | V |
| SLEW_RATE | Output slew rate(2) | SDA pin | 0.1 | | 1.0 | V/ns |
| **POWER SUPPLY** | | | | | | |
| $I_Q$ | Average current (serial bus inactive) | 125-ms conversion interval | | 4.7 | 10 | µA |
| $I_{DDR}$ | Average current (read operation) | 125-ms conversion interval, $f_{SCL}$ = 12.5 MHz | | 34 | | µA |
| $I_{DDW}$ | Average current (write operation) | 125-ms conversion interval, $f_{SCL}$ = 12.5 MHz | | 30 | | µA |
| $I_{ACT}$ | Active current | During 5.5-ms active conversion | | 92 | 140 | µA |
| $I_{DD1}$ | Standby current | Between active conversion during continuous conversion | | 0.6 | 4 | µA |
| $V_{PON}$ | Power-on reset threshold | Monotonic rise between $V_{PON}$ and $V_{DDSPD(MIN)}$ | 1.6 | | | V |
| $V_{POFF}$ | Power-off reset threshold for warm power on cycle | No ringback above $V_{POFF}$ | | | 0.3 | V |
| $t_{INIT}$ | Initialization time after Power-on reset(2) | Figure 7-2 | | | 10.0 | ms |
| $t_{POFF}$ | Warm power cycle off time(2) | Figure 7-3 | 1.0 | | | ms |
| $t_{SENSE\_SA}$ | Time from valid $V_{DDSPD}$ supply to sense SA pin for LID code assignment(2) | Figure 7-2 | | | 5.0 | ms |
| $t_{RST}$ | Device reinitialization time(2) (3) | | | | 40 | µs |

(1) Repeatability is the ability to reproduce a reading when the measured temperature is applied consecutively, under the same conditions.
(2) Parameter is specified by design
(3) Parameter is specified for RSTDAA Common Command Code

## 6.6 Timing Requirements

minimum and maximum specifications are over –40 °C to 125 °C and $V_{DDIO}$ = 0.95 V to 1.05 V (unless otherwise noted)(1)

| | | I²C MODE - OPEN DRAIN | | I3C MODE - PUSH PULL(1) | | UNIT |
|---|---|---|---|---|---|---|
| | | MIN | MAX | MIN | MAX | |
| $f_{SCL}$ | SCL operating frequency | 0.01 | 1 | 0.001 | 12.5 | MHz |
| $t_{HiGH}$ | Clock pulse width high time (Figure 6-1) | 260 | | 35 | | ns |
| $t_{LOW}$ | Clock pulse width low time (Figure 6-1) | 500 | | 35 | | ns |
| $t_{TIMEOUT}$ | Detect clock low timeout (Figure 7-4) | 10 | 50 | 10 | 50 | ms |
| $t_R$ | SDA rise time (Figure 6-1) | | 120 | | 5 | ns |
| $t_F$ | SDA fall time (Figure 6-1) | 4 | 120 | | 5 | ns |
| $t_{SUDAT}$ | Data setup time (Figure 6-1) | 50 | | 8 | | ns |
| $t_{HDDI}$ | Data hold time(2) (Figure 6-1) | 0 | | 3 | | ns |
| $t_{SUSTA}$ | START condition setup time (Figure 6-1) | 260 | | 19.2 | | ns |
| $t_{HDSTA}$ | Hold time after repeated START condition. After this period, the first clock is generated. (Figure 6-1) | 260 | | 38.4 | | ns |
| $t_{SUSTO}$ | STOP condition setup time (Figure 6-1) | 260 | | 19.2 | | ns |
| $t_{BUF}$ | Time between STOP condition and next START condition (Figure 6-1) | 500 | | 500 | | ns |
| $t_{AVAL}$ | Bus available time (no edges seen in SDA and SCL) | | | 1 | | µs |
| $t_{IBI\_ISSUE}$ | Time to issue IBI after an event is detected when bus is available | | | | 15 | µs |
| $t_{CLR\_I3C\_CMD\_DELAY}$ | Time from Clear Register Status to any I3C operation with START condition. PEC disabled | | | 4 | | µs |
| | Time from Clear Register Status to any I3C operation with START condition. PEC enabled | | | 15 | | µs |
| $t_{HDDAT}$ | SCL falling clock in to SDA data out hold time (Figure 6-4) | 0.5 | 350 | | | ns |
| $t_{DOUT}$ | SCL falling clock in to SDA valid data out time (Figure 6-2, Figure 6-3, Figure 6-5) | | | 0.5 | 12 | ns |
| $t_{DOFFS}$ | SCL rising clock in to SDA output off (Figure 6-2, Figure 6-3) | | | 0.5 | 12 | ns |
| $t_{DOFFM}$ | SCL rising clock in to host controller SDA output off | | | 0.5 | 30 | ns |
| $t_{CL\_R\_DAT\_F}$ | SCL rising clock in to host controller driving SDA low (Figure 6-2) | | | 40 | | ns |
| $t_{DEVCTRLCCC\_PEC\_DIS}$ | DEVCTRL CCC followed by DEVCTRL CCC or register read/write command delay | 3 | | 3 | | µs |
| $t_{WR\_RD\_DECLAY\_PEC\_EN}$ | Register write command followed by register read command delay in PEC enabled mode | | | 8 | | µs |
| $t_{I2C\_CCC\_UPDATE\_DELAY}$ | SETHID CCC or SETAASA CCC to any other CCC or read/write command delay | 2.5 | | | | µs |
| $t_{I3C\_CCC\_UPDATE\_DELAY}$ | RSTDAA CCC or ENEC CCC or DISEC CCC to any other CCC or read/write command delay | | | 2.5 | | µs |
| $t_{CCC\_DELAY}$ | Any CCC to RSTDAA CCC delay | | | 2.5 | | µs |

(1) The host and device have the same $V_{DD}$ value. Values are based on statistical analysis of samples tested during initial release.
(2) The maximum $t_{(HDDAT)}$ can be 0.9 µs for fast mode, and is less than the maximum $t_{(VDAT)}$ by a transition time.

## 6.7 Switching Characteristics

over operating free-air temperature range (unless otherwise noted)

| | | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $t_{LPF}$ | Spike filter for I3C compatibility valid in I²C mode only | SCL= 12.5 MHz | | | 50 | ns |

## 6.8 Timing Diagrams

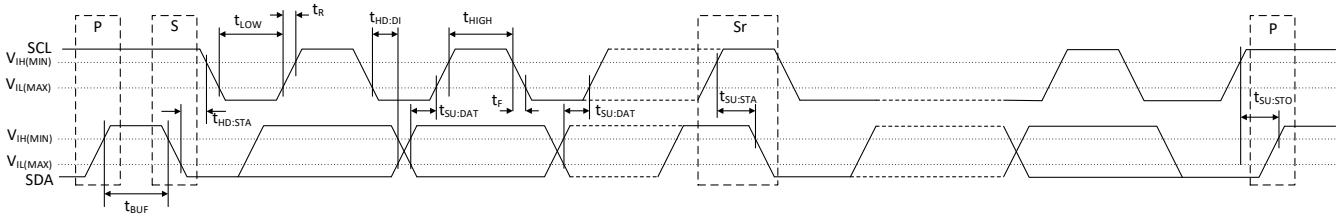



**Figure 6-1. I²C and I3C Basic Bus Input Timing Diagram**

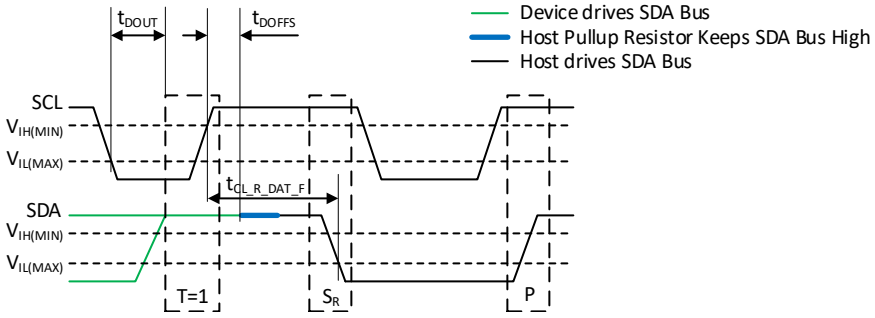



**Figure 6-2. T = 1 Host Ends Read with Repeated Start and Stop Timing Diagram**

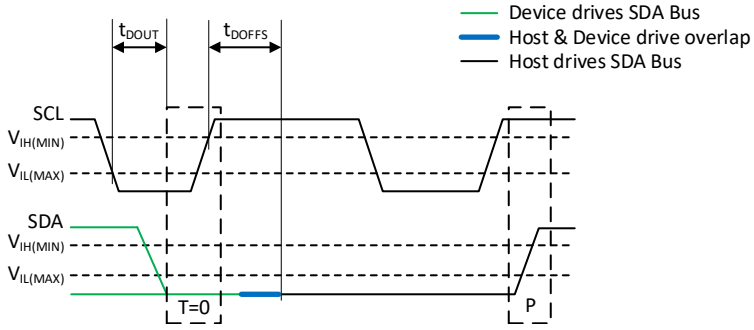



**Figure 6-3. T = 0 Device Ends Read and Host Generates Stop Timing Diagram**

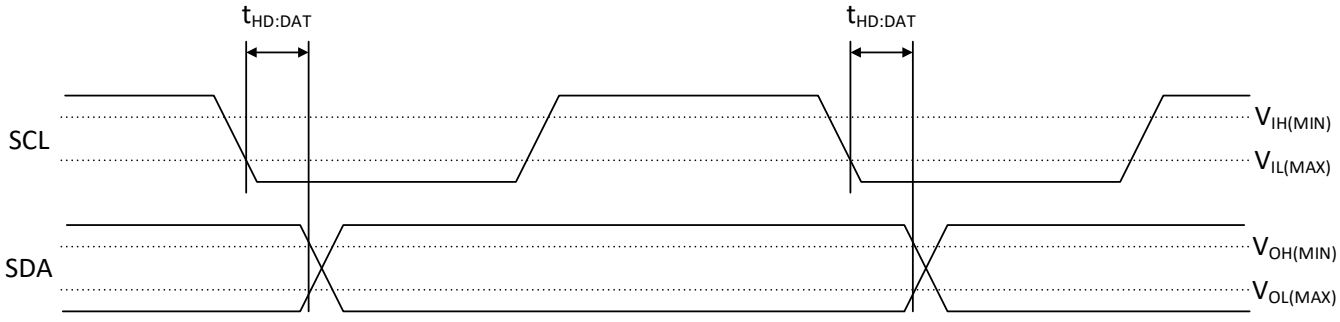



**Figure 6-4. I²C Basic Bus Output Timing Diagram**

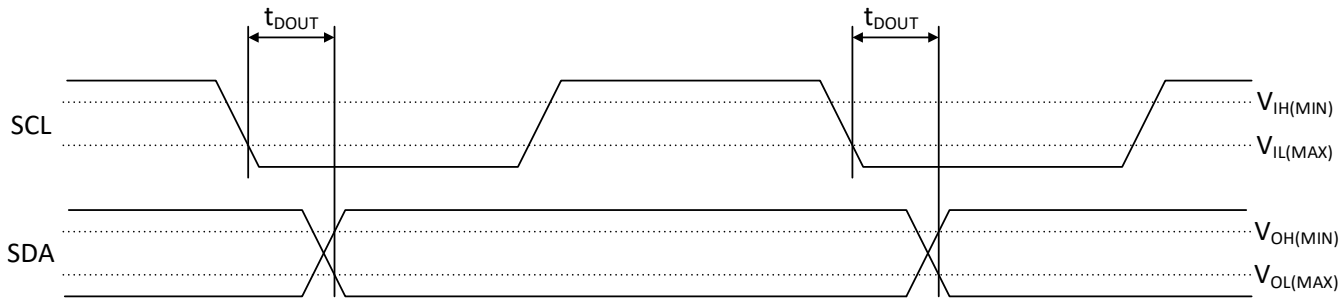



**Figure 6-5. I3C Basic Bus Output Timing Diagram**

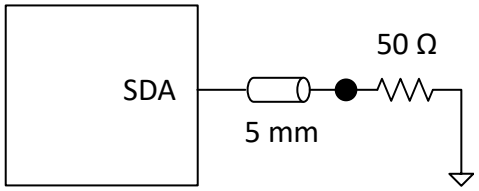



**Figure 6-6. Output Slew Rate and Output Timing Reference Load**

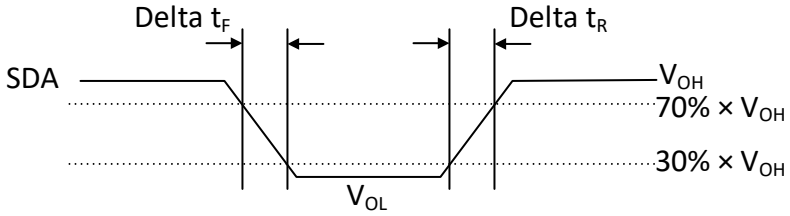



**Figure 6-7. Output Slew Rate Measurement Points**

## 6.9 Typical Characteristics

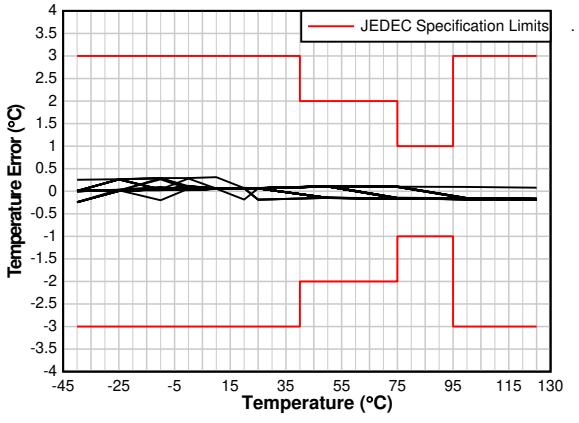
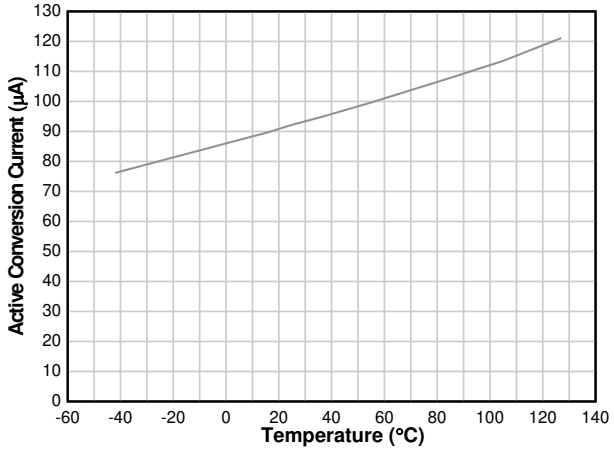
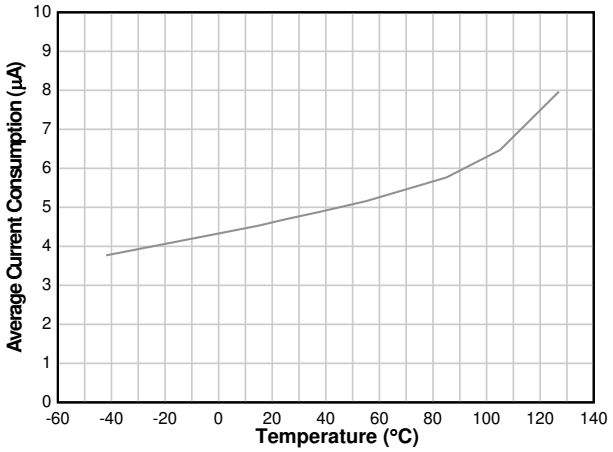



**Figure 6-8. Temperature Error vs Temperature**



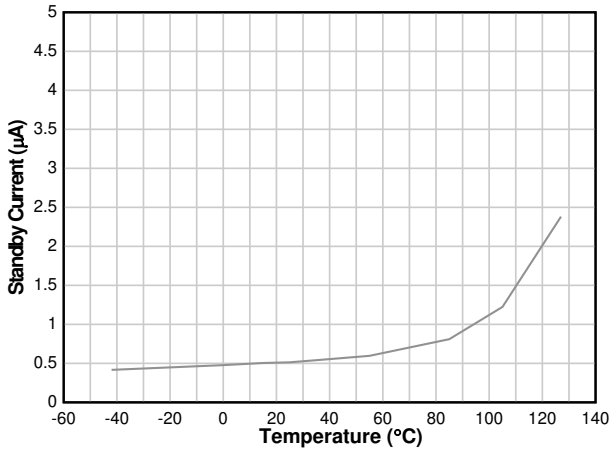

$V_{DDIO}$ = 1 V. $V_{DDSPD}$ = 1.8 V

**Figure 6-9. Active Conversion Current vs Temperature**



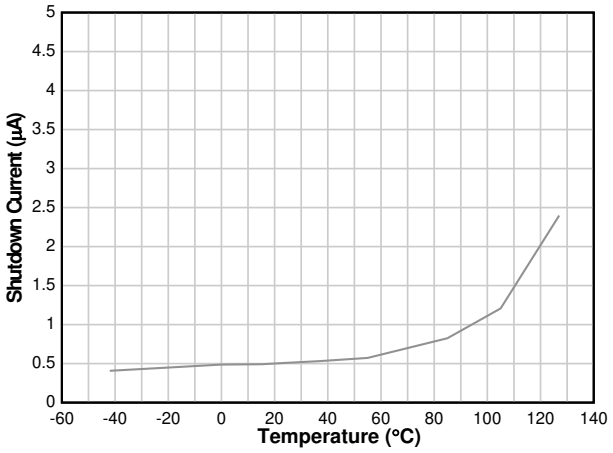

$V_{DDIO}$ = 1 V. $V_{DDSPD}$ = 1.8

**Figure 6-10. Average Current vs Temperature**




$V_{DDIO}$ = 1 V. $V_{DDSPD}$ = 1.8

**Figure 6-11. Standby Current vs Temperature**




$V_{DDIO}$ = 1 V. $V_{DDSPD}$ = 1.8

**Figure 6-12. Shutdown Current vs Temperature**

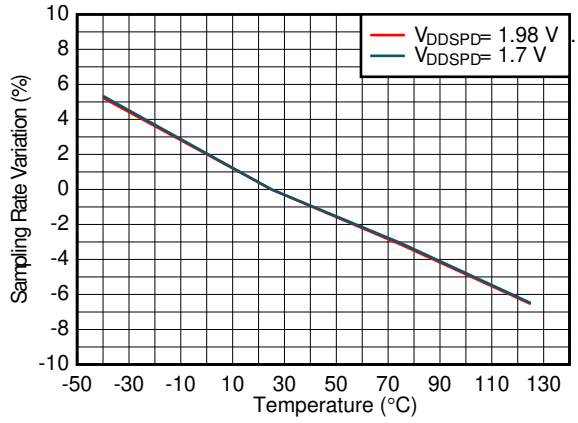



**Figure 6-13. Sampling Rate Change**

# 7 Detailed Description

## 7.1 Overview

The TMP139 is a high-accuracy temperature sensor that supports a power-up sequence, power-down and device reset, parity and packet error check functions, In Band Interrupts (IBI), and common command codes (CCC).

