NCA95118

8-Kbit Serial Presence Detect EEPROM with Hub and Integrated Temperature Sensor

Datasheet (EN) 1.0

Product Overview

The NCA95118 is a device compatible with JEDEC standard JESD300-5B. The SPD5 Hub device contains 1024 bytes of NVM (non-volatile memory) arranged as 16 blocks of 64 bytes per block. Each block may optionally be write-protected via software command. Write protection for each block may be overridden in an offline programmer environment while overrides are prevented in normal use. The SPD5 Hub device operates from 1.8 V nominal power supply input. The SPD5 Hub device is intended to operate up to 12.5 MHz on a 1.0V I3C Basic bus or up to 1 MHz on a 1.0 V to 3.3 V I^2C bus. The SPD5 Hub device is intended to interface to I^2C/I3C Basic buses which have multiple devices on a shared bus and must be uniquely addressed with fixed addressing on the same bus.

The SPD5 Hub device incorporates thermal sensing capability which is controlled and read over I^2C/I3C Basic bus.

Key Features

- SPD5 Hub Device, JEDEC JESD300-5B Compliant
- Two-wire programmable I^2C or I3C Basic bus serial Interface
- Up to 12.5 MHz transfer rate
- Power supply: 1.8 V V_{DDSPD}, 1.0 V V_{DDIO}
- Supports 1.0 V, 1.1 V and 1.2 V push-pull IO levels
- Supports 1.0 V, 1.1 V, 1.2 V, 1.8 V, 2.5 V and 3.3 V open-drain IO Levels
- 1024 bytes of NVM (16 blocks, 64 bytes per block)
- Write protection for each block of NVM
- Hub function with 3 address bits translation
- Integrated temperature sensor: 0.5 °C accuracy with 0.25 °C resolution
- Packet error check (PEC) function
- Parity error check function
- Bus reset function
- Up to 8 unique addresses
- In band interrupt (IBI)
- 9-pin thermally enhanced DFN package
- Temperature range: -40°C to 125°C
- Latch-up performance exceeds 100mA per JESD787E, Class II
- RoHS compliance

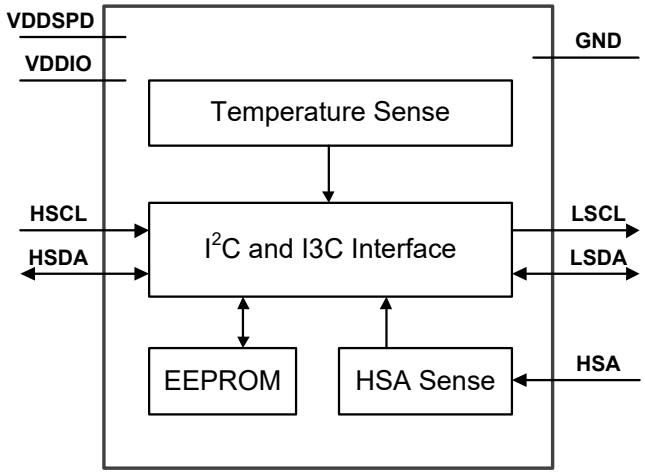
Applications

- DDR5 DIMM modules
- PC, server platforms
- Industrial temperature monitors

Device Information

Part Number	Package	Body Size
NCA95118-DDAKR	DFN8	2.0 mm × 3.0 mm

Functional Block Diagrams



Copyright © 2024, NOVOSENSE

INDEX

Copyright © 2024, NOVOSENSE

Copyright © 2024, NOVOSENSE

Copyright © 2024, NOVOSENSE

1. Pin Configuration

1.1. SPD5 Hub Device Pin Definition

Figure 1 Pinout of NCA95118

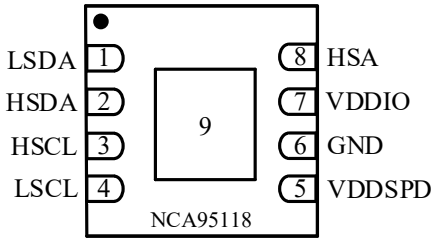


Table 1 Pin Description

Pin	Pin Name	Pin Type[1]	Description
5	V_{DDSPD}	Power	1.8 V input power supply. Connect minimum of 1.0 µF capacitor to GND
6	GND	GND	GND
3	HSCL	I	Host Bus - I²C/I3C Basic input clock
2	HSDA	IO	Host Bus - I²C/I3C Basic data
8	HSA	I	Host Bus - I²C/I3C Basic address Pin. See Table 2 for ID definition.
4	LSCL	O	Local Bus - I²C/I3C Basic output clock
1	LSDA	IO	Local Bus - I²C/I3C Basic data
7	V_{DDIO}	Power	1.0 V input power supply. Connect minimum of 1.0 µF capacitor to GND.
9	Thermal Pad, GND	GND	Connected to GND plane

[1] I = input, O = output, I/O = input and output.