## 7.2 Functional Block Diagram

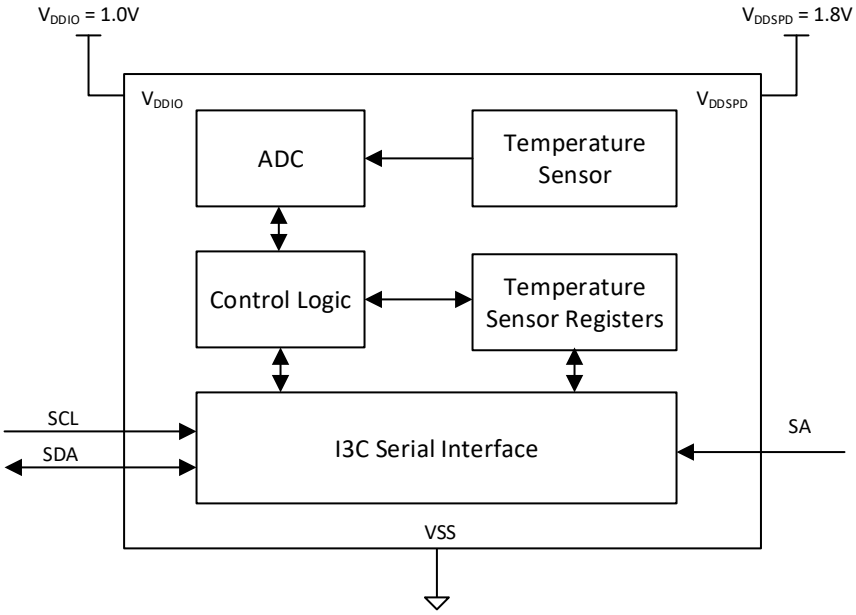



**Figure 7-1. TMP139 Functional Block Diagram**

## 7.3 Feature Description

### 7.3.1 Power-Up Sequence

The TMP139 has two supply pins: $V_{DDSPD}$ which is the core supply, and $V_{DDIO}$ which is the IO supply. To ensure that the device starts up correctly, the application must power up $V_{DDSPD}$ first followed by $V_{DDIO}$. Additionally, the power-on reset (POR) circuit is implemented to prevent improper operation in case of an incorrect power-up sequence.

As shown in Figure 7-2, the $V_{DDSPD}$ supply is applied first and must rise monotonically between $V_{PON(min)}$ and $V_{DDSPD(min)}$ without ring back. The $V_{DDIO}$ supply must ramp up next and must reach the correct level before any operation can be performed.

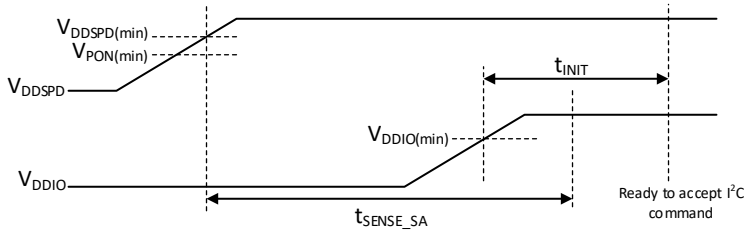



**Figure 7-2. Power-Up Sequence**

When the $V_{DDSPD}$ and $V_{DDIO}$ supply have ramped up above the minimum threshold values, the TMP139 performs the following steps:

1. Within the time $t_{SENSE\_SA}$, the device samples the SA pin to configure the LID code which forms part of the device address.

2. Within time $t_{INIT}$, enables the interface to accept the command from the host.

The device always powers up in the $I^2C$ mode of operation.

### 7.3.2 Power-Down and Device Reset

When the $V_{DDSPD}$ supply decreases, the device operation is not ensured below $V_{DDSPD(min)}$ level. To ensure that the device operates correctly, the application must ensure that the $V_{DDIO}$ and $V_{DDSPD}$ must remain below $V_{POFF}$ for $T_{POFF}$ as shown in Figure 7-3. Once the condition is met, the device shall be reset properly and the power-up sequence will initialize the device correctly.

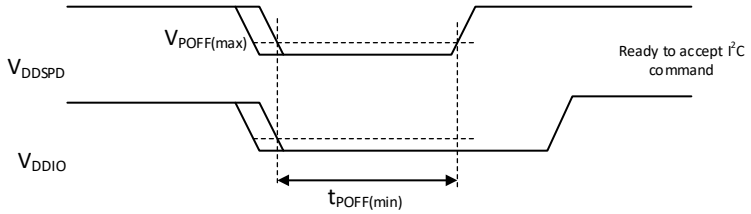



**Figure 7-3. Power-Down and Reset Sequence**

### 7.3.3 Temperature Result and Limits

All temperature result and limit registers are eleven bit values stored in two consecutive registers. The low byte register comes first followed by the high byte register as shown in Table 7-1 for the register map. The data is represented as a 11-bit signed number with the most significant bit for the temperature format being the signed bit. Each of the temperature value bits is assigned a weight which can be used to compute the temperature value. All unused bits read as 0 and any attempt to write a unused bit shall have no effect. The resolution of the temperature result and limit registers is always 0.25 °C and the values can range from –255.75 °C to +255.75 °C, even though the recommended operating range is from –40 °C to +125 °C.

**Table 7-1. Temperature Register Format**

| REGISTER BYTE | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|---|---|---|---|---|---|---|---|---|
| Low Byte | 8 | 4 | 2 | 1 | 0.5 | 0.25 | RSVD = 0 | RSVD = 0 |
| High Byte | RSVD = 0 | RSVD = 0 | RSVD = 0 | Sign | 128 | 64 | 32 | 16 |

The Table 7-2 shows examples of the temperature register read and its corresponding conversion in °C.

**Table 7-2. Temperature Register Examples**

| TEMPERATURE (°C) | HIGH BYTE | LOW BYTE |
|---|---|---|
| +255.75 | 0000 1111 | 1111 1100 |
| +125 | 0000 0111 | 1101 0000 |
| +95 | 0000 0101 | 1111 0000 |
| +85 | 0000 0101 | 0101 0000 |
| +75 | 0000 0100 | 1011 0000 |
| +1 | 0000 0000 | 0001 0000 |
| +0.25 | 0000 0000 | 0000 0100 |
| 0 | 0000 0000 | 0000 0000 |
| –0.25 | 0001 1111 | 1111 1100 |
| –1 | 0001 1111 | 1111 0000 |
| –25 | 0001 1110 | 0111 0000 |
| –40 | 0001 1101 | 1000 0000 |
| –255.75 | 0001 0000 | 0000 0000 |

### 7.3.4 Bus Reset

The bus reset mechanism is supported by the TMP139, to prevent a device from locking up the serial bus. The devices on the bus do not drive the SCL, therefore the bus reset mechanism uses the timeout scheme on the SCL as shown in Figure 7-4. When the SCL is held low by the host controller for a time which is greater than $T_{TIMEOUT(max)}$, TMP139 shall be reset and take the following action:

- Interface is reset, and as the bus reset is considered a Stop condition, any pending internal transaction is also cleared.
- The TMP139 returns to $I^2C$ mode of operation, and resets the following registers:
  - MR7 register, DEV_HID_CODE[2:0] is set to 3'b111.
  - MR18 register, PEC_EN, PAR_DIS and INF_SEL are set to 1'b0.
  - MR27 register, IBI_ERROR_EN is set to 1'b0.
  - MR52 register, PEC_ERROR_STATUS and PAR_ERROR_STATUS are set to 1'b0.
- TMP139 does not resample the SA pin.
- TMP139 floats the SDA pin so that the bus controller can pull up the line.

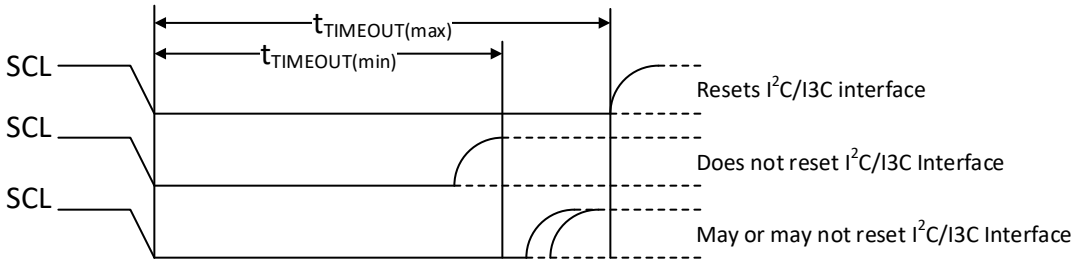



**Figure 7-4. $I^2C$ or I3C Basic Bus Reset**

### 7.3.5 Interrupt Generation

The TMP139 does not have a dedicated interrupt or alert pin, but instead supports interrupt generation using In Band Interrupts (IBI) on the SDA pin. Interrupt generation using IBI is only supported during I3C mode of operation. Hence the application must ensure that the device is first programmed to work in I3C mode before it enables the IBI. As there are multiple devices on the I3C basic bus—each capable of generating an IBI—an arbitration process is required.

The TMP139 generates an IBI only when it sees the bus in idle state for $T_{AVAL}$. Once this condition is met, the device pulls the SDA line low by $T_{IBI\_ISSUE}$ to indicate to the host that it has an IBI. The host shall start by driving the SCL low, which creates the Start bus condition. At this point the device shall send its device address on the bus with the R/W bit set.

There can be a condition when the host starts a new bus transaction at the same time as the TMP139 is generating an IBI. In such a case, the TMP139 shall arbitrate along with the host in the device address byte.

### 7.3.6 Parity Error Check

The parity error check implemented by the TMP139 is odd parity. In $I^2C$ mode, parity error check is not supported except for supported common command codes (CCC). In I3C mode the parity error check is supported for both CCC and host to device data transfers. The parity bit is only sent during the host write and the TMP139 shall check the parity to ensure that the data or CCC it received is correct. The device implements odd parity. If an odd number of bits in the byte are set as 1, the parity bit is set as 0. If an even number of bits in the byte are set as 1, the parity bit is set as 1.

If there is a parity error during a data transfer or CCC, then the TMP139 shall drop the bytes after the parity error is detected and shall wait for a Stop condition on the bus.

When a parity error is detected, the device shall set the IBI_STATUS bit in the MR48 register and PAR_ERROR_STATUS bit in the MR52 register.

### 7.3.7 Packet Error Check

The packet error check (PEC) is implemented at a CRC-8 with the polynomial given in Table 7-3.

**Table 7-3. PEC Rule Table**

| PEC Rule | Attributes |
|---|---|
| PEC width | 8-bits |
| PEC polynomial | $X^8 + X^2 + X^1 + 1$ |
| Initial seed value | 00h |
| Input data reflected | No |
| Output data reflected | No |
| XOR value | 00h |

The PEC is supported in I3C mode only and is computed on the device address, the R/W bit and the data packet. The seed value for the PEC function is reset to zero on either a Start or Repeated Start bus condition.

Any host transaction that results from a PEC enable or disable must be followed by a Stop condition on the bus immediately to allow an update on the PEC control bit in the MR18 register.

## 7.4 Device Functional Modes

This section describes the serial address structure of the TMP139 and how the device operates in both I²C mode and I3C basic mode, including the switching between the modes. This section also describes the behavior of the TMP139 during IBI and the bus reset sequence.

### 7.4.1 Conversion Mode

The TMP139 powers up in continuous conversion mode. In this mode, the device shall perform temperature conversions every 125 ms as shown in Figure 7-5.

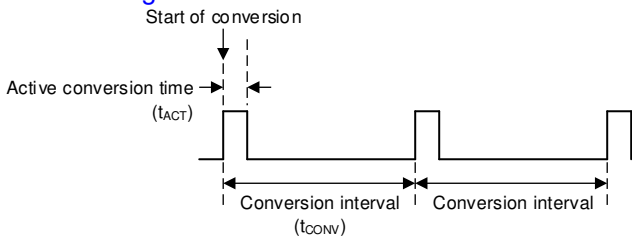



**Figure 7-5. Continuous Conversion Timing Diagram**

The application software can stop the conversion by clearing bit 0 of the MR26 register. When disabled, the device shall not update the result registers. When the temperature sensor is disabled, the host must wait for at least one active conversion cycle for the disable to take effect before any other writes are performed to the device. When the temperature sensor is re-enabled for continuous conversion mode, the host must wait for at least one conversion interval before reading the temperature results. During this time, read to other registers may be performed.

### 7.4.2 Serial Address

The TMP139 has a 7-bit serial address which is used by the host to communicate with the device in both I²C and I3C basic modes of operation. The Table 7-4 shows the serial address format for the TMP139. As described in the power-up sequence, the SA pin is sampled on power up. The sampled value of the SA pin is used to select one of the two possible local device type ID (LID) section of the serial address. The LID is concatenated with the host ID (HID) to form the 7-bit unique serial address.

**Table 7-4. Serial Address Format**

| BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|---|---|---|---|---|---|---|---|
| 0 | SA | 1 | 0 | 1 | 1 | 1 | R/W |
| Local Device Type ID (LID) | | | | Host ID (HID) | | | Read/Write |

If the SA pin is connected to GND, then the serial address for the TMP139 is encoded as 7'b0010111. If the SA pin is connected to $V_{DDSPD}$, then the serial address is encoded as 7'b0110111.

### 7.4.3 I²C Mode Operation

The I²C mode of operation is the primary mode of operation when the device is powered up, goes through a bus reset, or when a RSTDAA CCC is issued if the device is in I3C mode of operation. The maximum bus speed supported in this mode is up to 1.0 MHz. In this mode of operation, the following are not supported:

1. IBI: If IBI is enabled during I3C basic mode, then switching to I²C mode shall disable the IBI enabling mechanism. If there are device events that cause the generation of IBI, then the status of the events shall be logged by the device in the respective register.
2. Packet Error Check: This feature is not supported. If the host attempts to write data with a PEC byte, then the PEC byte shall be treated as a data byte and written to the register address in an incremental format.
3. Parity Error Check: The parity error check is not supported except for the CCC that are listed in Table 7-6.

In the I²C mode of operation, the TMP139 supports SETHID, DEVCTRL and SETAASA CCC, and data transfer packets without PEC. Additionally, a Start or Repeated Start followed by 7'h7E with W = 0 is only allowed for the purpose of issuing the supported CCCs. Any other operation involving a Repeated Start shall be considered illegal.

#### *7.4.3.1 Host I²C Write Operation*

For I²C write operation, the host controller sends the device address with R/W bit as 0, after a Start or Repeated Start as shown in Figure 7-6. This is followed by the 8-bit register address and then the data. The TMP139 writes the data to the register address specified. The internal write register address pointer is incremented after every data byte written. If the write results in an address rollover, then the device shall reset the internal write register address pointer and continue the write operation if possible. The TMP139 does not NACK the data byte for reserved or read-only register, but shall discard the data byte and not update the register.

| S or Sr | 0 | SA | 1 | 0 | HID2 | HID1 | HID0 | R/W=0 | ACK/NACK | |
|---|---|---|---|---|---|---|---|---|---|---|
| | RA = Register Address [7:0] | | | | | | | | ACK | |
| | $Data_{(RA)}$ | | | | | | | | ACK | |
| | $Data_{(RA+1)}$ | | | | | | | | ACK | |
| | … | | | | | | | | ACK | |
| | $Data_{(RA+N)}$ | | | | | | | | ACK | Sr or P |

**Figure 7-6. I²C Write Operation**

#### *7.4.3.2 Host I²C Read Operation*

For I²C read operation, the host controller sends the device address with R/W bit as 0, after a Start or repeated start. This is followed by the 8-bit register address. Once the register address is available to the TMP139, the host issues a Repeated Start and sends the device address with R/W bit as 1. At this point the device shall send the data from the register address incrementally, till the host sends a NACK. If the read operation results in the internal read register address pointer to rollover, then the TMP139 device behavior is not defined.


**Figure 7-7. I²C Read Operation**

### *7.4.3.3 Host I²C Read Operation in Default Read Address Pointer Mode*

The TMP139 provides a default read address pointer mode as shown in Figure 7-8 to read a specific register on the I²C bus. Since the number of bytes to be sent by the host are two less than a standard I²C read operation, this mode provides for a more efficient polling mechanism. The MR18 register, bit DEF_RD_ADDR_POINT_EN is used to enable the mode and bits DEF_RD_ADDR_POINT_Start are used to set the default read address pointer to a specific register in the register map. When enabled, the TMP139 shall set the internal read address pointer to the specific register when there is a Stop condition on the bus.

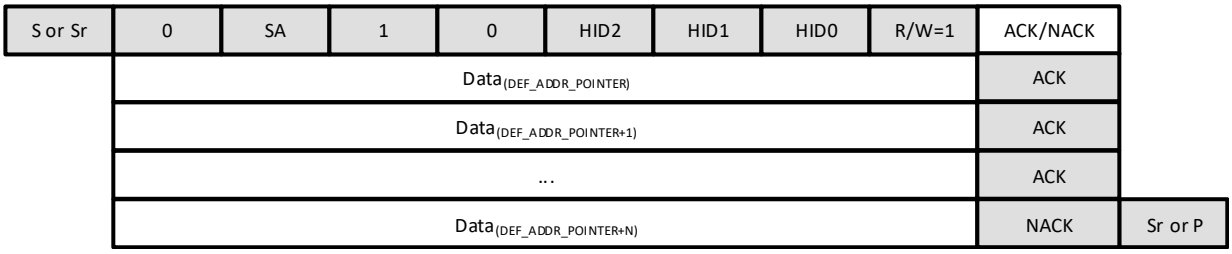



**Figure 7-8. I²C Default Read Address Pointer Mode**

There can be two specific cases in this mode of operation. In the first case as shown in Figure 7-9, there is a normal I²C read preceding the default read mode. If a Stop precedes the Start, then the internal read address pointer shall be set to the default address pointer and subsequent data reads shall result in the data bytes sent by the TMP139 corresponding to the default read address pointer. If a Repeated Start is issued instead of a Stop, then the TMP139 shall send data based on the default read address pointer.

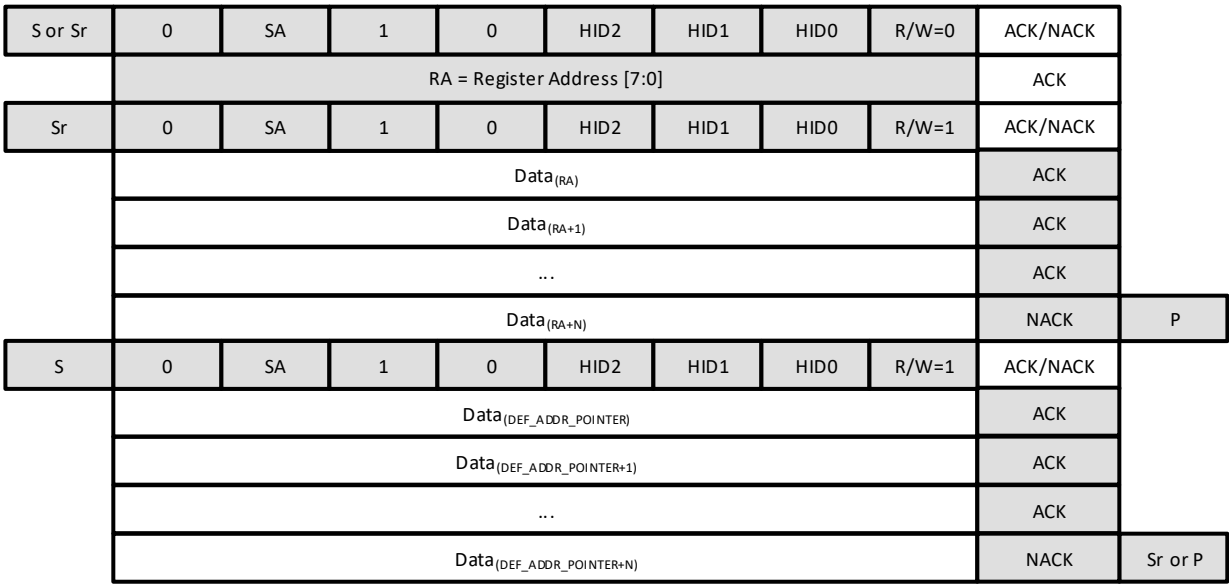



**Figure 7-9. I²C normal read followed by a Default Read Address**

In the second case as shown in Figure 7-10, there is a normal I²C write preceding the default read mode. If there is a Stop, followed by a write bus operation and then a Repeated Start for the read mode, then the TMP139 shall update its internal read address pointer to the default read address and transmit bytes to the host.