Table 2 shows the HSA pin resistor values and corresponding 3-bit HID for SPD5 Hub device.

Table 2 HSA pin resistor value and HID - SPD5 Family

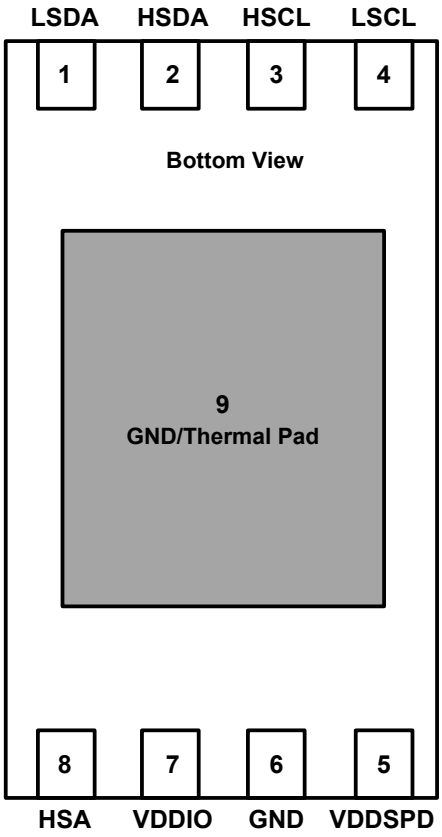
HSA Pin Connection	3-bit HID	Comment
10.0 kΩ to GND	HID = 000	1% resistor
15.4 kΩ to GND	HID = 001	
23.2 kΩ to GND	HID = 010	
35.7 kΩ to GND	HID = 011	
54.9 kΩ to GND	HID = 100	
84.5 kΩ to GND	HID = 101	
127 kΩ to GND	HID = 110	
196 kΩ to GND	HID = 111	
Tied directly to GND	HID = 000	Offline mode: Write protect override enabled

Copyright © 2024, NOVOSENSE

1.2. Package Pinout

The SPD5 Hub device is packaged in a 9-contact thermally enhanced DFN, compliance with PSON-8, MO-229 var WCED-3. The package size is 2.0 mm x 3.0 mm. The pinout is shown in Figure 2 below. The pinout is a bottom view.

Figure 2 Bottom view for NCA95118



Copyright © 2024, NOVOSENSE

2. Device Power Up, Reset and Initialization

2.1. Device Power Up

The SPD5 Hub device has one V_{DDSPD} supply input.

To prevent inadvertent operations during power up, a power-on reset (POR) circuit is included. On cold power on, V_{DDSPD} input supply must rise monotonically between VPON and $V_{DDSPDmin}$ and V_{DDIO} input supply must rise monotonically to $V_{DDIOmin}$ without ring back to ensure proper start up.

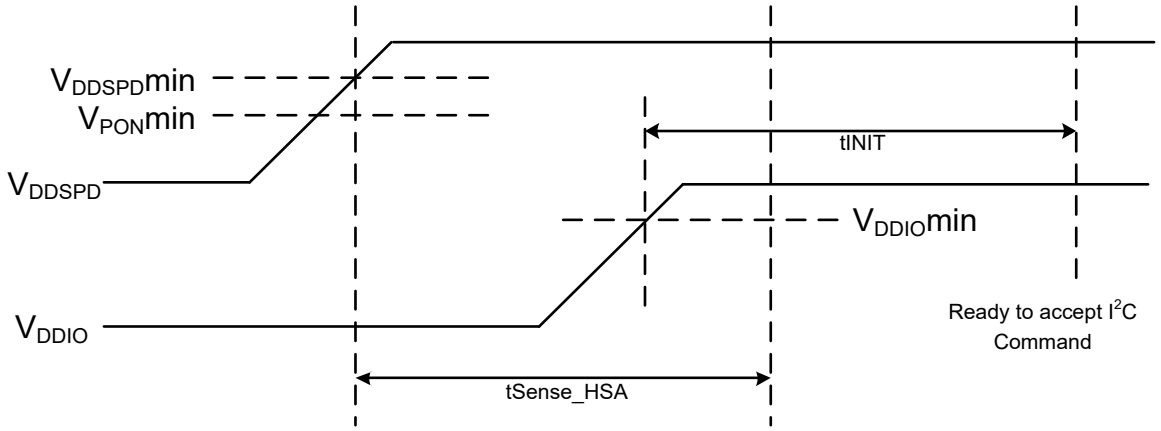
The SPD5 Hub device uses V_{DDIO} supply for its IO levels and it must reach $V_{DDIOmin}$ to ensure proper operation of I^2C or I3C Basic bus interface.

Once the V_{DDSPD} and V_{DDIO} supply is valid and stable, the SPD5 Hub device shall:

1. Once V_{DDSPD} supply is valid and stable, within t_{Sense_HSA} time, sense its HSA pin to determine if SPD5 Hub device is in application environment or in an offline tester program mode. Depending on what it senses on HSA pin, the SPD5 Hub device configures its HID code automatically based on what it detects on HSA pin at power up.
2. Enable I^2C interface within t_{INIT} time and be ready to receive the command from the host. The SPD5 Hub device is ready for operation after t_{INIT} time.