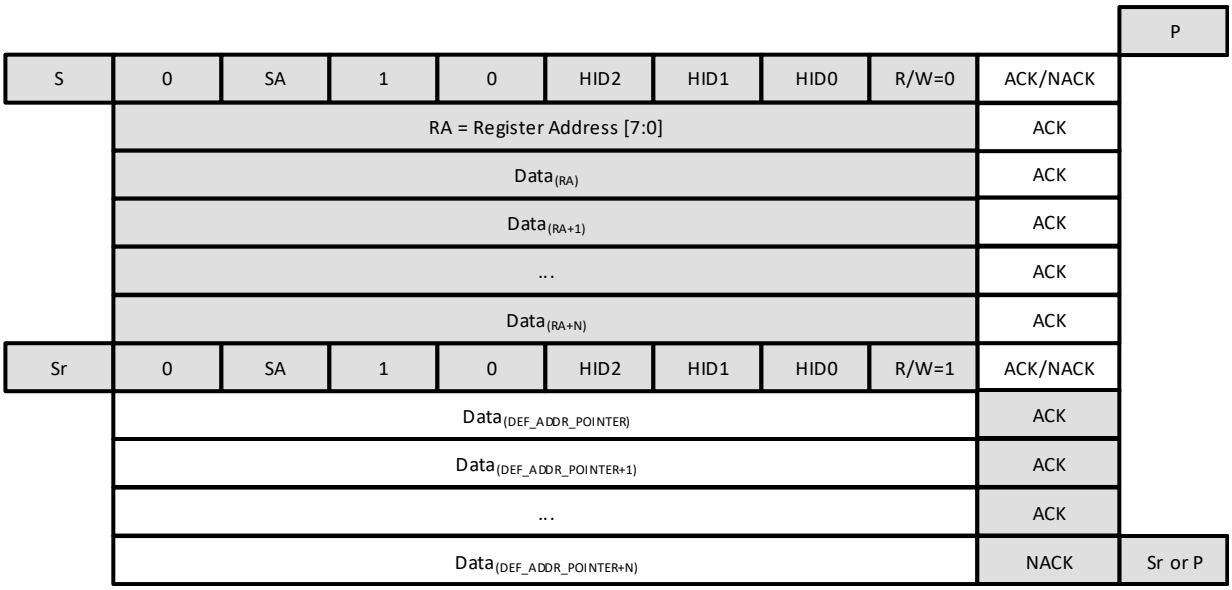



**Figure 7-10. I²C normal write followed by a Default Read Address**

### *7.4.3.4 Switching from I²C Mode to I3C Basic Mode*

As shown in Table 7-6, only DEVCTRL, SETHID and SETAASA CCCs are supported in I2C mode. The host may issue DEVCTRL and/or SETHID, before it can issue SETAASA for switching the device from I²C mode to I3C basic mode.

When the SETAASA is issued by the host, the device shall register the command by setting MR18 register INF_SEL bit to 1'b1 once the Stop condition on the bus is detected. After this, the TMP139 shall be in the I3C basic mode of operation.

### 7.4.4 I3C Basic Mode Operation

As described in the previous section, I3C basic mode of operation is always entered from the I$^2$C mode of operation. When in the I3C basic mode, the device can support data transfer rates of up to 12.5 MHz with a push-pull SDA driver. Additionally, the following may be supported by default or when enabled:

1. IBI: Disabled by default, the IBI can now be enabled.
2. Packet error check: Disabled by default, but the TMP139 can support the PEC feature when enabled by the host.
3. Parity check: Is always enabled by default.

In I3C basic mode of operation, the read and write packets may have different structures. The structure of the data payload shall be dependent on the feature that has been enabled.

#### *7.4.4.1 Host I3C Write Operation without PEC*

As shown in Figure 7-11 and Figure 7-12, an I3C basic write operation is the same as an I$^2$C write operation. For all bytes after the device address field, the 9$^{th}$ bit is the parity bit sent by the host. When the IBI is enabled by the host, it must send the IBI header byte which consists of 7'h7E+R/W=0, before it sends the device address. This allows the participating devices on the bus to arbitrate between themselves if more than one device has an interrupt condition that needs to be communicated to the host.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| S or Sr | 0 | SA | 1 | 0 | HID2 | HID1 | HID0 | R/W=0 | ACK/NACK |
| | RA = Register Address [7:0] | | | | | | | | T |
| | Data$_{(RA)}$ | | | | | | | | T |
| | Data$_{(RA+1)}$ | | | | | | | | T |
| | … | | | | | | | | T |
| | Data$_{(RA+N)}$ | | | | | | | | T Sr or P |

**Figure 7-11. I3C Basic Mode Write**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| S | 1 | 1 | 1 | 1 | 1 | 1 | 0 | R/W=0 | ACK |
| Sr | 0 | SA | 1 | 0 | HID2 | HID1 | HID0 | R/W=0 | ACK/NACK |
| | RA = Register Address [7:0] | | | | | | | | T |
| | Data$_{(RA)}$ | | | | | | | | T |
| | Data$_{(RA+1)}$ | | | | | | | | T |
| | … | | | | | | | | T |
| | Data$_{(RA+N)}$ | | | | | | | | T Sr or P |

**Figure 7-12. I3C Basic Mode Write with IBI Header**

If there is a parity error during the data transfer, the device shall discard all the bytes including the byte for which parity error was detected and set the parity error condition. If the host attempts to start a new transaction with a Repeated Start to the same device, then TMP139 shall NACK the device address to indicate an error condition to the host. The host must first clear the parity error condition before performing any new transfer to the TMP139. When IBI is enabled, the device can communicate to the host the error conditions seen, using IBI. However when IBI is not enabled, it is strongly recommended that the host check the error status register to ensure that no parity error was detected on the bus.

#### *7.4.4.2 Host I3C Write Operation with PEC*

As shown in Figure 7-13 and Figure 7-14, when the PEC is enabled by the host, an additional byte is added by the host after sending the register address. The format for the additional byte is described in Table 7-5.

**Table 7-5. Command Truth Table - PEC Enabled Mode**

| CMD | RW | Command Name | Command Description |
|---|---|---|---|
| 000 | 0 | W1R | Write 1 Byte to Register address specified in data packet |
| | 1 | R1R | Read 1 Byte from Register address specified in data packet |
| 001 | 0 | W2R | Write 2 Bytes to Register address specified in data packet |
| | 1 | R2R | Read 2 Bytes from Register address specified in data packet |
| 010 – 111 | X | RSVD | Reserved |

If the CMD value sent by the host is not valid for TMP139, the device shall not write any data to the register specified.

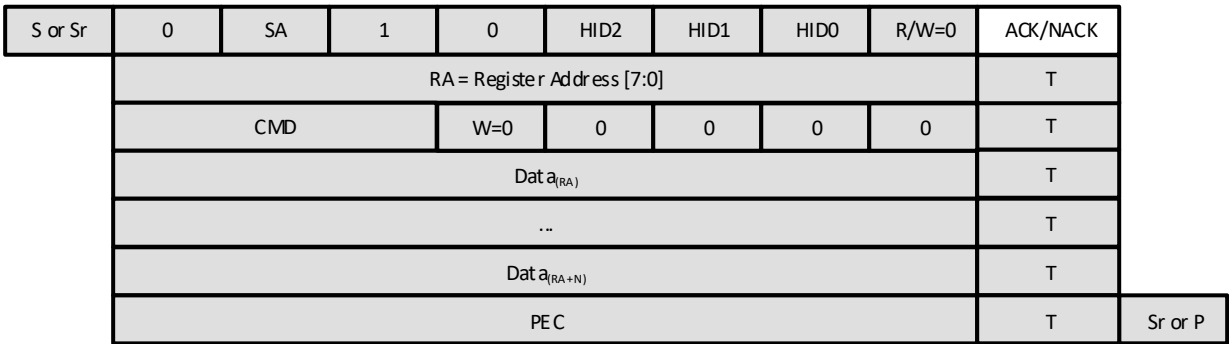



**Figure 7-13. I3C Basic Mode Write with PEC enabled**

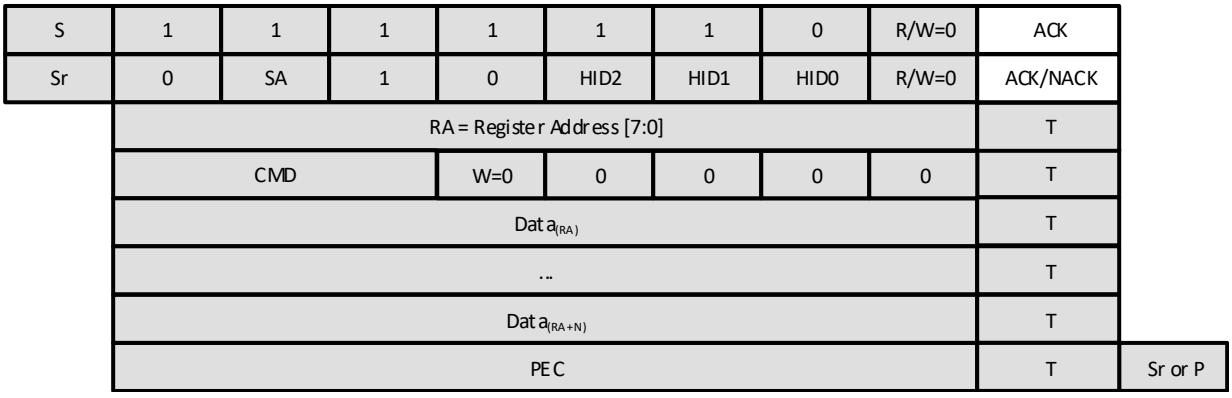



**Figure 7-14. I3C Basic Mode Write with IBI Header and PEC enabled**

If there is a parity error during the data transfer, the device shall discard all the bytes including the byte for which parity error was detected and set the parity error condition. If the host attempts to start a new transaction with a Repeated Start to the same device, then TMP139 shall NACK the device address to indicate an error condition to the host. The host must first clear the parity error condition before performing any new transfer to the TMP139.

If there is a PEC error, then the TMP139 shall discard the entire data packet and set the PEC error condition. If the host attempts to start a new transaction with a Repeated Start to the same device, then TMP139 shall NACK the device address to indicate an error condition to the host. The host must first clear the PEC error condition before performing any new transfer to the TMP139.

When IBI is enabled, the device can communicate to the host the error conditions seen, using IBI. However when IBI is not enabled, it is strongly recommended that the host check the error status register to ensure that no parity or PEC error was detected on the bus.

#### *7.4.4.3 Host I3C Read Operation without PEC*

As shown in Figure 7-15 and Figure 7-16, an I3C basic mode read is same as $I^2C$ read operation. For all bytes sent by the device, the $9^{th}$ bit is the T-bit, which is used by the device and host to negotiate continuation of the read transfer. During the read phase, the device drives the T-bit as 1, before the rising edge to tell the host that it can send more bytes or drives the T-bit as 0, to indicate to the host that the device wants to terminate the transfer and the host shall respond with either a Stop or Repeated Start on the bus. The host may also terminate the transfer by driving the T-bit as 0, only when device sends the T-bit as 1, which creates a repeated start condition on the bus. Additionally, the host may send a Stop on the bus. When the IBI is enabled by the host, it must send the IBI header byte which consists of 7'h7E+R/W = 0, before it sends the device address. This allows the participating devices on the bus, to arbitrate between themselves if more than one device has an interrupt condition that needs to be communicated to the host.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| S or Sr | 0 | SA | 1 | 0 | HID2 | HID1 | HID0 | R/W=0 | ACK/NACK |
| | RA = Register Address [7:0] | | | | | | | | T |
| Sr | 0 | SA | 1 | 0 | HID2 | HID1 | HID0 | R/W=1 | ACK/NACK |
| | $Data_{(RA)}$ | | | | | | | | T=1 |
| | $Data_{(RA+1)}$ | | | | | | | | T=1 |
| | ... | | | | | | | | T=1 |
| | $Data_{(RA+N)}$ | | | | | | | | T=1 Sr or P |

**Figure 7-15. I3C Basic Mode Read**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| S | 1 | 1 | 1 | 1 | 1 | 1 | 0 | R/W=0 | ACK |
| Sr | 0 | SA | 1 | 0 | HID2 | HID1 | HID0 | R/W=0 | ACK/NACK |
| | RA = Register Address [7:0] | | | | | | | | T |
| Sr | 0 | SA | 1 | 0 | HID2 | HID1 | HID0 | R/W=1 | ACK/NACK |
| | $Data_{(RA)}$ | | | | | | | | T=1 |
| | $Data_{(RA+1)}$ | | | | | | | | T=1 |
| | ... | | | | | | | | T=1 |
| | $Data_{(RA+N)}$ | | | | | | | | T=1 Sr or P |

**Figure 7-16. I3C Basic Mode Read with IBI Header**

The TMP139 shall NACK the read phase of the transaction if there was a parity error in the write phase before the repeated start. The device shall also send the T-bit as 0, if the host attempts to read data continuously such that the internal read address pointer reaches 255, which is the last register in the register map table. Additionally, if the host attempts to start a new transaction with a Repeated Start to the same device, when there was a parity error in the previous transaction, then TMP139 shall NACK the device address to indicate an error condition to the host. The host must first clear the parity error condition before performing any new transfer to the TMP139. When IBI is enabled, the device can communicate to the host the error conditions seen, using IBI. However when IBI is not enabled, it is strongly recommended that the host check the error status register to ensure that no parity error was detected on the bus.

### 7.4.4.4 Host I3C Read Operation with PEC

As shown in Figure 7-17 and Figure 7-18, when the PEC is enabled by the host, an additional byte is added by the host after sending the register address. The format for the additional byte is described in Table 7-5. Since only one and two byte reads are permitted by the CMD byte, the device shall terminate the read phase after sending one byte of data and PEC byte or two byte of data and PEC byte, followed by the T-bit as 0. In an unlikely case where the host sets the register address as 255, and attempts a read of two bytes, the device result is not guaranteed.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| S or Sr | 0 | SA | 1 | 0 | HID2 | HID1 | HID0 | R/W=0 | ACK/NACK | |
| | RA = Register Address [7:0] | | | | | | | | T | |
| | CMD | | | R=1 | 0 | 0 | 0 | 0 | T | |
| | PEC | | | | | | | | T | |
| Sr | 0 | SA | 1 | 0 | HID2 | HID1 | HID0 | R/W=1 | ACK/NACK | |
| | $Data_{(RA)}$ | | | | | | | | T=1 | |
| | ... | | | | | | | | T=1 | |
| | $Data_{(RA+N)}$ | | | | | | | | T=1 | |
| | PEC | | | | | | | | T=0 | Sr or P |

**Figure 7-17. I3C Basic Mode Read with PEC enabled**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| S | 1 | 1 | 1 | 1 | 1 | 1 | 0 | R/W=0 | ACK | |
| Sr | 0 | SA | 1 | 0 | HID2 | HID1 | HID0 | R/W=0 | ACK/NACK | |
| | RA = Register Address [7:0] | | | | | | | | T | |
| | CMD | | | R=1 | 0 | 0 | 0 | 0 | T | |
| | PEC | | | | | | | | T | |
| Sr | 0 | SA | 1 | 0 | HID2 | HID1 | HID0 | R/W=1 | ACK/NACK | |
| | $Data_{(RA)}$ | | | | | | | | T=1 | |
| | ... | | | | | | | | T=1 | |
| | $Data_{(RA+N)}$ | | | | | | | | T=1 | |
| | PEC | | | | | | | | T=0 | Sr or P |

**Figure 7-18. I3C Basic Mode Read with PEC enabled and IBI Header**

If the CMD value sent by the host is not valid for TMP139, the device shall NACK the read phase.

The TMP139 shall NACK the read phase of the transaction if there was a parity error in the write phase before the Repeated Start. If the host attempts to start a new transaction with a Repeated Start to the same device, then TMP139 shall NACK the device address to indicate an error condition to the host. The host must first clear the parity error condition before performing any new transfer to the TMP139.

If there is a PEC error, then the TMP139 shall NACK the read phase of the transaction. If the host attempts to start a new transaction with a Repeated Start to the same device, then TMP139 shall NACK the device address to indicate an existing error condition to the host. The host must first clear the PEC error condition before performing any new transfer to the TMP139.

When IBI is enabled, the device can communicate to the host the error conditions seen, using IBI. However when IBI is not enabled, it is strongly recommended that the host check the error status register to ensure that no parity or PEC error was detected on the bus.

#### *7.4.4.5 Host I3C Read Operation in Default Read Address Pointer Mode*

The default read address pointer mode in I3C basic mode works the same way as $I^2C$ mode as shown in Figure 7-19 to Figure 7-22. The device shall also send the T-bit as 0, if the host attempts to read data continuously such that the internal read address pointer reaches 255, which is the last register in the register map table. The host may also terminate the transfer by driving the T-bit as 0 only when PEC is not enabled.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| S or Sr | 0 | SA | 1 | 0 | HID2 | HID1 | HID0 | R/W=1 | ACK/NACK | |
| | $Data_{(RA)}$ | | | | | | | | T=1 | |
| | $Data_{(RA+1)}$ | | | | | | | | T=1 | |
| | ... | | | | | | | | T=1 | |
| | $Data_{(RA+N)}$ | | | | | | | | T=1 | Sr or P |

**Figure 7-19. I3C Basic Mode Default Read Address Enabled**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| S | 1 | 1 | 1 | 1 | 1 | 1 | 0 | R/W=0 | ACK | |
| Sr | 0 | SA | 1 | 0 | HID2 | HID1 | HID0 | R/W=1 | ACK/NACK | |
| | $Data_{(RA)}$ | | | | | | | | T=1 | |
| | $Data_{(RA+1)}$ | | | | | | | | T=1 | |
| | ... | | | | | | | | T=1 | |
| | $Data_{(RA+N)}$ | | | | | | | | T=1 | Sr or P |

**Figure 7-20. I3C Basic Mode Default Read Address Enabled with IBI Header**

When PEC is enabled, then the MR18 register sets the default number of bytes that shall be sent, after which the device shall send the PEC byte with T-bit as 0.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| S or Sr | 0 | SA | 1 | 0 | HID2 | HID1 | HID0 | R/W=1 | ACK/NACK | |
| | $Data_{(RA)}$ | | | | | | | | T=1 | |
| | ... | | | | | | | | T=1 | |
| | $Data_{(RA+N)}$ | | | | | | | | T=1 | |
| | PEC | | | | | | | | T=0 | Sr or P |

**Figure 7-21. I3C Basic Mode Default Read Address Enabled with PEC Enabled**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| S | 1 | 1 | 1 | 1 | 1 | 1 | 0 | R/W=0 | ACK | |
| Sr | 0 | SA | 1 | 0 | HID2 | HID1 | HID0 | R/W=1 | ACK/NACK | |
| | $Data_{(RA)}$ | | | | | | | | T=1 | |
| | ... | | | | | | | | T=1 | |
| | $Data_{(RA+N)}$ | | | | | | | | T=1 | |
| | PEC | | | | | | | | T=0 | Sr or P |

**Figure 7-22. I3C Basic Mode Default Read Address Enabled with PEC Enabled and IBI Header**

The TMP139 shall NACK the address phase, during a Repeated Start, if there was an error in the previous transaction.

### 7.4.5 In Band Interrupt

The In Band Interrupt (IBI) is an elegant method to inform the host of an event in the TMP139. There are two types of events that the TMP139 generates:

1. Error event: Events corresponding to parity or PEC error.
2. Temperature event: Events corresponding to the temperature exceeding the higher temperature limits or falling below the lower temperature limits.

By default, all interrupt sources are disabled when the device powers up. The interrupt source can only be enabled when the device is in I3C basic mode of operation, as enabling the interrupt source shall generate an IBI which is not allowed in $I^2C$ mode of operation. An IBI can only be requested by the TMP139 when the bus has been in an inactive state for the $T_{AVAL}$ period. Once the condition for an inactive state on the bus is met, and there is no bus transaction, the TMP139 shall initiate an IBI by driving the SDA low to indicate to the host of a pending IBI.

#### *7.4.5.1 In Band Interrupt Arbitration Rules*

Based on the state of the host controller readiness and due to the fact that there are multiple devices on the bus, the IBI generation and arbitration must follow some rules, as described below. All of these conditions assume that the bus has been inactive for $T_{AVAL}$ period.

1. When the host controller starts a write or read with IBI header, TMP139 shall start driving its own address on the bus. The host on seeing a value other than the IBI header shall no longer drive the SDA, allowing the TMP139 to transmit its device header along with R/W bit set to 1.
2. If the host controller can accept the IBI from the device, it shall ACK the device address, release the bus on the falling edge of SCL and shall accept the bytes sent by the TMP139.
3. If the host controller cannot accept the IBI from the device, it shall NACK the device address and issue a Stop condition on the bus. The TMP139 shall retry another IBI only after $T_{AVAL}$ period.
4. When the host controller starts a write or read without an IBI header to a device on the bus which has a lower device address than the TMP139, the device on detecting a mismatch, shall no longer participate on the bus and retry another IBI only after $T_{AVAL}$ period.
5. When the host controller starts a write or read without an IBI header to a device on the bus which has a higher device address than the TMP139, the device wins the bus arbitration and the host shall no longer participate on the bus. The host may accept the IBI by sending an ACK or disregard the IBI by sending a NACK. In the latter case, TMP139 shall retry another IBI only after $T_{AVAL}$ period.
6. When the host controller starts a write or read transaction without an IBI header to the TMP139 which is also requesting an IBI, either the host or TMP139 can win.
7. If the host controller starts a write transaction, then it shall win the bus arbitration and the TMP139 shall let go of the bus. The TMP139 shall retry another IBI only after $T_{AVAL}$ period.
8. If the host controller starts a read transaction, then all the bits shall match. However at this point the host is expecting an ACK from the TMP139 for the read request, while the TMP139 is waiting for an ACK from the host for the IBI. As a result there shall be a NACK on the bus. In such a case, the TMP139 shall retry the IBI only after $T_{AVAL}$ period. However if the host issues start (or Repeated Start) and attempts the read transaction before the $T_{AVAL}$ period, it shall get an ACK from the TMP139 and the host read shall win the arbitration on the bus.
9. As described above, in the case when there are multiple devices initiating an IBI at the same time, the device which has the lowest device address shall win the bus arbitration and the TMP139 when it detects a loss on the bus arbitration, shall retry another IBI only after $T_{AVAL}$ period

#### *7.4.5.2 In Band Interrupt Bus Transaction*

As shown in Figure 7-23 and Figure 7-24, when the device has to send an IBI, wins arbitration on the bus and the IBI is ACK by the host, it shall always send the mandatory data byte (MDB) as 8'h00, followed by the MR51 and MR52 register values. After transmitting the last byte, it shall set the T-bit as 0, after which the host controller must send a Stop condition on the bus.

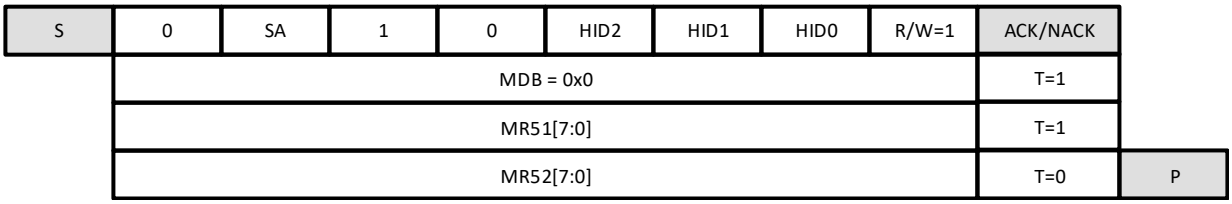



**Figure 7-23. IBI Payload Packet with PEC Disabled**

If PEC is enabled, then after MR52 register value, the PEC byte is sent with the T-bit set as 0. Again, the host must send a Stop condition on the bus.

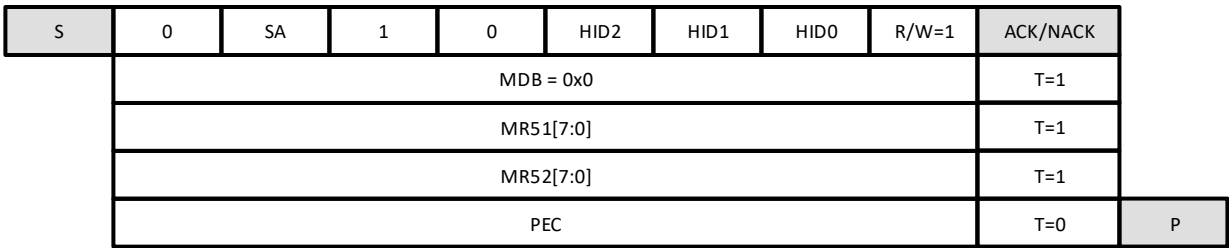



**Figure 7-24. IBI Payload Packet with PEC Enabled**

When an IBI is asserted by the device and it successfully transmits the IBI, including the MDB, MR51, MR52 and PEC (if PEC mode is enabled) bytes, the device shall automatically clear the IBI_STATUS bit in MR48 register.

### 7.4.6 Common Command Codes Support

The TMP139 supports a subset of the CCC as listed in the I3C basic specification and shown in Table 7-6. Only CCC specified in the JESD302-1 are supported and the TMP139 shall either NACK unsupported CCC (if possible) or ignore the actions when on a generic I3C bus. Similarly, for supported CCC—depending on whether the TMP139 is in $I^2C$ or I3C mode—if a non-applicable CCC is sent, the device shall ignore the actions.

The TMP139 requires a Stop condition on the bus after receiving any CCC before it can process a device specific read or write operation. Similarly, when processing a device specific read or write condition, a Stop on the bus should follow before any CCC can be issued.

The TMP139 can receive a direct CCC with a Repeated Start condition after another direct CCC. Similarly, it is valid to send a broadcast CCC following another broadcast CCC with a Repeated Start in between. In such a case, the action taken by the device will only be updated following a Stop condition on the bus. The behavior of TMP139 is not defined if a direct CCC is followed by a broadcast CCC or vice-versa with a Repeated Start. For example, it is a legal combination in $I^2C$ mode to send a SETHID CCC, followed by a Repeated Start, then a SETAASA CCC followed by a Stop condition. However, in I3C mode, sending a direct ENEC CCC followed by Repeated Start and then a broadcast DEVCTRL CCC is not a valid condition for the TMP139. The host must issue a Stop after ENEC CCC, before it sends a broadcast DEVCTRL CCC.

The CCC sent to the TMP139 may either be a broadcast code or a direct code. All CCC operations require the host to send 7'h7E with R/W = 0, followed by the CCC and payload bytes specific to the CCC. For a direct CCC, the host shall issue a Repeated Start on the bus after the CCC byte followed by the payload bytes.

**Table 7-6. Supported CCC**

| CCC | Mode | Code | Description | Applicable in $I^2C$ Mode | Applicable in I3C Mode |
|---|---|---|---|---|---|
| ENEC | Broadcast | 0x00 | Enable Event Interrupts | No | Yes |
| | Direct | 0x80 | | | |
| DISEC | Broadcast | 0x01 | Disable Event Interrupts | No | Yes |
| | Direct | 0x81 | | | |
| RSTDAA | Broadcast | 0x06 | Put the device in $I^2C$ mode | No | Yes |

**Table 7-6. Supported CCC (continued)**

| CCC | Mode | Code | Description | Applicable in I²C Mode | Applicable in I3C Mode |
|---|---|---|---|---|---|
| SETAASA | Broadcast | 0x29 | Put the device in I3C Basic Mode | Yes | No |
| GETSTATUS | Direct | 0x90 | Get Device Status | No | Yes |
| DEVCAP | Direct | 0xE0 | Get Device Capability | No | Yes |
| SETHID | Broadcast | 0x61 | TMP139 updates 3-bit HID field | Yes | No |
| DEVCTRL | Broadcast | 0x62 | Configure Device | Yes | Yes |

#### *7.4.6.1 ENEC CCC*

The ENEC CCC is issued by the host controller to enable the event interrupt generation. The CCC takes effect after a Stop has been issued by the host controller. Once the ENEC is received, the TMP139 shall update the MR27 register bit IBI_ERROR_EN to 1'b1.

**Note**

It is illegal for the host controller to send the ENINT bit as 0.

The command may be issued either as a broadcast command or as a direct command to TMP139 as shown in Figure 7-25 and Figure 7-26.

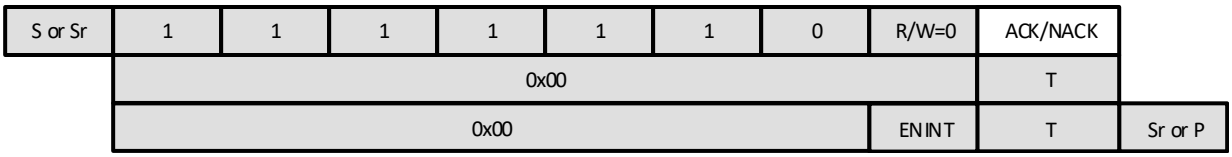



**Figure 7-25. ENEC CCC Broadcast**


**Figure 7-26. ENEC CCC Direct**

The command may be issued either as a broadcast or as a direct command, as shown in Figure 7-27 and Figure 7-28, with PEC Enabled. In such a case the host controller shall append the PEC byte calculated on all bytes except the byte with 7'h7E and R/W=0 after the Start or Repeated Start.

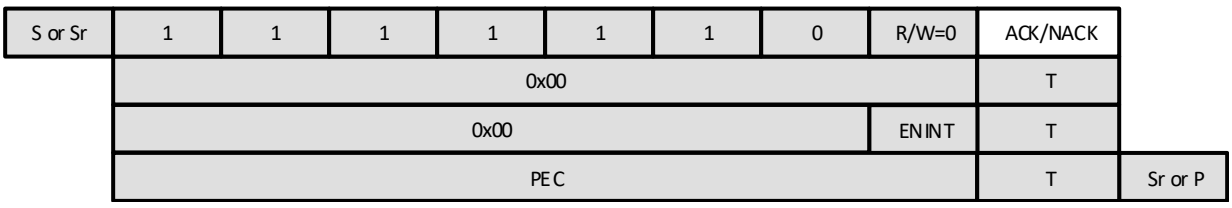



**Figure 7-27. ENEC CCC Broadcast with PEC Enabled**

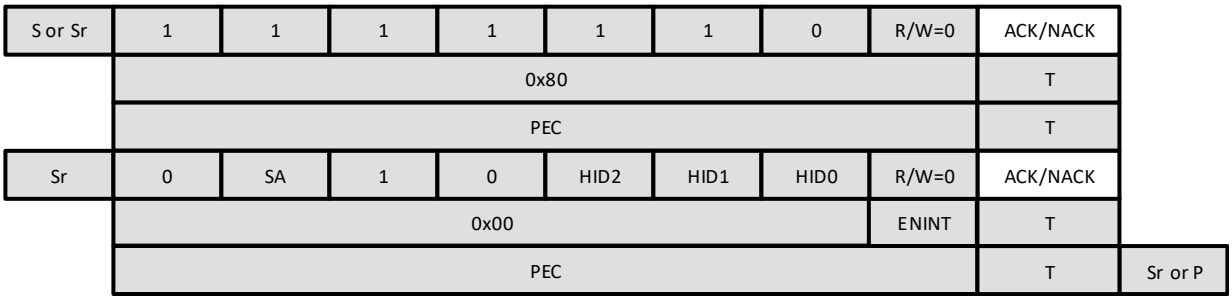



**Figure 7-28. ENEC CCC Direct with PEC Enabled**

**Note**

TMP139 NACKs the ENEC CCC if the previous transaction has a parity or PEC error and the host starts the transaction with a Repeated Start.

### *7.4.6.2 DISEC CCC*

The DISEC CCC is issued by the host controller to disable the event interrupt generation. The CCC takes effect after a Stop has been issued by the host controller. Once the DISEC is received, the TMP139 shall the update MR27 register bit IBI_ERROR_EN to 1'b0.

**Note**

It is illegal for the host controller to send the DISINT bit as 0.

The command may be issued either as a broadcast command or as a direct command to a specific device as shown in Figure 7-29 and Figure 7-30.

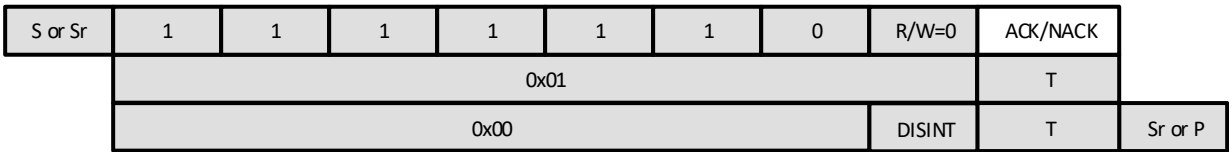



**Figure 7-29. DISEC CCC Broadcast**

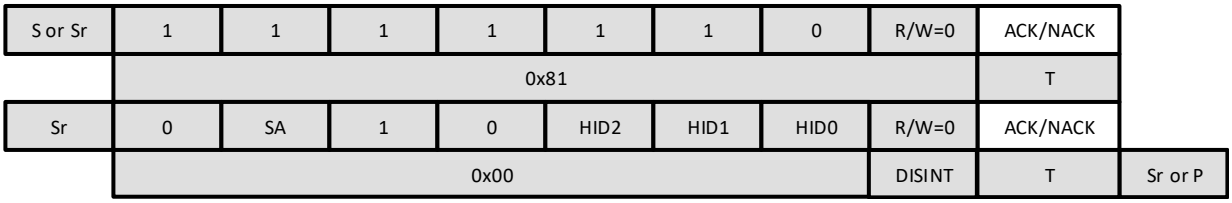



**Figure 7-30. DISEC CCC Direct**

The command may be issued either as a broadcast or as a direct command, as shown in Figure 7-31 and Figure 7-32, with PEC Enabled. In such a case the host controller shall append the PEC byte calculated on all bytes except the byte with 7'h7E and R/W=0 after the Start or Repeated Start.

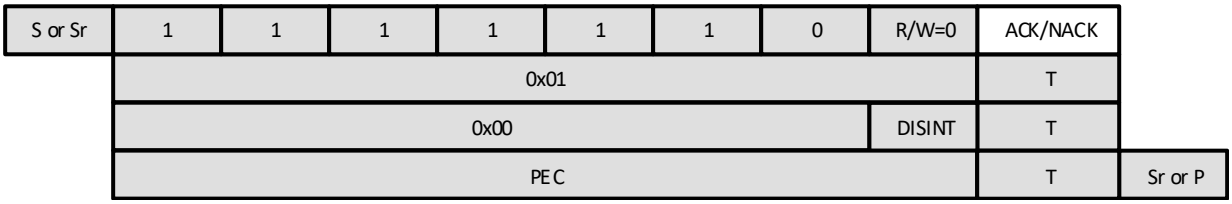



**Figure 7-31. DISEC CCC Broadcast with PEC Enabled**

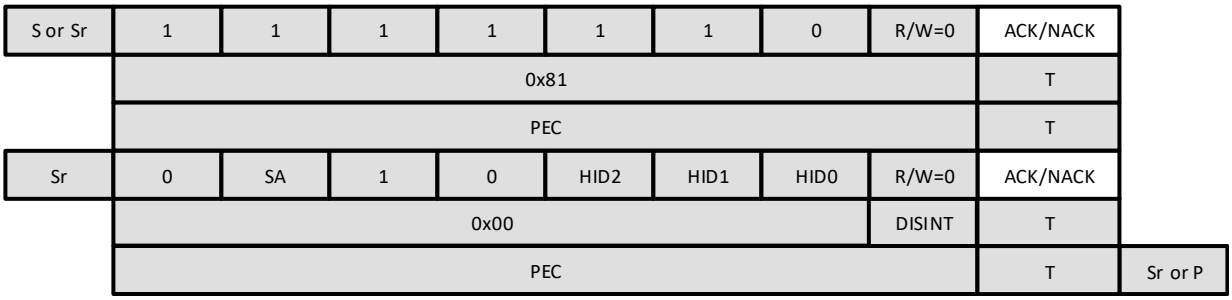



**Figure 7-32. DISEC CCC Direct with PEC Enabled**

**Note**

TMP139 NACKs the DISEC CCC if the previous transaction has a parity or PEC error and the host starts the transaction with a Repeated Start.

### *7.4.6.3 RSTDAA CCC*

When the RSTDAA CCC is issued by the host controller to the TMP139, it shall switch from I3C basic mode to $I^2C$ mode. The CCC takes effect after a Stop has been issued by the host controller. Once the RSTDAA is received, the TMP139 shall perform the following actions:

- Update the MR18 register bit INF_SEL as 1'b0 for $I^2C$ mode of operation.
- Update the MR18 register bit PEC_EN as 1'b0 to disable PEC, if it was previously enabled.
- Update the MR18 register bit PAR_DIS as 1'b0 to enable parity check, if it was previously disabled.
- Update the MR27 register bit IBI_ERROR_EN as 1'b0 to disable IBI, if it was previously enabled.

The command is always issues as a broadcast command as shown in Figure 7-33.

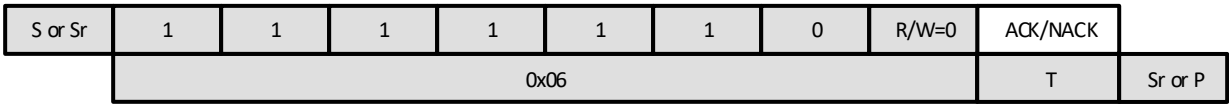



**Figure 7-33. RSTDAA CCC Broadcast**

The command may also be issued with PEC enabled, as shown in Figure 7-34. In such a case, the host controller shall append the PEC byte calculated on all bytes except the byte with 7'h7E and R/W=0 after the Start or Repeated Start.