The host interface HSDA and HSCL signal pins are pulled up to a pullup voltage through a pullup resistor on the platform or on the host controller. The pullup voltage can be available before or after V_{DDSPD} or V_{DDIO} is valid and stable. If HSDA and HSCL pullup voltage is available before V_{DDSPD} or V_{DDIO} is available, the HSDA and HSCL signal is high and remains high with no leakage path or damage to the SPD Hub device.

Figure 3 Device Power Up Sequence



2.2. Device Reset and Initialization

At power down (phase during which V_{DDSPD} input supply decreases continuously), as soon as V_{DDSPD} input supply drops below the $V_{DDSPDmin}$, the SPD5 Hub device does not guarantee the operation.

On warm power cycling, the V_{DDSPD} and V_{DDIO} input supply must remain below VPOFF for t_{POFF} and must meet cold power on reset timing when restoring the power.

2.3. Bus Clear

The SPD Hub device supports the following described Bus Clear feature in I^2C mode only. Any attempt by Host to perform I^2C Bus Clear on a Target device in I3C mode may result in an active drive bus contention on the SDA data line.

There may be abnormal circumstances when the host abruptly stops clocking SCL while the target device is in the middle of outputting data for read operation. For this type of events, the SDA data line may appear as stuck low as the device is expecting to receive more clock pulses from the host. Eventually when the host has control of the SCL clock, the host may optionally clear the device that is stuck low on the SDA data line by sending continuous 18 clock pulses without driving the SDA data line followed by stop operation. The device floats the SDA line within 18 clock pulses and returns to the Idle state. The device is ready for normal new transaction with start condition.

Copyright © 2024, NOVOSENSE

2.4. Bus Reset

To prevent a malfunctioning device from locking up the I²C bus or I3C Basic bus, a bus reset mechanism is defined. It uses a timeout mechanism on HSCL as shown in Figure 4 to force a device bus reset. All devices (All SPD5 Hub and all local target devices behind the hub) on an I²C or I3C Basic bus reset simultaneously. Bus reset operation works in the same way regardless of whether the device is operating in I²C or I3C basic mode.

To guarantee the device resets I²C bus or I3C Basic bus, the HSCL clock input Low time has to be greater than or equal to $t_{TIMEOUT(Max)}$.

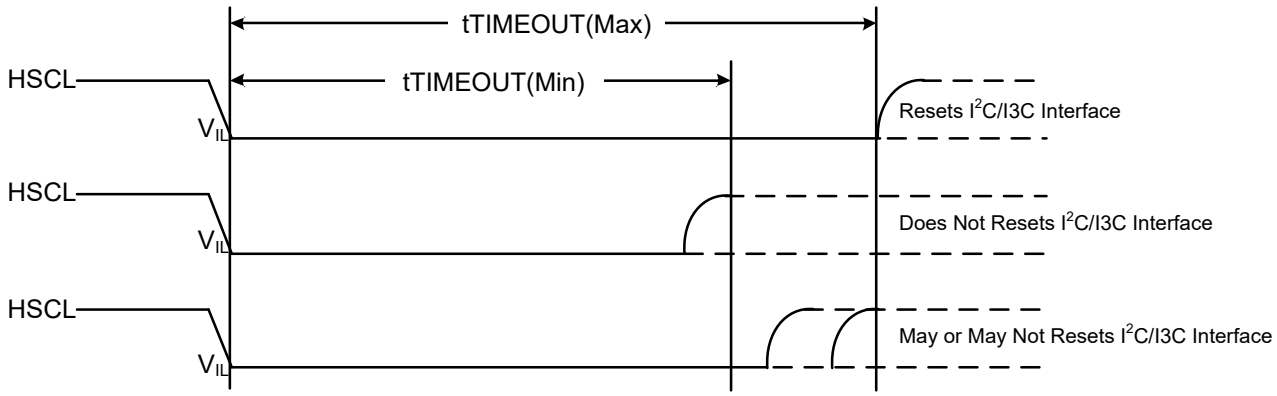
The SPD5 Hub device does not reset I²C bus or I3C Basic bus if the HSCL clock input Low time is less than $t_{TIMEOUT(Min)}$.

If the HSCL clock input Low time is between $t_{TIMEOUT(Min)}$ and $t_{TIMEOUT(Max)}$, the SPD5 Hub device does not guarantee and it may or may not reset the I²C bus or I3C Basic bus.

When RESET, the SPD5 Hub device takes the following actions:

1. Interface and any pending command or transactions are cleared.
2. All internal register values are preserved unless noted otherwise in item # 3 below.
3. Device returns to I²C mode of operation; **MR18**[7:5] resets to '000'; **MR27**[4] resets to '0'; **MR52**[1:0] resets to '00'.
4. Device does not re-sample HSA pin.
5. Device floats the HSDA such that it gets pulled High by external/external/another pullup. The device pulls LSDA pin High.
6. Device treats bus resets as stop operation.