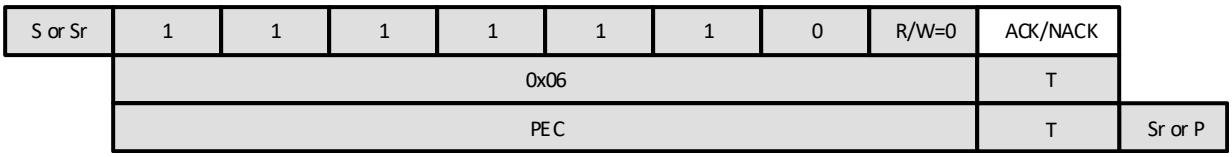



**Figure 7-34. RSTDAA CCC Broadcast with PEC**

**Note**

TMP139 NACKs the RSTDAA CCC if the previous transaction has a parity or PEC error and the host starts the transaction with a Repeated Start.

### *7.4.6.4 SETAASA CCC*

When the SETAASA CCC is issued by the host controller to the TMP139, it shall switch from $I^2C$ mode to I3C basic mode. The CCC takes effect after a Stop has been issued by the host controller. Once the SETAASA is received, the TMP139 shall set the MR18 register bit INF_SEL as 1'b1 for I3C basic mode of operation.

The CCC is always issues as a broadcast command, as shown in Figure 7-35, with no PEC bytes since it is applicable only during $I^2C$ mode.

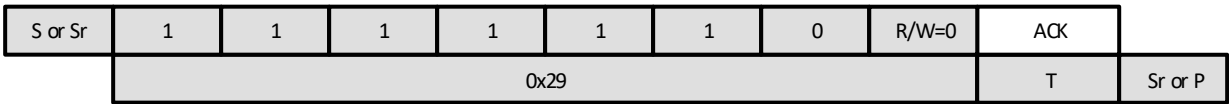



**Figure 7-35. SETAASA CCC**

**Note**

TMP139 NACKs the SETAASA CCC if the previous CCC transaction has a parity error and the host starts the transaction with a Repeated Start.

### *7.4.6.5 GETSTATUS CCC*

The GETSTATUS CCC is issued by the host controller to the TMP139 to get the status of any pending parity error, PEC error, or interrupt event. Once the GETSTATUS is received, the TMP139 shall not clear the status and host must issue additional transactions on the bus to clear the status flags individually or by writing 1'b1 to the MR27 register CLR_GLOBAL bit.

The command is issued only in direct mode as shown in Figure 7-36, when PEC is disabled, and in Figure 7-37, when PEC is enabled. In the latter case, the host controller shall append the PEC byte calculated on all bytes except the byte with 7'h7E and R/W=0 after the Start or Repeated Start. The TMP139 calculates the PEC on the data bytes that are sent to the host.

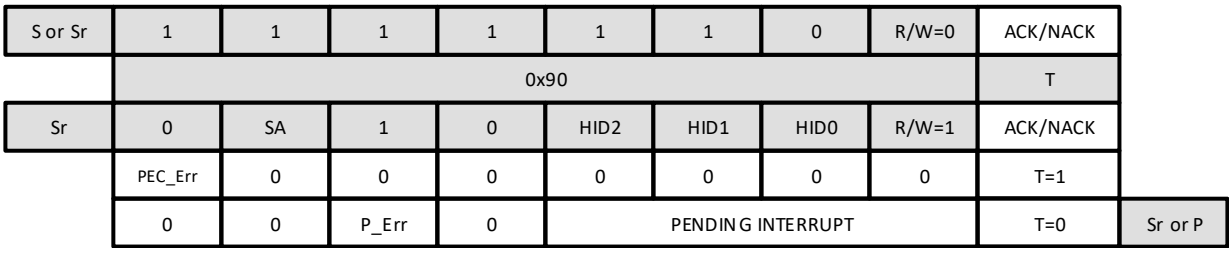



**Figure 7-36. GETSTATUS CCC Direct**

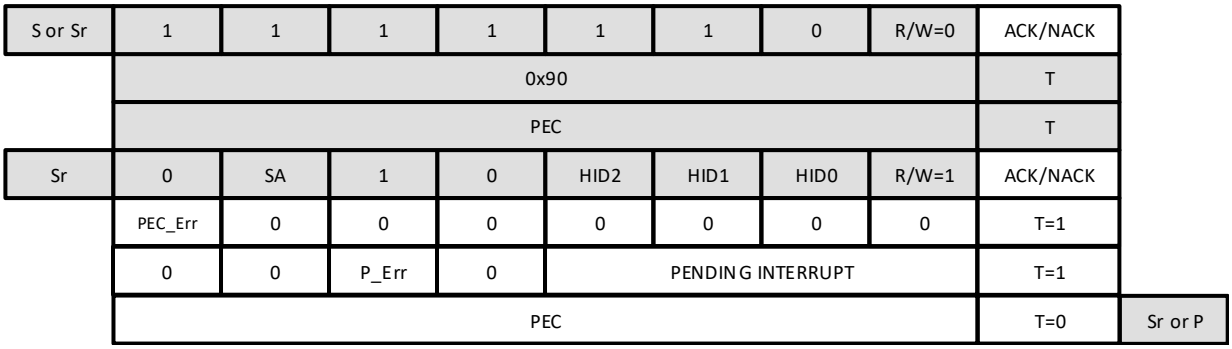



**Figure 7-37. GETSTATUS CCC Direct with PEC Enabled**

**Note**

TMP139 NACKs the GETSTATUS CCC if the previous transaction has a parity or PEC error and the host starts the transaction with a Repeated Start.

### *7.4.6.6 DEVCAP CCC*

The DEVCAP CCC is issued by the host controller to the TMP139, to get the optional device capabilities that are supported as given in Table 7-7.

The command is issued only in direct mode as shown in Figure 7-38, when PEC is disabled, and in Figure 7-39, when PEC is enabled. In the latter case the host controller shall append the PEC byte calculated on all bytes

except the byte with 7'h7E and R/W=0 after the Start or Repeated Start. The TMP139 calculates the PEC on the data bytes that are sent to the host.

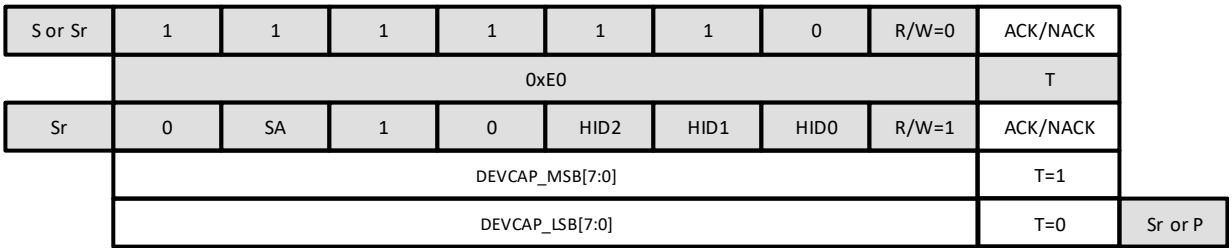



**Figure 7-38. DEVCAP CCC Direct**

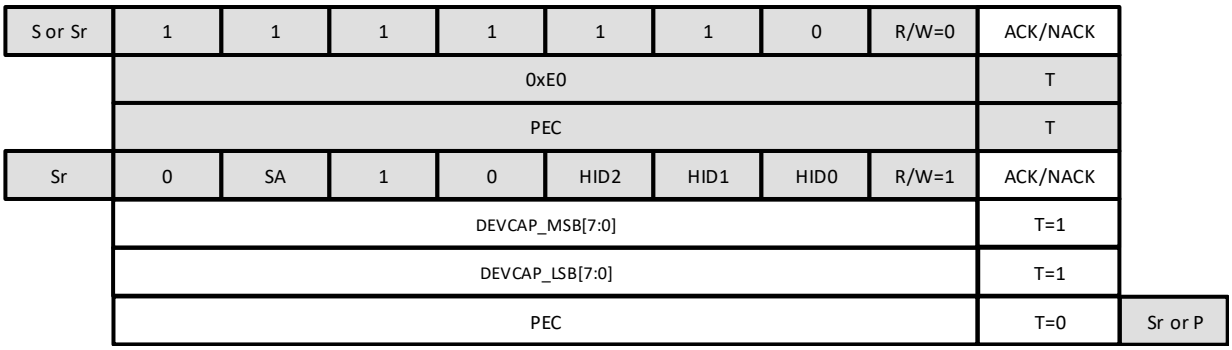



**Figure 7-39. DEVCAP CCC Direct with PEC**

**Table 7-7. DEVCAP Data Byte Description**

| Bit | Value | Comments |
|---|---|---|
| DEVCAP_MSB[7:3] | 00000 | Reserved |
| DEVCAP_MSB[2] | 1 | 0 = No support for Timer Based Reset<br>1 = Timer Based Reset supported |
| DEVCAP_MSB[1:0] | 00 | Reserved |
| DEVCAP_LSB[7:0] | 8'h00 | Reserved |

**Note**

TMP139 NACKs the DEVCAP CCC if the previous transaction has a parity or PEC error and the host starts the transaction with a Repeated Start.

### *7.4.6.7 SETHID CCC*

The SETHID CCC is issued by the host controller to the TMP139 to update the HID code of the device serial address. The CCC takes effect after a Stop has been issued by the host controller. Once the SETHID is received, the TMP139 shall update the MR7 register bits DEV_HID_CODE[2:0] to the value HID[2:0] as sent in the CCC data payload when Stop bus condition is sent by the host.

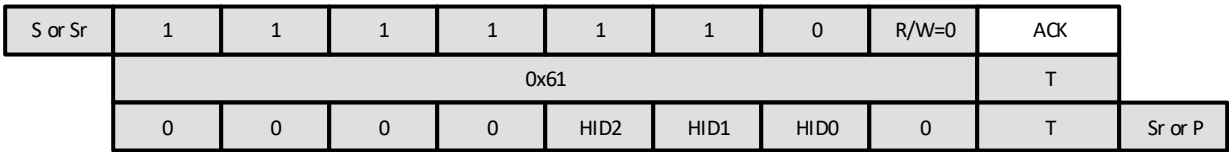



**Figure 7-40. SETHID CCC Broadcast**

**Note**

TMP139 NACKs the SETHID CCC if the previous transaction has a parity error and the host starts the transaction with a Repeated Start.

### *7.4.6.8 DEVCTRL CCC*

The DEVCTRL CCC is issued by the host controller for enable or disable operations that are common to devices on the bus and TMP139 shall recognize the DEVCTRL CCC.

The command is generally issued in broadcast mode, but may be issued as unicast or multicast mode as well. The host may issue the DEVCTRL CCC as a generic access with RegMod field set as 0 or for a specific register access with RegMod set as 1. When RegMod field is set to 0, Figure 7-41 shows the DEVCTRL CCC packet structure when PEC is disabled. Figure 7-42 shows the structure of the DEVCTRL CCC when RegMod field is set to 0 and PEC is enabled. In the latter case the host controller shall append the PEC byte calculated on all bytes except the byte with 7'h7E and R/W=0 after the Start or Repeated Start.

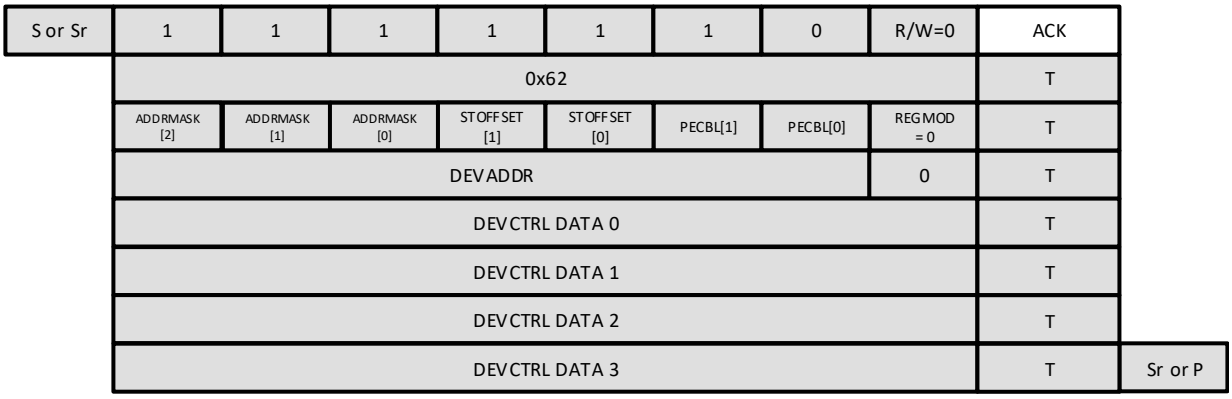



**Figure 7-41. DEVCTRL CCC With REGMOD = 0 and PEC Disabled**

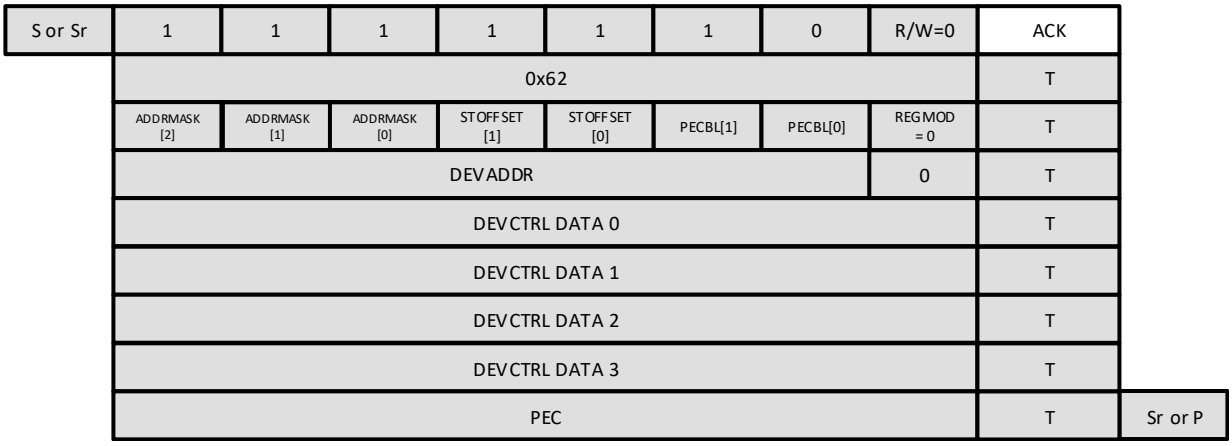



**Figure 7-42. DEVCTRL CCC With REGMOD = 0 and PEC Enabled**

When RegMod field is set to 1, Figure 7-43 shows the DEVCTRL CCC packet structure when PEC is disabled. Figure 7-44 shows the structure of the DEVCTRL CCC when RegMod field is set to 1 and PEC is enabled. In the latter case the host controller shall append the PEC byte calculated on all bytes except the byte with 7'h7E and

R/W = 0 after the Start or Repeated Start. If the CMD field indicates that there is only one byte to write, then the optional register data must not be sent by the host.

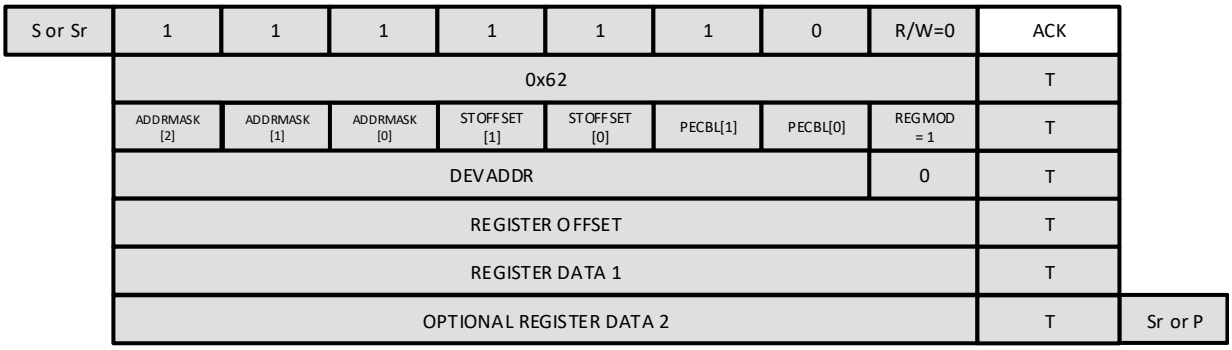



**Figure 7-43. DEVCTRL CCC With REGMOD = 1 and PEC Disabled**

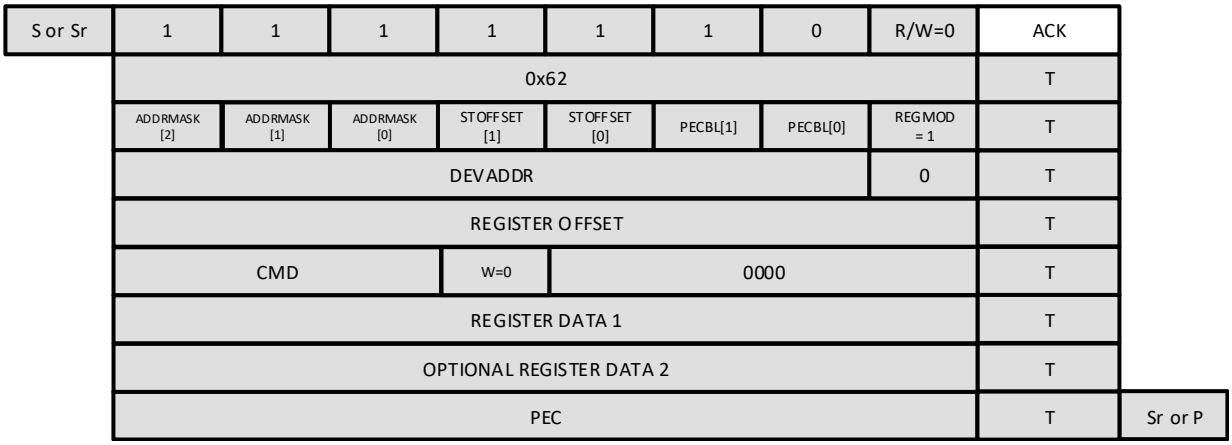



**Figure 7-44. DEVCTRL CCC With REGMOD = 1 and PEC Enabled**

**Note**

TMP139 NACKs the DEVCTRL CCC if the previous transaction has a parity or PEC error and the host starts the transaction with a Repeated Start.

The Table 7-8 describes the definition of the command fields.

**Table 7-8. DEVCTRL CCC Command Definitions**

| Field | Description | Values | Action |
|---|---|---|---|
| ADDRMASK[2:0] | Broadcast, multicast or unicast selection | 000 = Unicast command | TMP139 matches the DEVADDR[6:0] field with its serial address . |
| | | 011 = Multicast command | TMP139 matches the DEVADDR[6:3] field with its LID code in the serial address. |
| | | 111 = Broadcast command | TMP139 ignores the DEVADDR[6:0] and performs the required action. |
| STOFFSET[1:0] | Start offset byte | 00 = Byte 0 | TMP139 identifies which byte is the first byte out of DEVCTRL DATA 0, DEVCTRL DATA 1, DEVCTRL DATA 2 and DEVCTRL DATA 3 and updates its register accordingly.<br>This field is valid only when REGMOD = 0. |
| | | 01 = Byte 1 | |
| | | 10 = Byte 2 | |
| | | 11 = Byte 3 | |
| PECBL[1:0] | Identifies the burst length for PEC byte position | 00 = 1 Byte | TMP139 identifies the position of the PEC byte after the DEVCTRL DATA bytes are sent.<br>This field is valid only when REGMOD = 0 and PEC is enabled. |
| | | 01 = 2 Byte | |
| | | 10 = 3 Byte | |
| | | 11 = 4 Byte | |

**Table 7-8. DEVCTRL CCC Command Definitions (continued)**

| Field | Description | Values | Action |
|---|---|---|---|
| REGMOD | Identifies if it is a generic or specific register access | 0 = Generic Access | TMP139 understand the DEVCTRL DATA byte as generic data bytes described in Table 7-9 |
| | | 1 = Register Access | TMP139 understand the DEVCTRL DATA byte as specific register access bytes.<br>If PEC is disabled, the format used for specific register access is as per Figure 7-11.<br>If PEC is enabled, the format used for specific register access is as per Figure 7-13 |

**Table 7-9. Generic Data Byte Format**

| DEVCTRL DATA Bit | Function | Values | Action |
|---|---|---|---|
| DEVCTRL DATA 0 [7] | PEC Enable | 0 = Disable | MR18 register PEC_EN bit is updated |
| | | 1 = Enable | |
| DEVCTRL DATA 0 [6] | Parity Disable | 0 = Enable | MR18 register PAR_DIS bit is updated |
| | | 1 = Disable | |
| DEVCTRL DATA 0 [5:0] | Reserved | Reserved | |
| DEVCTRL DATA 1 [7:4] | Reserved | Reserved | |
| DEVCTRL DATA 1 [3] | Global IBI Clear | 0 = No action | MR27 register CLR_GLOBAL bit is updated |
| | | 1 = Clear all events and pending IBI | |
| DEVCTRL DATA 1 [2:0] | Reserved | Reserved | |
| DEVCTRL DATA 2 [7:0] | Reserved | Reserved | |
| DEVCTRL DATA 3 [7:0] | Reserved | Reserved | |

**Note**

TMP139 NACKs the DEVCTRL CCC if the previous transaction has a parity or PEC error and the host starts the transaction with a Repeated Start.

### 7.4.7 I/O Operation

The device comes up in the $I^2C$ mode of operation with an open-drain I/O for its interface. However, when the device is in I3C mode, the I/O may be either open drain or push pull. The dynamic switching between open-drain and push-pull mode is primarily meant to support In Band Interrupts (IBI). Table 7-10 describes the different modes of operation for the I/O for each cycle.

**Table 7-10. TMP139 Dynamic I/O Operation for I3C Mode**

| OPERATION | OPEN-DRAIN MODE | PUSH-PULL MODE |
|---|---|---|
| Start + Device Address | Yes | No |
| Start + 7'h7E IBI Header Byte | Yes | No |
| REPEAT Start + Device Address | No | Yes |
| REPEAT Start + 7'h7E IBI Header Byte | No | Yes |
| CCC Bytes (after 7'h7E+R/W=0+ACK) | No | Yes |
| STOP | No | Yes |
| ACK/NACK Responses | Yes | No |
| Interrupt Request by TMP139 + Device Address | Yes | No |
| Command and address operations | No | Yes |
| IBI Payload | No | Yes |
| Write Data, T-bit sequence | No | Yes |
| Read Data, T-bit sequence | No | Yes |

**Table 7-10. TMP139 Dynamic I/O Operation for I3C Mode (continued)**

| OPERATION | OPEN-DRAIN MODE | PUSH-PULL MODE |
|---|---|---|
| PEC, T-bit sequence | No | Yes |

### 7.4.8 Timing Diagrams

The TMP139 is a $I^2C$ and I3C interface-compatible device. Figure 6-1 to Figure 6-3 describe the various bus conditions that are supported on the bus. The following lists the definitions for the bus conditions:

1. **Bus Idle:** Both SDA and SCL lines remain high after a Stop condition.
2. **Start (S) condition:** A change in the state of the SDA line from high to low, when the SCL is high defines a Start condition. The Start condition is preceded by a bus idle.
3. **Stop (P) condition:** A change in the state of the SDA line from low to high, when the SCL is high defines a Stop condition.
4. **Repeated Start ($S_R$) condition:** A change in the state of the SDA line from high to low, when the SCL is high and is preceded by a data transfer defines a Repeated Start condition.
5. **Data Transfer:** The number of data bytes transferred between a Start and Stop condition and determined by the host or device.
6. **Acknowledge:** Each receiving device, when addressed, is obliged to generate an acknowledge (ACK) bit during device address and host to device write transfer. A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable low during the high period of the acknowledge clock pulse. On a host receive, the termination of the data transfer can be signaled by the host generating a Not-Acknowledge (NAK) on the last byte that is transmitted by the target device. This behavior is as per $I^2C$ mode of operation.

   During I3C mode of operation. each receiving device shall only acknowledge its device address. Additionally, the host shall acknowledge the device address during a successful IBI address arbitration.
7. **T-Bit:** The T-bit is only applicable in I3C mode of operation or when the host sends a common command code (CCC) during $I^2C$ mode of operation. The T-bit contains the parity information when the host writes to the targeted device(s). During a read, if the T-bit is sampled as 1 on the rising edge of the $9^{th}$ clock, the bit indicates a continuation of a read by the device. If a host wants to terminate the read, then the host can activate the pullup while the device drives the line high as shown in Figure 6-2. When the device stops driving the line and tri-states its output, the pull up keeps the line high momentarily before the host claims control of the bus to generate a Repeated Start and Stop to end the read. If the host can accept more data from the device, the host must not drive the line. The device samples the SDA on the falling edge of the $9^{th}$ clock, and if the T-bit is sampled as 1, the device resumes driving the SDA for the next byte. During a read, if the T-bit is sampled as 0 on the rising edge of the $9^{th}$ clock, the bit is used to indicate a termination of the read by the device as shown in Figure 6-3. The host shall also drive the SDA low, such that when the device stops driving the line and tri-states its output, the host has control of the bus to generate a Stop to end the read.

## 7.5 Programming

This section describes the programming model for specific operations of the TMP139.

### 7.5.1 Enabling Interrupt Mechanism

IBI can only be enabled in I3C basic mode. Figure 7-45 shows the programming model the host controller must follow to correctly enable the IBI for the TMP139.

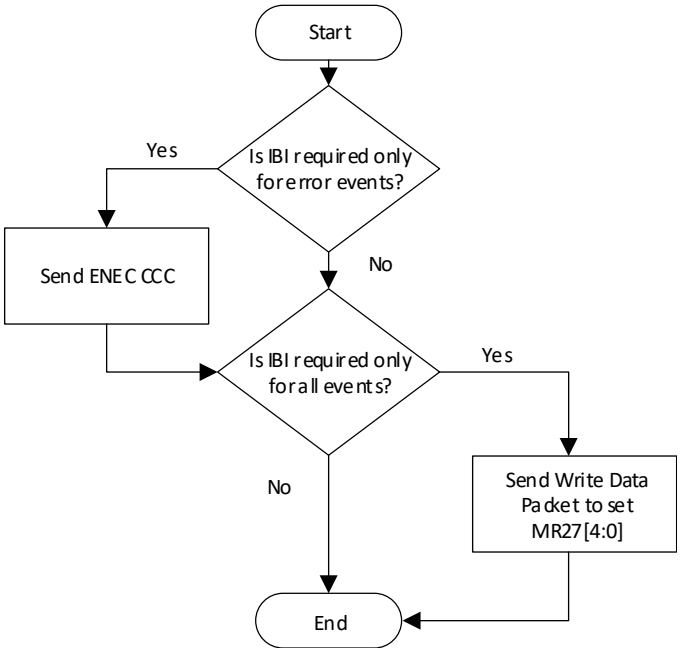



**Figure 7-45. Interrupt Enable Flowchart**

### 7.5.2 Clearing Interrupt

While IBI can be generated in I3C basic mode, the TMP139 shall update the status bit for different events (other than PEC error) even in $I^2C$ mode. Figure 7-46 shows the programming model for the host controller to clear an IBI in I3C basic mode. In $I^2C$ mode, the host controller can poll the TMP139 using register data read as already described in Section 7.4.3.2.

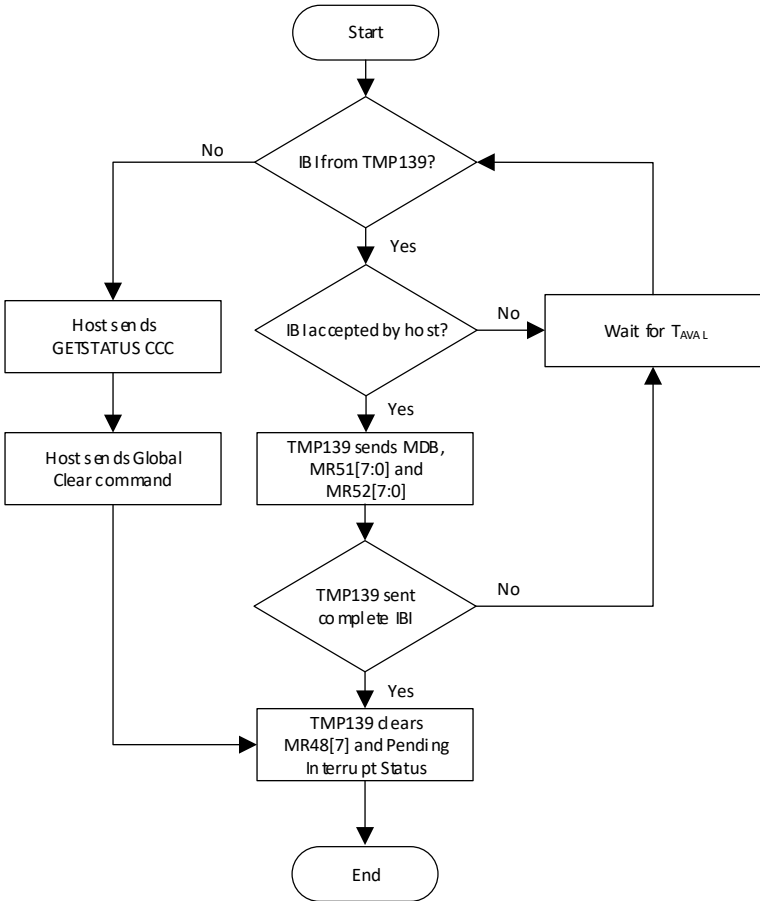



**Figure 7-46. Interrupt Clear Flowchart**

## 7.6 Register Map

**Table 7-11. TMP139 Register Map**

| ADDRESS | TYPE | RESET | REGISTER NAME | REGISTER DESCRIPTION | SECTION |
|---|---|---|---|---|---|
| 00h | R | 51h | MR0 | Device Type; Most Significant Byte | Go |
| 01h | R | 10h | MR1 | Device Type: Least Significant Byte | Go |
| 02h | R | 02h | MR2 | Device Revision | Go |
| 03h | R | 80h | MR3 | Vendor ID Byte 0 | Go |
| 04h | R | 97h | MR4 | Vendor ID Byte 1 | Go |
| 07h | RW | 0Eh | MR7 | Device Configuration - HID | Go |
| 12h | RW | 00h | MR18 | Device Configuration | Go |
| 13h | W1C | 00h | MR19 | Clear Register MR51 Temperature Status Command | Go |
| 14h | W1C | 00h | MR20 | Clear Register MR52 Error Status Command | Go |
| 1Ah | RW | 00h | MR26 | TS Configuration | Go |
| 1Bh | RW | 00h | MR27 | Interrupt Configurations | Go |
| 1Ch | RW | 70h | MR28 | TS Temp High Limit Configuration - Low Byte | Go |
| 1Dh | RW | 03h | MR29 | TS Temp High Limit Configuration - High Byte | Go |
| 1Eh | RW | 00h | MR30 | TS Temp Low Limit Configuration - Low Byte | Go |
| 1Fh | RW | 00h | MR31 | TS Temp Low Limit Configuration - High Byte | Go |
| 20h | RW | 50h | MR32 | TS Critical Temp High Limit Configuration - Low Byte | Go |
| 21h | RW | 05h | MR33 | TS Critical Temp High Limit Configuration - High Byte | Go |
| 22h | RW | 00h | MR34 | TS Critical Temp Low Limit Configuration - Low Byte | Go |
| 23h | RW | 00h | MR35 | TS Critical Temp Low Limit Configuration - High Byte | Go |
| 30h | R | 00h | MR48 | Device Status | Go |
| 31h | R | 00h | MR49 | TS Current Sensed Temperature - Low Byte | Go |
| 32h | R | 00h | MR50 | TS Current Sensed Temperature - High Byte | Go |
| 33h | R | 00h | MR51 | TS Temperature Status | Go |
| 34h | R | 00h | MR52 | Miscellaneous Error Status | Go |

**Table 7-12. Register Section Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| RC | R<br>C | Read<br>to Clear |
| RV | RV | Reserved for future expansion |
| Write Type | | |
| W | W | Write |
| W1C | W<br>1C | W<br>1 to clear |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |

### 7.6.1 MR0: Device Type, Most Significant Byte (address = 00h) [reset = 51h]

Return to Register Map.

**Figure 7-47. MR0: Device Type Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MSB_DEV_TYPE[7:0] | | | | | | | |
| R-51h | | | | | | | |

**Table 7-13. MR0: Device Type Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:0 | MSB_DEV_TYPE[7:0] | R | 51h | Device type most significant byte. Used in conjunction with MR1 register. |

### 7.6.2 MR1: Device Type, Least Significant Byte (address = 01h) [reset = 10h]

Return to Register Map.

**Figure 7-48. MR1: Device Type Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| LSB_DEV_TYPE[7:0] | | | | | | | |
| R-10h | | | | | | | |

**Table 7-14. MR1: Device Type Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:0 | LSB_DEV_TYPE[7:0] | R | 10h | Device type least significant byte. Used in conjunction with MR0 register.<br>Indicates a Grade-B temperature sensor |

### 7.6.3 MR2: Device Revision (address = 02h) [reset = 02h]

Return to Register Map.

**Figure 7-49. MR2: Device Revision Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | DEV_REV_MAJOR[1:0] | | DEV_REV_MINOR[2:0] | | | Reserved |
| R-00 | | R-00 | | R-001 | | | R-0 |

**Table 7-15. MR2: Device Revision Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:6 | Reserved | R | 00 | Reserved |
| 5:4 | DEV_REV_MAJOR[1:0] | R | 00 | Indicates the major revision number |
| 3:1 | DEV_REV_MINOR[2:0] | R | 001 | Indicates the minor revision number |
| 0 | Reserved | R | 0 | Reserved |

### 7.6.4 MR3: Vendor ID Byte 0 (address = 03h) [reset = 80h]

Return to Register Map.

**Figure 7-50. MR3: Vendor ID Byte 0 Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VENDOR_ID_BYTE0[7:0] | | | | | | | |
| R-80h | | | | | | | |

**Table 7-16. MR3: Vendor ID Byte 0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:6 | VENDOR_ID_BYTE0[7:0] | R | 80h | Indicates the lower byte of the Vendor ID. |

### 7.6.5 MR4: Vendor ID Byte 1 (address = 04h) [reset = 97h]

Return to Register Map.

**Figure 7-51. MR4: Vendor ID Byte 1 Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VENDOR_ID_BYTE1[7:0] | | | | | | | |
| R-97h | | | | | | | |

**Table 7-17. MR4: Vendor ID Byte 1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:6 | VENDOR_ID_BYTE1[7:0] | R | 97h | Indicates the upper byte of the Vendor ID. |

### 7.6.6 MR7: Device Configuration - HID (address = 04h) [reset = 0Eh]

The MR7 register reads the HID configured by the host controller. This register can only be updated by the SETHID CCC when device is in $I^2C$, by the RSTDAA when the device is in I3C mode, or by a bus reset.

Return to Register Map.

**Figure 7-52. MR7: Device Configuration - HID Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | DEV_HID_CODE[2:0] | | | Reserved |
| R-0h | | | | RW-111 | | | R-0 |

**Table 7-18. MR7: Device Configuration - HID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:4 | Reserved | R | 0h | Reserved |
| 3:1 | DEV_HID_CODE[2:0] | RW | 111 | Device HID Code. The TMP139 device responds to unique 7-bit address as formed by a 4-bit LID code as Table 7-4 and 3-bit HID code as configured in this register.[1] |
| 0 | Reserved | R | 0 | Reserved |

1. This register is updated only when SETHID CCC is sent to the TMP139 or when the device goes through a bus reset sequence.

**Note**

Any host transaction which results in write or update to MR7 register must be immediately followed by a Stop condition. A Repeated Start may result in unpredictable behavior.

### 7.6.7 MR18: Device Configuration (address = 12h) [reset = 00h]

The MR18 register is used to configure the device features. In I3C mode, it allows the PEC to be enabled and Parity (T-bit) to be disabled. It also controls the default read address mode for both $I^2C$ and I3C bus operations. The burst length for the PEC byte is allowed only in I3C mode and the host controller must not update the bit in the $I^2C$ mode of operation.

Return to Register Map.

**Figure 7-53. MR18: Device Configuration Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PEC_EN | PAR_DIS | INF_SEL | DEF_RD_ADDR_POINT_EN | DEF_RD_ADDR_POINT_Start[1:0] | | DEF_RD_ADDR_POINT_BL | Reserved |
| RW-0 | RW-0 | R-0 | RW-0 | RW-0 | | RW-0 | R-0 |

**Table 7-19. MR18: Device Configuration Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | PEC_EN | RW | 0 | PEC enable[1]<br>0 = PEC is disabled<br>1 = PEC is enabled |
| 6 | PAR_DIS | RW | 0 | Parity (T-bit) disable[1]<br>0 = Parity or T-bit is enabled<br>1 = Parity or T-bit is disabled |
| 5 | INF_SEL | R | 0 | Interface selection<br>0 = $I^2C$ protocol (maximum speed of 1 MHz)<br>1 = I3C basic protocol |
| 4 | DEF_RD_ADDR_POINT_EN | RW | 0 | Default read address pointer enable<br>0 = Disable default read address pointer (address pointer is set by host)<br>1 = Enable default read address pointer (address selected by MR7 register, DEF_RD_ADDR_POINT_Start[1:0] bits |
| 3:2 | DEF_RD_ADDR_POINT_Start[1:0] | RW | 00 | Default read address pointer starting address[2]<br>00 = MR49 register<br>01 = Reserved<br>10 = Reserved<br>11 = Reserved |
| 1 | DEF_RD_ADDR_POINT_BL | RW | 0 | Burst length for read pointer address for PEC calculation<br>0 = 2 bytes<br>1 = 4 bytes |
| 0 | Reserved | R | 0 | Reserved |

1. PEC enable and parity disable are automatically updated when RSTDAA CCC is issued on the bus or a bus reset sequence is applied.
2. Setting any of the reserved values shall result in unpredictable behavior from TMP139.

**Note**

Any host transaction which results in write or update to MR18 register must be immediately followed by a Stop condition. A Repeated Start may result in unpredictable behavior.

### 7.6.8 MR19: Clear MR51 Temperature Status Command (address = 13h) [reset = 00h]

The MR19 register is written by the host to clear status for the temperature comparison after the most recent conversion.

Return to Register Map.