Figure 4 I²C or I3C Basic Bus Reset - SPD5 Hub Device



3. Device Interface - IO Voltage Configuration

The SPD5 Hub device supports configurable open-drain and push-pull IO levels to accommodate broad range of DDR5 platform.

3.1. Open-Drain Interface with Internal On Die Pullup Resistor

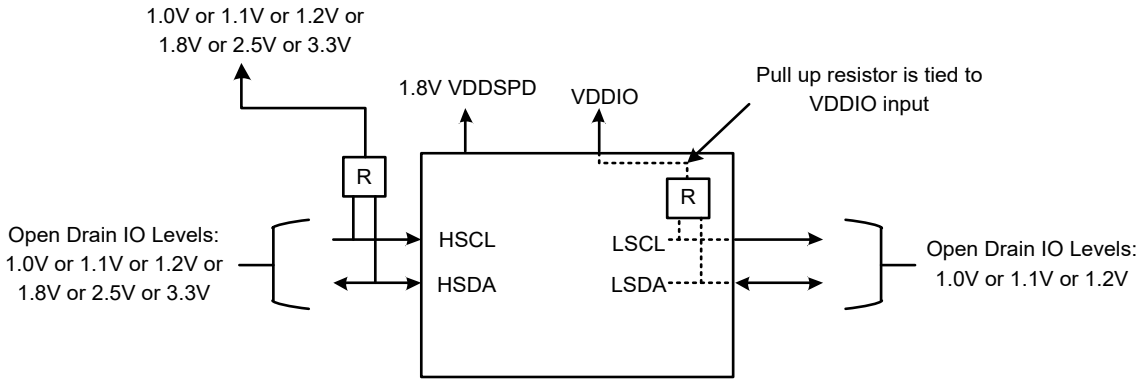
Figure 5 below shows the SPD5 Hub device configuration options for open-drain interface for both Host side and Local side of the device. In this configuration, the SPD5 Hub device supports open-drain IO levels on both host and local side. However, the IO voltage levels on Host side and Local side are independent and can be different.

On host side, the SPD5 device can support IO levels from 1.0 V to 3.3 V depending on the supply rail Host may have pulled up the resistor to. The host side pullup resistor can be on motherboard or on die inside the host logic device.

On local side, the SPD5 device can support IO levels from 1.0 V to 1.2 V and register **MR14**[5] must be configured to '0'. Typical value of on die pullup resistor is 1 kΩ.

Copyright © 2024, NOVOSENSE

Figure 5 Open-Drain Interface; Internal Pullup Resistor



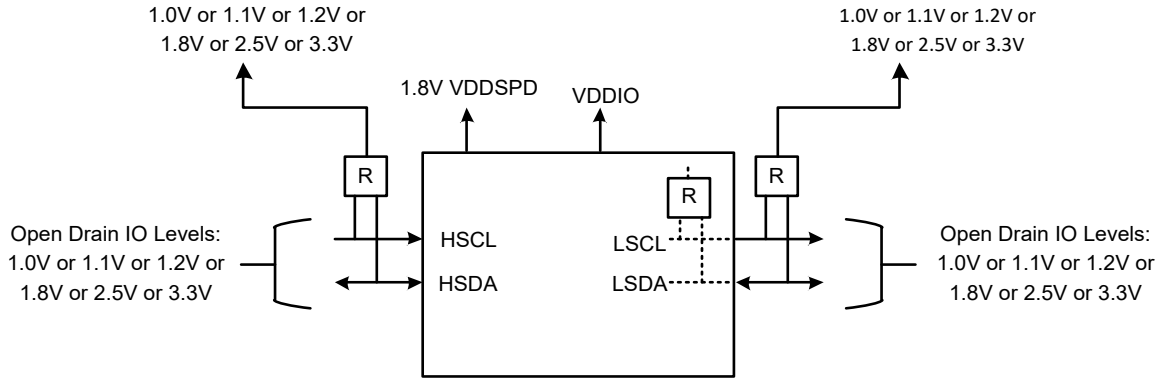
3.2. Open-Drain Interface with External Pullup Resistor

Figure 6 below shows the SPD5 Hub device configuration options for open-drain interface for both host side and local side of the device. In this configuration, the SPD5 device supports open-drain IO levels on both host and local side. However, the IO voltage levels on host side and Local side are independent and can be different.

On host side, the SPD5 Hub device can support IO levels from 1.0 V to 3.3 V depending on the supply rail host may have pulled up the resistor to. The host side pullup resistor can be on motherboard or on die inside the host logic device.

On local side, the SPD5 device can support IO levels from 1.0 V to 3.3 V and **MR14**[5] must be configured to '1'.

Figure 6 Open-Drain Interface; External On-Board Pullup Resistor



3.3. Push-Pull Interface with Internal On Die Pullup Resistor

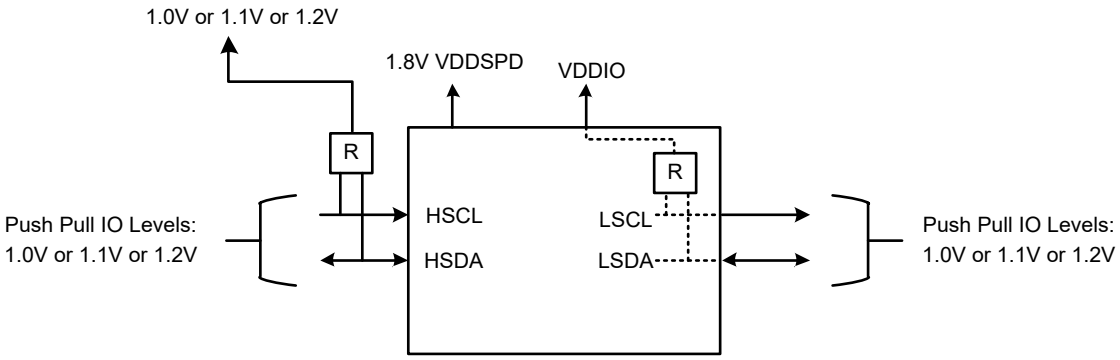
The configuration option shown in Figure 7 is only supported when device is in I3C Basic mode. Figure 7 below shows the SPD5 Hub device configuration options for push-pull interface for both host side and local side of the device. In this configuration, the SPD5 Hub device supports push-pull IO levels on both host and local side. However, the pullup IO voltage levels on host side and local side are independent and can be different.

On host side, the SPD5 Hub device can support IO levels from 1.0 V to 1.2 V. The host side pullup resistor can be on motherboard or on die inside the host logic device.

On local side, the SPD5 Hub device can support IO levels from 1.0 V to 1.2 V and **MR14**[5] can be configured to '0' or '1'.

Copyright © 2024, NOVOSENSE

Figure 7 Push-Pull Interface; On Die Pullup



3.4. IO Operation

The SPD5 Hub device supports configurable IO operation scheme of either open-drain or push-pull on its Host side of the interface (HSCL and HSDA) and Local side of the interface (LSCL and LSDA). Further, the SPD5 Hub device supports independent IO configuration on host side and local side of the interface. The voltage levels for either open-drain mode or push-pull mode can also be independently configurable.

At power on, by default, the SPD5 Hub device comes up in legacy I^2C mode of operation with open-drain IO for both its host side and local side of the interface. The maximum speed is limited to 1 MHz and supported IO voltage levels are from 1.0 V to 3.3 V.

After power on, the host may put the SPD5 Hub device in I3C mode of operation.

In I3C mode, the host may drive the HSCL clock input of the SPD5 device using either push-pull output driver or using the open-drain output driver. It is expected that for all DDR5 DIMM family environment, the host may always drive the HSCL clock input using a push-pull output driver.

To support IBI, the SPD5 device supports dynamic switching between open-drain mode and push-pull mode on its HSDA and LSDA bus for various events. The Table 3 below describes the different mode of operation by the SPD5 device for each cycle.

Table 3 SPD5 Hub Device Dynamic IO Operation Mode Switching

Features	open-drain Mode	push-pull Mode
START + Device Select Code	Yes	No
START + 7'h7E IBI Header Byte	Yes	No
REPEAT START + Device Select Code	No	Yes
REPEAT START + 7'h7E Header Byte	No	Yes
CCC Bytes (i.e. after 7'h7E + W=0 + ACK)	No	Yes
STOP	No	Yes
ACK/NACK Responses	Yes	No
Command, Block Address, Address Operation	No	Yes
Interrupt Request by Target + Device Select Code	Yes	No
IBI Payload	No	Yes
Write Data, T-bit sequence	No	Yes
Read Data, T-bit sequence	No	Yes
PEC, T-bit sequence	No	Yes

Copyright © 2024, NOVOSENSE

4. Device Interface - Protocol

4.1. I²C and I3C Basic Operation

At power on, by default, the SPD5 Hub device comes up in I²C mode of operation. Following applies in I²C mode:

1. The maximum operation speed is limited to 1 MHz.
2. In-band interrupts are not supported.
3. Bus reset is supported.
4. Parity check is not supported except for supported CCCs.
5. Packet Error check is not supported.

The SPD5 Hub device shall operate in the I²C mode until put into I³C Basic mode via command.

The host may put the SPD5 Hub device in I3C Basic mode by issuing SETAASA CCC.

Following applies in I3C mode:

1. The maximum operation speed is up to 12.5 MHz.
2. In-band interrupts are supported.
3. Bus reset is supported.
4. Parity check is always enabled by default.
5. Packet error check is supported and by default is disabled.