**Figure 7-54. MR19: Clear MR51 Temperature Status Command Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CLR_TS_CRIT_LOW | CLR_TS_CRIT_HIGH | CLR_TS_LOW | CLR_TS_HIGH |
| R-0h | | | | R0-W1C | R0-W1C | R0-W1C | R0-W1C |

**Table 7-20. MR19: Clear MR51 Temperature Status Command Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:4 | Reserved | R | 0h | Reserved |
| 3 | CLR_TS_CRIT_LOW | R0-W1C | 0 | Clear temperature sensor critical low status<br>1 = Write '1' to clear MR51 TS_CRIT_LOW_STATUS bit<br>Writing a '0' has no effect on MR51 TS_CRIT_LOW_STATUS bit |
| 2 | CLR_TS_CRIT_HIGH | R0-W1C | 0 | Clear temperature sensor critical high status<br>1 = Write '1' to clear MR51 TS_CRIT_HIGH_STATUS bit<br>Writing a '0' has no effect on MR51 TS_CRIT_HIGH_STATUS bit |
| 1 | CLR_TS_LOW | R0-W1C | 0 | Clear temperature sensor low status<br>1 = Write '1' to clear MR51 TS_LOW_STATUS bit<br>Writing a '0' has no effect on MR51 TS_LOW_STATUS bit |
| 0 | CLR_TS_HIGH | R0-W1C | 0 | Clear temperature sensor high status<br>1 = Write '1' to clear MR51 TS_HIGH_STATUS bit<br>Writing a '0' has no effect on MR51 TS_HIGH_STATUS bit |

### 7.6.9 MR20: Clear MR52 Error Status Command (address = 14h) [reset = 00h]

The MR20 register is written by the host to clear error condition when the PEC checksum is incorrect or when the last write from the host results in a parity error in the T-bit. This register is valid in I3C mode only.

Return to Register Map.

**Figure 7-55. MR20: Clear MR52 Error Status Command Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CLR_PEC_ERROR | CLR_PAR_ERROR |
| R-00h | | | | | | W1C | W1C |

**Table 7-21. MR20: Clear MR52 Error Status Command Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:2 | Reserved | R | 00h | Reserved |
| 1 | CLR_PEC_ERROR | R0-W1C | 0 | Clear packet error status<br>1 = Write '1' to clear MR52 PEC_ERROR_STATUS bit<br>Writing a '0' has no effect on MR52 PEC_ERROR_STATUS bit |
| 0 | CLR_PAR_ERROR | R0-W1C | 0 | Clear parity error status<br>1 = Write '1' to clear MR52 PEC_ERROR_STATUS bit<br>Writing a '0' has no effect on MR52 PAR_ERROR_STATUS bit |

### 7.6.10 MR26: TS Configuration (address = 1Ah) [reset = 00h]

The MR26 register may be used by the host to disable the temperature sensor. The device will stop temperature conversion or, if there is an ongoing conversion when the bit is set, then it shall complete the current conversion and then disable the temperature sensor.

Return to Register Map.

**Figure 7-56. MR26: Temperature Sensor Configuration Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | DIS_TS |
| R-00h | | | | | | | RW-0 |

**Table 7-22. MR26: Temperature Sensor Configuration Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:1 | Reserved | R | 00h | Reserved |
| 0 | DIS_TS | RW | 0 | Disable temperature sensor<br>0 = Enable temperature sensor.<br>1 = Disable temperature sensor. |

### 7.6.11 MR27: Interrupt Configuration (address = 1Bh) [reset = 00h]

Return to Register Map.

**Figure 7-57. MR27: Interrupt Configuration Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CLR_GLOBAL | Reserved | | IBI_ERROR_EN | IBI_TS_CRIT_LOW_EN | IBI_TS_CRIT_HIGH_EN | IBI_TS_LOW_EN | IBI_TS_HIGH_EN |
| W1C | R-00 | | R-0 | RW-0 | RW-0 | RW-0 | RW-0 |

**Table 7-23. MR27: Interrupt Configuration Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | CLR_GLOBAL | R0-W1C | 0 | Global clear event status and In Band Interrupt (IBI) status<br>1 = Write '1' to clear the registers MR48, MR51 and MR52.<br>Writing a '0' has no effect on registers MR48, MR51 and MR52. |
| 6:5 | Reserved | R | 00 | Reserved |
| 4 | IBI_ERROR_EN | R | 0 | In band interrupt (IBI) enable for MR52 error log.[1]<br>0 = Disable. Errors logged in MR52 register bits do not generate an IBI to host.<br>1 = Enable. Errors logged in MR52 register bits generate an IBI to host. |
| 3 | IBI_TS_CRIT_LOW_EN | RW | 0 | In band interrupt (IBI) enable for temperature sensor critical low.<br>0 = Disable. MR51 register TS_CRIT_LOW_STATUS bit does not generate an IBI to host.<br>1 = Enable. MR51 register TS_CRIT_LOW_STATUS bit generates an IBI to host. |
| 2 | IBI_TS_CRIT_HIGH_EN | RW | 0 | In band interrupt (IBI) enable for temperature sensor critical high.<br>0 = Disable. MR51 register TS_CRIT_HIGH_STATUS bit does not generate an IBI to host.<br>1 = Enable. MR51 register TS_CRIT_HIGH_STATUS bit generates an IBI to host. |
| 1 | IBI_TS_LOW_EN | RW | 0 | In band interrupt (IBI) enable for temperature sensor low.<br>0 = Disable. MR51 register TS_LOW_STATUS bit does not generate an IBI to host.<br>1 = Enable. MR51 register TS_LOW_STATUS bit generates an IBI to host. |

**Table 7-23. MR27: Interrupt Configuration Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | IBI_TS_HIGH_EN | RW | 0 | In band interrupt (IBI) enable for temperature sensor high.<br>0 = Disable. MR51 register TS_HIGH_STATUS bit does not generate an IBI to host.<br>1 = Enable. MR51 register TS_HIGH_STATUS bit generates an IBI to host. |

1. IBI_ERROR_EN can only be updated by ENEC CCC, DISEC CCC, RSTDAA CCC or bus reset sequence. A direct write to the register or through DEVCTRL CCC shall not update the bit and may lead to unpredictable behavior.

### 7.6.12 MR28: Temperature Sensor High Limit-Low Byte Configuration (address = 1Ch) [reset = 70h]

The status flag for temperature high limit is set when the result of the temperature conversion is greater than the programmed value in the MR29 and MR28 registers. The application must ensure that the critical temperature high limit registers must have a value greater than the temperature high limit registers.

Return to Register Map.

**Figure 7-58. MR28: Temperature Sensor High Limit-Low Byte Configuration Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TS_HIGH_LIMIT_LOW[7:0] | | | | | | | |
| RW-70h | | | | | | R-0 | R-0 |

**Table 7-24. MR28: Temperature Sensor High Limit-Low Byte Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:0 | TS_HIGH_LIMIT_LOW[7:0] | RW | 70h | Low byte of the high limit temperature for the thermal sensor.[1] MR29 and MR28 together define the high limit for the thermal sensor. |

1. The bits marked as R-0, shall not be updated when host writes 1 and shall read as 0.

### 7.6.13 MR29: Temperature Sensor High Limit-High Byte Configuration (address = 1Dh) [reset = 03h]

The status flag for temperature high limit is set when the result of the temperature conversion is greater than the programmed value in the MR29 and MR28 registers. The application must ensure that the critical temperature high limit registers must have a value greater than the temperature high limit registers.

Return to Register Map.

**Figure 7-59. MR29: Temperature Sensor High Limit-High Byte Configuration Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TS_HIGH_LIMIT_HIGH[7:0] | | | | | | | |
| R-0 | R-0 | R-0 | RW-03h | | | | |

**Table 7-25. MR29: Temperature Sensor High Limit-High Byte Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:0 | TS_HIGH_LIMIT_HIGH[7:0] | RW | 03h | High byte of the high limit temperature for the thermal sensor.[1] MR29 and MR28 together define the high limit for the thermal sensor. |

1. The bits marked as R-0, shall not be updated when host writes 1 and shall read as 0.

### 7.6.14 MR30: Temperature Sensor Low Limit-Low Byte Configuration (address = 1Eh) [reset = 00h]

The status flag for critical low limit is set when the result of the temperature conversion is less than the programmed value in the MR31 and MR30 registers. The application must ensure that the critical temperature low limit registers must have a value lower than the temperature low limit registers.

Return to Register Map.

**Figure 7-60. MR30: Temperature Sensor Low Limit-Low Byte Configuration Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TS_LOW_LIMIT_LOW[7:0] | | | | | | | |
| RW-00h | | | | | | R-0 | R-0 |

**Table 7-26. MR30: Temperature Sensor Low Limit-Low Byte Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:0 | TS_LOW_LIMIT_LOW[7:0] | RW | 00h | Low byte of the low limit temperature for the thermal sensor.[1] MR31 and MR30 together define the low limit for the thermal sensor. |

1. The bits marked as R-0, shall not be updated when host writes 1 and shall read as 0.

### 7.6.15 MR31: Temperature Sensor Low Limit-High Byte Configuration (address = 1Fh) [reset = 00h]

The status flag for critical low limit is set when the result of the temperature conversion is less than the programmed value in the MR31 and MR30 registers. The application must ensure that the critical temperature low limit registers must have a value lower than the temperature low limit registers.

Return to Register Map.

**Figure 7-61. MR31: Temperature Sensor Low Limit-High Byte Configuration Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TS_LOW_LIMIT_HIGH[7:0] | | | | | | | |
| R-0 | R-0 | R-0 | RW-00h | | | | |

**Table 7-27. MR31: Temperature Sensor Low Limit-High Byte Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:0 | TS_LOW_LIMIT_HIGH[7:0] | RW | 00h | High byte of the low limit temperature for the thermal sensor.[1] MR31 and MR30 together define the low limit for the thermal sensor. |

1. The bits marked as R-0, shall not be updated when host writes 1 and shall read as 0.

### 7.6.16 MR32: Temperature Sensor Critical High Temperature Limit-Low Byte Configuration (address = 20h) [reset = 50h]

The status flag for critical temperature high limit is set when the result of the temperature conversion is greater than the programmed value in the MR33 and MR32 registers. The application must ensure that the critical temperature high limit registers must have a value greater than the temperature high limit registers.

Return to Register Map.

**Figure 7-62. MR32: Temperature Sensor Critical Temperature High Limit-Low Byte Configuration Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TS_CRIT_HIGH_LIMIT_LOW[7:0] | | | | | | | |
| RW-50h | | | | | | R-0 | R-0 |

**Table 7-28. MR32: Temperature Sensor Critical Temperature High Limit-Low Byte Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:0 | TS_CRIT_HIGH_LIMIT_LOW[7:0] | RW | 50h | Low byte of the critical high limit temperature for the thermal sensor.[1]<br>MR33 and MR32 together define the critical high limit temperature for the thermal sensor. |

1. The bits marked as R-0, shall not be updated when host writes 1 and shall read as 0.

### 7.6.17 MR33: Temperature Sensor Critical Temperature High Limit-High Byte Configuration (address = 21h) [reset = 05h]

The status flag for critical temperature high limit is set when the result of the temperature conversion is greater than the programmed value in the MR33 and MR32 registers. The application must ensure that the critical temperature high limit registers must have a value greater than the temperature high limit registers.

Return to Register Map.

**Figure 7-63. MR33: Temperature Sensor Critical Temperature High Limit-High Byte Configuration Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TS_CRIT_HIGH_LIMIT_HIGH[7:0] | | | | | | | |
| R-0 | R-0 | R-0 | RW-05h | | | | |

**Table 7-29. MR33: Temperature Sensor Critical Temperature High Limit-High Byte Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:0 | TS_CRIT_HIGH_LIMIT_HIGH[7:0] | RW | 05h | High byte of the critical high limit temperature for the thermal sensor.[1]<br>MR33 and MR32 together define the critical high limit temperature for the thermal sensor. |

1. The bits marked as R-0, shall not be updated when host writes 1 and shall read as 0.

### 7.6.18 MR34: Temperature Sensor Critical Temperature Low Limit-Low Byte Configuration (address = 22h) [reset = 00h]

The status flag for critical temperature low limit is set when the result of the temperature conversion is less than the programmed value in the MR35 and MR34 registers. The application must ensure that the critical temperature low limit registers must have a value lesser than the temperature low limit registers.

Return to Register Map.

**Figure 7-64. MR34: Temperature Sensor Critical Temperature Low Limit-Low Byte Configuration Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TS_CRIT_LOW_LIMIT_LOW[7:0] | | | | | | | |
| RW-00h | | | | | | R-0 | R-0 |

**Table 7-30. MR34: Temperature Sensor Critical Temperature Low Limit-Low Byte Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:0 | TS_CRIT_LOW_LIMIT_LOW[7:0] | RW | 00h | Low byte of the critical low limit temperature for the thermal sensor.[1]<br>MR35 and MR34 together define the critical low limit temperature for the thermal sensor. |

1. The bits marked as R-0, shall not be updated when host writes 1 and shall read as 0.

### 7.6.19 MR35: Temperature Sensor Critical Temperature Low Limit-High Byte Configuration (address = 23h) [reset = 00h]

The status flag for critical temperature low limit is set when the result of the temperature conversion is less than the programmed value in the MR35 and MR34 registers. The application must ensure that the critical temperature low limit registers must have a value lesser than the temperature low limit registers.

Return to Register Map.

**Figure 7-65. MR35: Temperature Sensor Critical Temperature Low Limit-High Byte Configuration Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TS_CRIT_LOW_LIMIT_HIGH[7:0] | | | | | | | |
| R-0 | R-0 | R-0 | RW-00h | | | | |

**Table 7-31. MR35: Temperature Sensor Critical Temperature Low Limit-High Byte Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:0 | TS_CRIT_LOW_LIMIT_HIGH[7:0] | RW | 00h | High byte of the critical low limit temperature for the thermal sensor.[1]<br>MR35 and MR34 together define the critical low limit temperature for the thermal sensor. |

1. The bits marked as R-0, shall not be updated when host writes 1 and shall read as 0.

### 7.6.20 MR48: Device Status (address = 30h) [reset = 00h]

The MR48 register provides the status of the IBI when the TMP139 is in I3C mode.

Return to Register Map.

**Figure 7-66. MR48: Device Status Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| IBI_STATUS | Reserved | | | | | | |
| R-0 | R-00h | | | | | | |

**Table 7-32. MR48: Device Status Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | IBI_STATUS | R | 0 | Device event In Band Interrupt (IBI) status.<br>0 = No pending IBI.<br>1 = Pending IBI. |
| 6:0 | Reserved | R | 00h | Reserved |

### 7.6.21 MR49: Current Sensed Temperature Low Byte (address = 31h) [reset = 00h]

The MR49 register, stores the lower 8-bits of the temperature output from the most recent conversion.

Return to Register Map.

**Figure 7-67. MR49: Current Sensed Temperature Low Byte Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TS_SENSE_LOW[7:0] | | | | | | | |
| R-00h | | | | | | | |

**Table 7-33. MR49: Current Sensed Temperature Low Byte Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:0 | TS_SENSE_LOW[7:0] | R | 00h | Low byte of the of the current temperature returned after most recent conversion by the thermal sensor.<br>MR50 and MR49 together provide the temperature returned after the most recent conversion. |

### 7.6.22 MR50: Current Sensed Temperature High Byte (address = 32h) [reset = 00h]

The MR50 register, stores the upper 8-bits of the temperature output from the most recent conversion..

Return to Register Map.

**Figure 7-68. MR50: Current Sensed Temperature High Byte Configuration Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TS_SENSE_HIGH[7:0] | | | | | | | |
| R-00h | | | | | | | |

**Table 7-34. MR50: Current Sensed Temperature High Byte Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:0 | TS_SENSE_HIGH[7:0] | R | 00h | High byte of the of the current temperature returned after most recent conversion by the thermal sensor.<br>MR49 and MR50 together provide the temperature returned after the most recent conversion. |

### 7.6.23 MR51: Temperature Status (address = 33h) [reset = 00h]

The MR51 registers stores the status from comparison of the most recent conversion temperature output to each of the four threshold levels defined in MR28 to MR35.

Return to Register Map.

**Figure 7-69. MR51: Temperature Status Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | TS_CRIT_LOW_STATUS | TS_CRIT_HIGH_STATUS | TS_LOW_STATUS | TS_HIGH_STATUS |
| R-0h | | | | R-0 | R-0 | R-0 | R-0 |

**Table 7-35. MR51: Temperature Status Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6:5 | Reserved | R | 00 | Reserved |
| 3 | TS_CRIT_LOW_STATUS | R | 0 | Temperature sensor critical low status.<br>0 = Temperature is above the limit set in registers MR35 and MR34.<br>1 = Temperature is below the limit set in registers MR35 and MR34. |
| 2 | TS_CRIT_HIGH_STATUS | R | 0 | Temperature sensor critical high status.<br>0 = Temperature is below the limit set in registers MR33 and MR32.<br>1 = Temperature is above the limit set in registers MR33 and MR32. |
| 1 | TS_LOW_STATUS | R | 0 | Temperature sensor low status.<br>0 = Temperature is above the limit set in registers MR31 and MR30.<br>1 = Temperature is below the limit set in registers MR31 and MR30. |

**Table 7-35. MR51: Temperature Status Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | TS_HIGH_STATUS | R | 0 | Temperature sensor high status<br>0 = Temperature is below the limit set in registers MR29 and MR28.<br>1 = Temperature is above the limit set in registers MR29 and MR28. |

### 7.6.24 MR52: Miscellaneous Error Status (address = 34h) [reset = 00h]

The MR52 register stores the status for PEC checksum failure when PEC mode is enabled and parity error on the T-bit when the host writes to the device in I3C mode.

Return to Register Map.

**Figure 7-70. MR52: Miscellaneous Error Status Register**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | PEC_ERROR_STATUS | PAR_ERROR_STATUS |
| R-00h | | | | | | R-0 | R-0 |

**Table 7-36. MR52: Miscellaneous Error Status Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7:2 | Reserved | R | 00 | Reserved |
| 1 | PEC_ERROR_STATUS | R | 0 | Packet error status.<br>0 = No PEC error.<br>1 = PEC error in one or more packets. |
| 0 | PAR_ERROR_STATUS | R | 0 | Parity check error status<br>0 = No parity error.<br>1 = Parity error in one or more bytes. |

# 8 Application and Implementation

**Note**

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes, as well as validating and testing their design implementation to confirm system functionality.

## 8.1 Application Information

The TMP139 is used to measure the temperature of the memory components on a DIMM card. The TMP139 features an $I^2C$ and I3C bus, and there can be up to 2 devices on the bus as required by the DDR5 application. As the TMP139 operates on the I3C bus, the device does not require an external pull up resistor on the SDA or SCL pin.

## 8.2 Typical Application

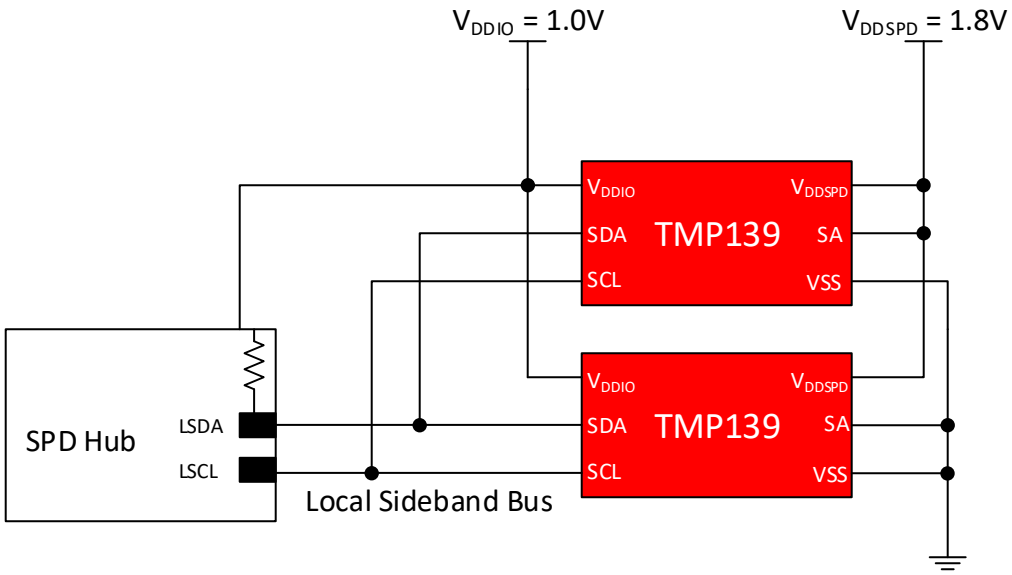



**Figure 8-1. Typical Connections**

### 8.2.1 Design Requirements

The I3C bus does not require an external pullup resistor on the SDA pin as the pullup is embedded in the host controller. The SCL pin is an input-only pin that is driven by the host controller in push-pull mode, and the pin must be connected directly. The SA pin can only be connected to the $V_{DDSPD}$ or GND.