4.2. Serial Address of the SPD5 Hub Device

The 7-bit serial address of the SPD5 Hub device and all local devices behind the SPD5 Hub applies to both I²C and I3C Basic mode of operation identically.

The SPD5 Hub device type LID is 4-bit binary value of 1010b.

The SPD5 Hub device samples the status of the HSA pin on power up. The sampled status of the HSA pin determines the unique host ID (HID) of the device. The host identifier value (HID) is merged with the SPD5 Hub device type LID to establish the 7-bit address (SPD5 Hub DevID [6:0]) for the device on the I²C or I3C Basic bus as shown in Table 4. For example, if the value sensed on HSA pin identifier is 2 (010 binary), then the unique 7-bit address for this device is 1010 010b.

Table 4 7-bit Address of SPD5 Hub Device

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	1	0	HID[2]	HID[1]	HID[0]	R/W
SPD5 Hub Device Type ID (LID)				Host ID (HID)			Read/Write

4.3. Serial Address of the Local Devices

Each local device behind the SPD5 Hub device has a unique 4-bit local device ID (LID) code. For example, the PMIC local device type ID is 4-bit binary value of 1001b.

The 3 HID bits of the local device type has a default value of '111' and is merged with its unique 4 bit local ID code to establish the 7-bit address (DevID[6:0]) for the local device on the I²C or I3C Basic bus as shown in Table 5. For example, the PMIC local device behind the SPD5 Hub has 7-bit address of 1001 111b.

Table 5 7-bit Address of the Local Devices (e.g., PMIC Device)

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

Copyright © 2024, NOVOSENSE

4.4. Switch from I²C Mode to I3C Basic Mode

By default, when SPD5 Hub first powers on, it operates in I²C mode. The SPD5 Hub device shall operate in I²C mode until put into I3C Basic mode via command.

In I²C mode, the host is allowed to issue only 3 CCCs (DEVCTRL, SETHID, SETAASA). All other CCCs are not supported and the SPD5 Hub device may simply ignore it. The host must issue DEVCTRL and SETHID CCC first (if required) followed by SETAASA CCC.

The host puts the SPD5 Hub device in I3C Basic mode by issuing SETAASA CCC.

When SETAASA CCC is registered by the SPD5 Hub device, it updates the **MR18**[5] to '1'.

When SETHID CCC is registered by the SPD5 Hub device, it stops the 3-bit HID translation for the local target device as explained in **Section 4.7**.

4.5. Switch from I3C Basic Mode to I²C Mode

The host can put the SPD5 Hub device back in I²C mode from I3C mode at any time by issuing RSTDAA CCC.

When RSTDAA CCC is registered by the SPD5 Hub device, it updates the **MR18**[5] to '0'.

4.6. SPD5 Hub Device Selection

The host can access any SPD5 Hub device as shown in Table 6 below in both I²C mode and I3C Basic mode. The last 3 bits represent the HID bits.

Table 6 7-bit Address of Each Hub Devices on I²C / I3C Bus

Host Access to:	7-bit Address
DIMM 0 SPD Hub	1010 000
DIMM 1 SPD Hub	1010 001
DIMM 2 SPD Hub	1010 010
DIMM 3 SPD Hub	1010 011
DIMM 4 SPD Hub	1010 100
DIMM 5 SPD Hub	1010 101
DIMM 6 SPD Hub	1010 110
DIMM 7 SPD Hub	1010 111

4.7. Local Device Selection Through the SPD5 Hub Device (Before SETHID CCC)

By default, the SPD5 Hub device and all local target devices power up in I²C mode of operation, prior to Host issuing SETHID CCC, the host can access any local device behind the SPD5 Hub device. To access the local device, the host sends 7-bit address that is made up of 4 bits of local device code (LID) followed by the 3 bits of HID code. The LID code represents which local device the host is intended to target. The HID code represents which DIMM the local device resides on. The Table 7 below shows an example of four different local device address codes behind SPD5 Hub on each DIMM.

Table 7 7-bit Address of Each Hub Devices on I²C / I3C Bus

Host Access to:	7-bit Address LID = 1011 (RCD)	7-bit Address LID = 1001 (PMIC)	7-bit Address LID = 0010 (TS0)	7-bit Address LID = 0110 (TS1)
DIMM 0 Local Device	1011 000	1001 000	0010 000	0110 000
DIMM 1 Local Device	1011 001	1001 001	0010 001	0110 001
DIMM 2 Local Device	1011 010	1001 010	0010 010	0110 010

Copyright © 2024, NOVOSENSE

DIMM 3 Local Device	1011 011	1001 011	0010 011	0110 011
DIMM 4 Local Device	1011 100	1001 100	0010 100	0110 100
DIMM 5 Local Device	1011 101	1001 101	0010 101	0110 101
DIMM 6 Local Device	1011 110	1001 110	0010 110	0110 110
DIMM 7 Local Device	1011 111	1001 111	0010 111	0110 111

The SPD5 Hub Device monitors the LID code coming from the host. When it detects the host access is for the target device, it compares the last 3 bits of the HID information (shown in Red) coming from the host against its own unique HID code that it has stored at power on. It compares each 3 bits one at a time. If there is a match, the SPD5 Hub device substitutes that bit with '1' and forwards it to the local device interface. If there is a mismatch, the SPD5 Hub device substitutes that bit with '0' and forwards it to the local device interface. As a result, only the targeted local device will see its last three HID bits as '111' and all non-targeted local devices will see their last three HID bits as anything other than '111' which is not a valid code as shown in Table 5.

There are two exceptions when SPD5 Hub device does not substitute its own HID code when it forwards it to the local target interface:

1. Host issues start followed by 7'h7E with W=0 (or host issues start followed by 0xFC).
2. After SPD5 Hub executes SETHID CCC command that host issues. See **Section 4.8**

Figure 8 gives an example of host accessing local RCD Device on DIMM0. The figure shows host sends 7-bit address '1011 000'. Each SPD5 Hub device receives this address. Each SPD5 Hub device forwards first four bits of binary address '1011' (LID) on the local device interface. Each SPD5 Hub compares last 3 bits of binary address '000' from the host against its own unique HID code and substitutes the bits on the local device interface.

Figure 9 gives another example of host accessing local temperature Sensor 0 device on DIMM3. The figure shows host sends 7-bit address '0010 011'. Each SPD5 Hub device receives this address. Each SDP5 Hub device forwards first four bits of binary address '0010' (LID) on the local device interface. Each SPD5 Hub compares last 3 bits of binary address '011' from the host against its own unique HID code and substitutes the bits on the local device interface.

Copyright © 2024, NOVOSENSE

Figure 8 Example: Host Accessing RCD on DIMM 0

	Hub SPD		RCD		PMIC0		PMIC1		PMIC2		TS0		TS1	
DIMM0	101 0000	50	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	101 1000	58												
Hub Sends			101 1111	5F	101 1111	5F	101 1111	5F	101 1111	5F	101 1111	5F	101 1111	5F
DIMM1	101 0001	51	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	101 1000	58												
Hub Sends			101 1110	5E	101 1110	5E	101 1110	5E	101 1110	5E	101 1110	5E	101 1110	5E
DIMM2	101 0010	52	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	101 1000	58												
Hub Sends			101 1101	5D	101 1101	5D	101 1101	5D	101 1101	5D	101 1101	5D	101 1101	5D
DIMM3	101 0011	53	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	101 1000	58												
Hub Sends			101 1100	5C	101 1100	5C	101 1100	5C	101 1100	5C	101 1100	5C	101 1100	5C
DIMM4	101 0100	54	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	101 1000	58												
Hub Sends			101 1011	5B	101 1011	5B	101 1011	5B	101 1011	5B	101 1011	5B	101 1011	5B
DIMM5	101 0101	55	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	101 1000	58												
Hub Sends			101 1010	5A	101 1010	5A	101 1010	5A	101 1010	5A	101 1010	5A	101 1010	5A
DIMM6	101 0110	56	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	101 1000	58												
Hub Sends			101 1001	59	101 1001	59	101 1001	59	101 1001	59	101 1001	59	101 1001	59
DIMM7	101 0111	57	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	101 1000	58												
Hub Sends			101 1000	58	101 1000	58	101 1000	58	101 1000	58	101 1000	58	101 1000	58

Figure 9 Example: Host Accessing Temperature Sensor 0 on DIMM 3

	Hub SPD		RCD		PMIC0		PMIC1		PMIC2		TS0		TS1	
DIMM0	101 0000	50	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	001 0011	13												
Hub Sends			001 0100	14	001 0100	14	001 0100	14	001 0100	14	001 0100	14	001 0100	14
DIMM1	101 0001	51	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	001 0011	13												
Hub Sends			001 0101	15	001 0101	15	001 0101	15	001 0101	15	001 0101	15	001 0101	15
DIMM2	101 0010	52	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	001 0011	13												
Hub Sends			001 0110	16	001 0110	16	001 0110	16	001 0110	16	001 0110	16	001 0110	16
DIMM3	101 0011	53	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	001 0011	13												
Hub Sends			001 0111	17	001 0111	17	001 0111	17	001 0111	17	001 0111	17	001 0111	17
DIMM4	101 0100	54	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	001 0011	13												
Hub Sends			001 0000	10	001 0000	10	001 0000	10	001 0000	10	001 0000	10	001 0000	10
DIMM5	101 0101	55	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	001 0011	13												
Hub Sends			001 0001	11	001 0001	11	001 0001	11	001 0001	11	001 0001	11	001 0001	11
DIMM6	101 0110	56	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	001 0011	13												
Hub Sends			001 0010	12	001 0010	12	001 0010	12	001 0010	12	001 0010	12	001 0010	12
DIMM7	101 0111	57	101 1111	5F	100 1111	4F	100 0111	47	110 0111	67	001 0111	17	011 0111	37
Host Sends	001 0011	13												
Hub Sends			001 0011	13	001 0011	13	001 0011	13	001 0011	13	001 0011	13	001 0011	13

Copyright © 2024, NOVOSENSE

4.8. Local Device Selection Through the SPD5 Hub Device (After SETHID CCC)

When SETHID CCC is registered by the SPD5 Hub device, it stops the 3-bit HID translation for the local target device as explained in **Section 4.7**. After host sends SETHID CCC, the host still accesses all local target devices behind the SPD5 Hub as shown in Table 7. There is no change in how host accesses the SPD5 Hub device and all local target devices behind the SPD5 Hub device before or after SETHID CCC.