### 8.2.2 Detailed Design Procedure

Place the TMP139 devices in close proximity to the heat source that must be monitored with a proper layout for good thermal coupling. The placement ensures that the temperature changes are captured within the shortest possible time interval.

### 8.2.3 Application Curves

Table 8-1 shows the curves for this application example.

**Table 8-1. Table of Graphs**

| NAME | GRAPH |
|---|---|
| Temperature Error vs Temperature | Figure 6-8 |
| Active Conversion Current vs Temperature | Figure 6-9 |
| Average Current vs Temperature | Figure 6-10 |
| Standby Current vs Temperature | Figure 6-11 |
| Shutdown Current vs Temperature | Figure 6-12 |
| Sampling Rate Change | Figure 6-13 |

## 9 Power Supply Recommendations

The TMP139 operates with dual supply pins. The supply $V_{DDIO}$ is used for the bus interface and operates in the range of 0.95 V to 1.05 V. The pin VDDSPD is used as the supply for the core and operates in the range of 1.7 V to 1.98 V. A power-supply bypass capacitor is required for precision and stability. Place these power-supply capacitors as close to the supply and ground pins of the device as possible. A typical value of these supply bypass capacitor is 0.01 µF. Applications with noisy or high-impedance power supplies can require a bigger bypass capacitor to reject power-supply noise.

# 10 Layout

## 10.1 Layout Guidelines

Place the power-supply bypass capacitor as close as possible to the supply and ground pins as possible. The recommended value of this bypass capacitor is 0.01 µF. The SCL does not require a pull up as it is driven in push-pull mode by the hub device. The SDA does not require an external pull up, as in I3C the pull up resistor is integrated in the hub device as well.

## 10.2 Layout Example

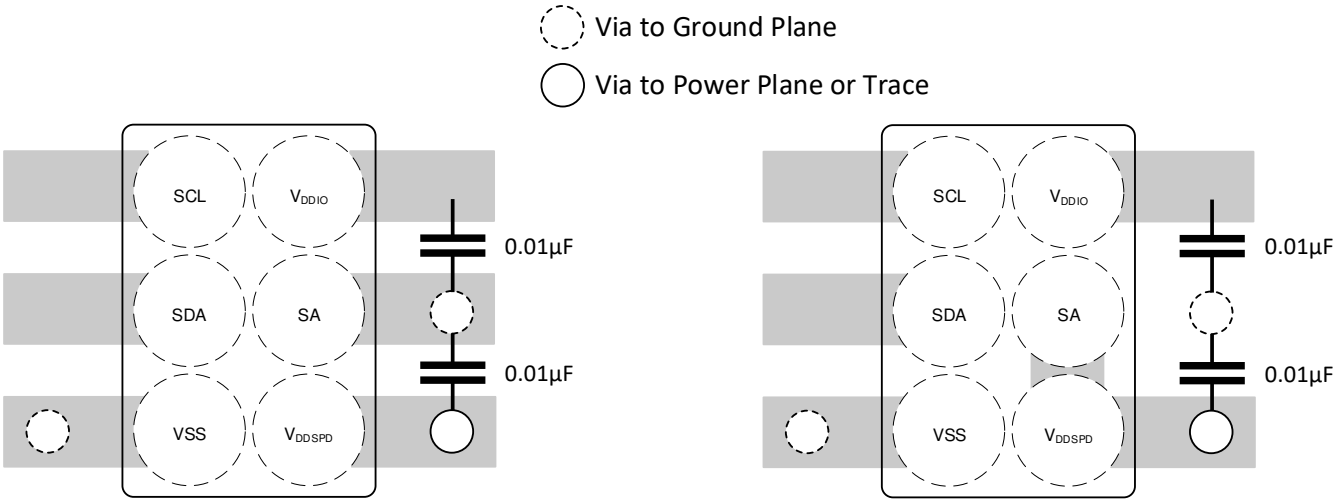



**Figure 10-1. Layout Example**

# 11 Device and Documentation Support

## 11.1 Receiving Notification of Documentation Updates

To receive notification of documentation updates, navigate to the device product folder on ti.com. Click on *Subscribe to updates* to register and receive a weekly digest of any product information that has changed. For change details, review the revision history included in any revised document.

## 11.2 Support Resources

TI E2E™ support forums are an engineer's go-to source for fast, verified answers and design help — straight from the experts. Search existing answers or ask your own question to get the quick design help you need.

Linked content is provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

## 11.3 Trademarks

TI E2E™ is a trademark of Texas Instruments.
All trademarks are the property of their respective owners.

## 11.4 Electrostatic Discharge Caution

This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

## 11.5 Glossary

**TI Glossary** This glossary lists and explains terms, acronyms, and definitions.

# 12 Mechanical, Packaging, and Orderable Information

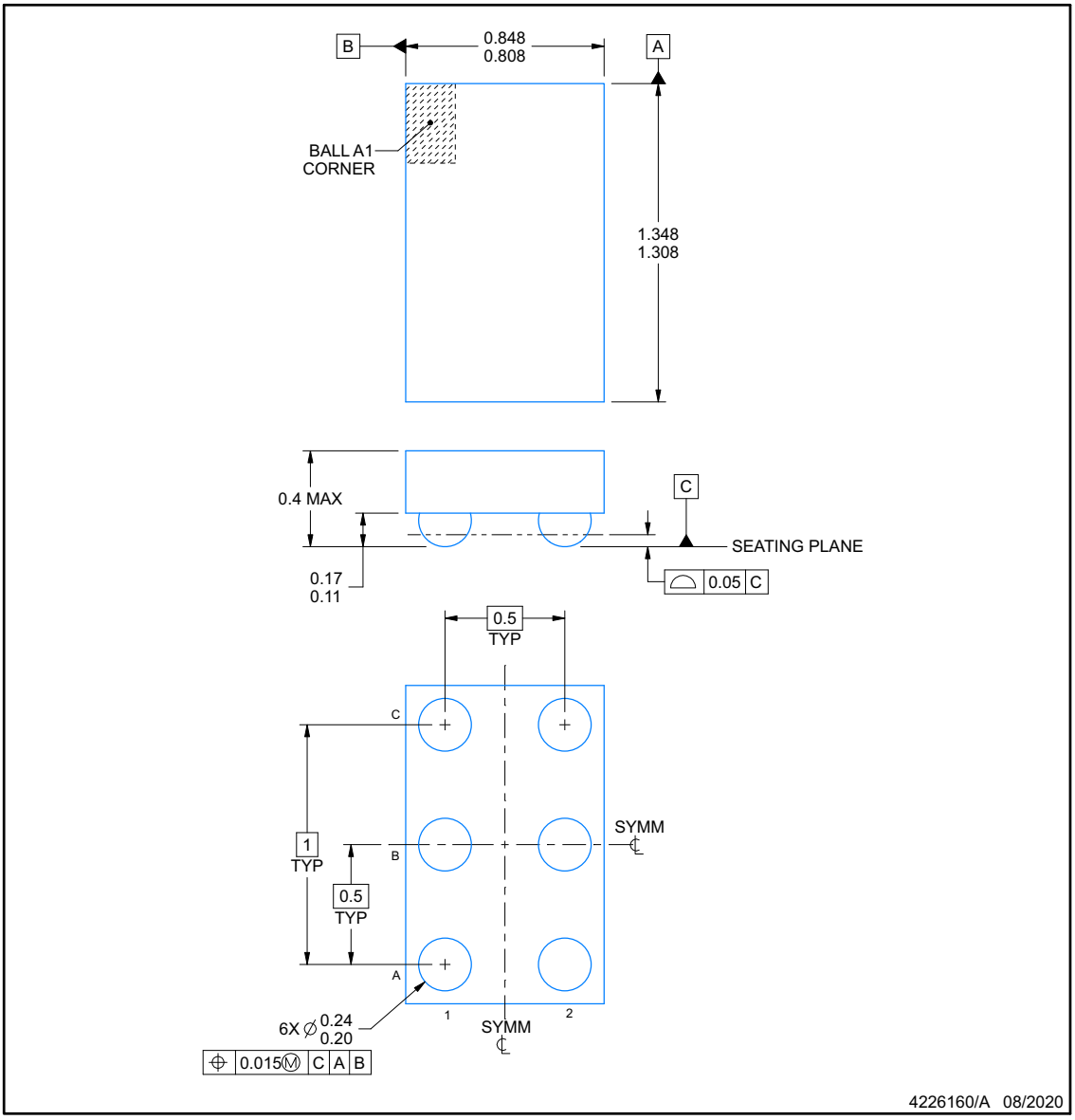
The following pages include mechanical, packaging, and orderable information. This information is the most current data available for the designated devices. This data is subject to change without notice and revision of this document. For browser-based versions of this data sheet, refer to the left-hand navigation.

# PACKAGE OUTLINE

## YAH0006-C01

## DSBGA - 0.4 mm max height

DIE SIZE BALL GRID ARRAY




NOTES:

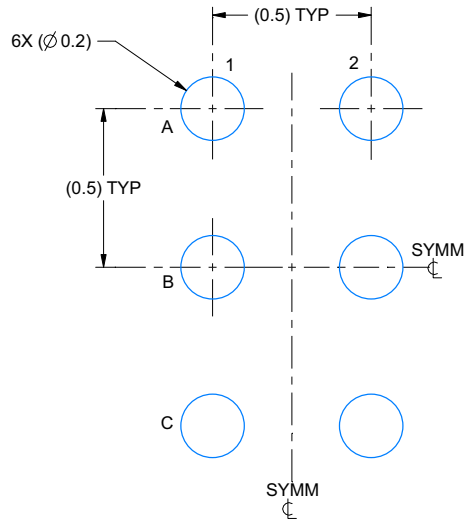
1. All linear dimensions are in millimeters. Any dimensions in parenthesis are for reference only. Dimensioning and tolerancing per ASME Y14.5M.
2. This drawing is subject to change without notice.

## EXAMPLE BOARD LAYOUT

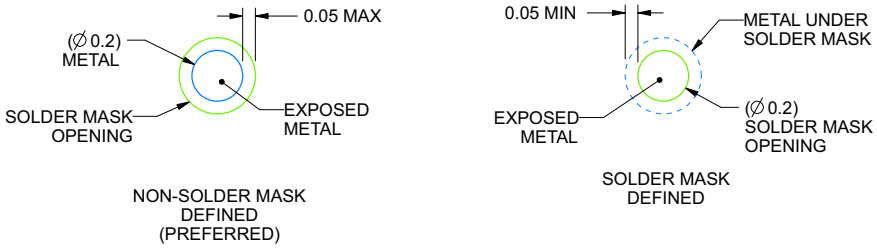
### YAH0006-C01

### DSBGA - 0.4 mm max height

DIE SIZE BALL GRID ARRAY




LAND PATTERN EXAMPLE
EXPOSED METAL SHOWN
SCALE: 50X




SOLDER MASK DETAILS
NOT TO SCALE

4226160/A 08/2020

NOTES: (continued)

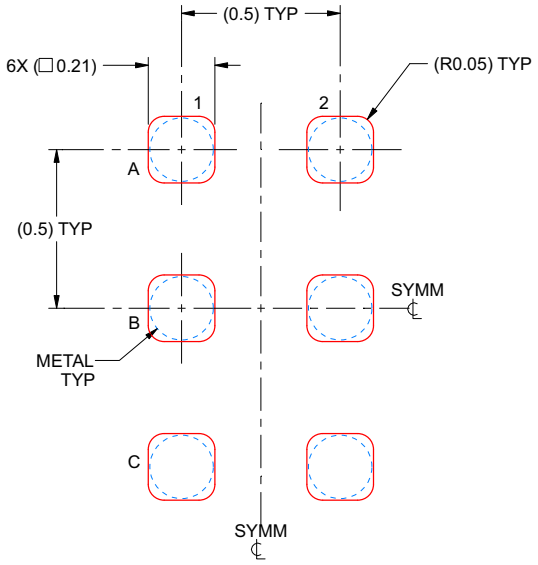
3. Final dimensions may vary due to manufacturing tolerance considerations and also routing constraints.
   See Texas Instruments Literature No. SNVA009 (www.ti.com/lit/snva009).

# EXAMPLE STENCIL DESIGN

## YAH0006-C01

### DSBGA - 0.4 mm max height

DIE SIZE BALL GRID ARRAY




SOLDER PASTE EXAMPLE
BASED ON 0.075 mm THICK STENCIL
SCALE: 50X

4226160/A 08/2020

NOTES: (continued)

4. Laser cutting apertures with trapezoidal walls and rounded corners may offer better paste release.

## PACKAGING INFORMATION

| Orderable Device | Status (1) | Package Type | Package Drawing | Pins | Package Qty | Eco Plan (2) | Lead finish/ Ball material (6) | MSL Peak Temp (3) | Op Temp (°C) | Device Marking (4/5) | Samples |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TMP139AIYAHR | ACTIVE | DSBGA | YAH | 6 | 12000 | RoHS & Green | SNAGCU | Level-1-260C-UNLIM | -40 to 125 | 28VL | Samples |
| TMP139AIYAHT | ACTIVE | DSBGA | YAH | 6 | 250 | RoHS & Green | SNAGCU | Level-1-260C-UNLIM | -40 to 125 | 28VL | Samples |

(1) The marketing status values are defined as follows:
**ACTIVE:** Product device recommended for new designs.
**LIFEBUY:** TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
**NRND:** Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
**PREVIEW:** Device has been announced but is not in production. Samples may or may not be available.
**OBSOLETE:** TI has discontinued the production of the device.

(2) **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".
**RoHS Exempt:** TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.
**Green:** TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

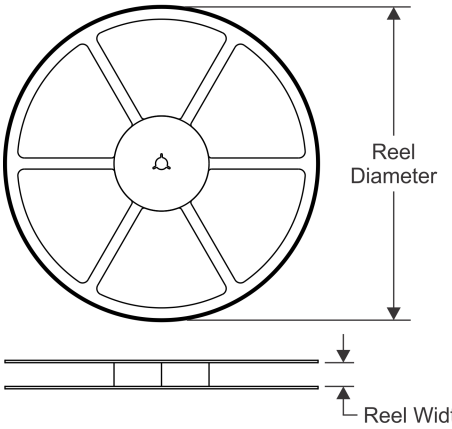
(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

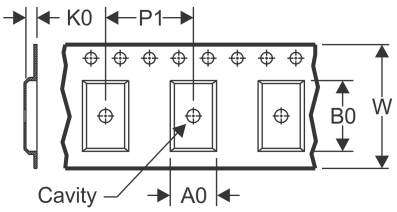
(6) Lead finish/Ball material - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

## TAPE AND REEL INFORMATION

**REEL DIMENSIONS**

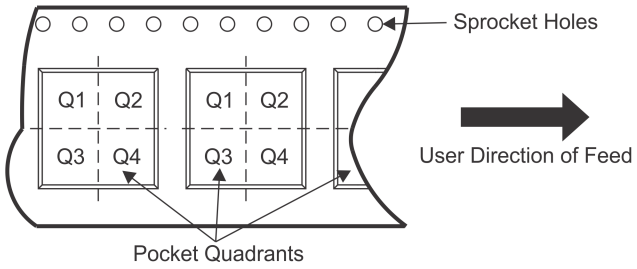



**TAPE DIMENSIONS**




| | |
|---|---|
| A0 | Dimension designed to accommodate the component width |
| B0 | Dimension designed to accommodate the component length |
| K0 | Dimension designed to accommodate the component thickness |
| W | Overall width of the carrier tape |
| P1 | Pitch between successive cavity centers |

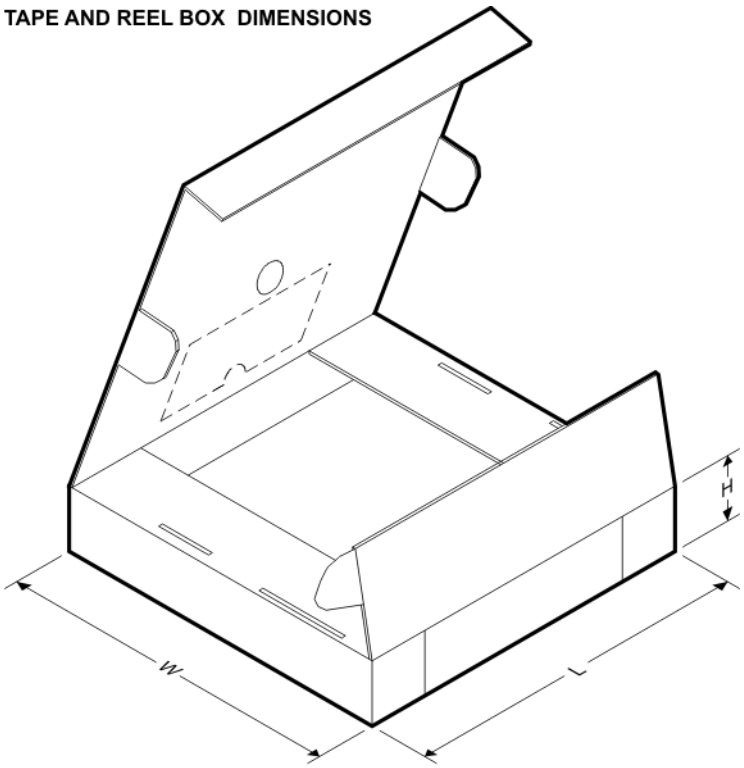
**QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE**




*All dimensions are nominal

| Device | Package Type | Package Drawing | Pins | SPQ | Reel Diameter (mm) | Reel Width W1 (mm) | A0 (mm) | B0 (mm) | K0 (mm) | P1 (mm) | W (mm) | Pin1 Quadrant |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMP139AIYAHR | DSBGA | YAH | 6 | 12000 | 180.0 | 8.4 | 0.93 | 1.43 | 0.47 | 2.0 | 8.0 | Q1 |
| TMP139AIYAHT | DSBGA | YAH | 6 | 250 | 180.0 | 8.4 | 0.93 | 1.43 | 0.47 | 2.0 | 8.0 | Q1 |

TAPE AND REEL BOX DIMENSIONS




*All dimensions are nominal

| Device | Package Type | Package Drawing | Pins | SPQ | Length (mm) | Width (mm) | Height (mm) |
|---|---|---|---|---|---|---|---|
| TMP139AIYAHR | DSBGA | YAH | 6 | 12000 | 182.0 | 182.0 | 20.0 |
| TMP139AIYAHT | DSBGA | YAH | 6 | 250 | 182.0 | 182.0 | 20.0 |

# IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (https:www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265