4.9. I²C Target Protocol - Host to SPD5 Hub Device

The SPD5 Hub devices operate on a standard I²C serial interface. Transactions where the SPD5 Hub device is the targeted target device begin with the host issuing a start condition followed by a 7-bit SPD5 Hub device address then a read or write bit, R/W. All data are transmitted with the most significant bit (MSB) first. During the address followed by R/W bit transmission, the SPD5 Hub device typically replies with an ACK unless there are conditions when it may passively assert a NACK.

The SPD5 Hub device does not allow any read or write operations to its non-volatile memory when it is busy with internal write operation to non-volatile memory. The host should either check **MR48**[3] register status or ensure that write time (t_W) parameter is satisfied prior to performing another write or read operations to non-volatile memory. If host violates this and performs either write or read operation to non-volatile memory, the SPD5 Hub device ACKs the address byte to allow possible volatile register access. However, the SPD5 Hub device NACKs the subsequent write or read operation if MemReg = '1' and sets the **MR52**[7] = '1'.

The SPD5 Hub device does allow any read or write operation to its volatile memory (MemReg = '0') when it is busy with internal write operation to non-volatile memory.

To allow compatibility with legacy I²C host controller, the SPD5 Hub device offers two ways to address 1024 bytes of non-volatile memory when it is operating in I²C mode only. By default, the SPD5 Hub device accepts 1 byte of address which covers first 128 bytes of non-volatile memory. The host must select the page pointer through volatile register **MR11**[2:0] to address the entire 1024 bytes of non-volatile memory.

Alternatively, at initial power on, the host can set the **MR11**[3] = '1' to address the entire 1024 bytes of non-volatile memory with 2 bytes of address and hence not required to go through page selection to address entire non-volatile memory.

This two-byte address mode is only applicable to SPD5 Hub device, and it is not applicable to PMIC or TS or RCD device in I²C mode.

The SPD5 Hub device volatile register space does not require the page selection process as all volatile registers are within first 128 bytes.

4.9.1. Write Operation - Data Packet

The MemReg bit determines if the target of the transaction is an NVM location (MemReg = '1') or an internal register (MemReg = '0'). When MemReg = '0', there is no concept of "Block Address"; Block Address bits are treated simply upper address bits.

Table 8 Write Command Data Packet; MR11[3] = '0'

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr[1]	1	0	1	0	HID			W=0	A	
	MemReg	Blk Addr [0]	Address [5:0]						A	
	Data								A	
	…								A	
	Data								A	Sr or P

1 In I2C mode, start or repeat start operation followed by 7'h7E with W=0 is only allowed for the purpose of issuing CCCs that are allowed in I²C mode. Any other operation including another repeat start is considered an illegal operation.

Table 9 Write Command Data Packet; MR11[3] = '1'

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr[1]	1	0	1	0	HID			W=0	A	
	MemReg	Blk Addr [0]	Address [5:0]						A	
	0	0	0	0	Blk Addr [4:1][2]				A	

Copyright © 2024, NOVOSENSE

	Data	A	
	…	A	
	Data	A	Sr or P

1. In I2C mode, start or repeat start operation followed by 7'h7E with W=0 is only allowed for the purpose of issuing CCCs that are allowed in I²C mode. Any other operation including another repeat start is considered an illegal operation.
2. The memory size of SPD5 Hub device is limited to 1024 Bytes. SPD5 Hub device ignores Blk Addr [4] bit.

4.9.2. Read Operation - Data Packet

The MemReg bit determines if the target of the transaction is an NVM location (MemReg = '1') or an internal register (MemReg = '0'). When MemReg = '0', there is no concept of "Block Address"; Block Address bits are treated simply upper address bits.

Table 10 Read Command Data Packet; MR11[3] = '0'

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr[1]	1	0	1	0	HID			W=0	A	
	MemReg	Blk Addr [0]	Address [5:0]						A	
Sr	1	0	1	0	HID			R=1	A[2]	
	Data								A	
	…								A	
	Data								N	Sr or P

1. In I2C mode, start or repeat start operation followed by 7'h7E with W=0 is only allowed for the purpose of issuing CCCs that are allowed in I²C mode. Any other operation including another repeat start is considered an illegal operation.
2. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times as possible it may desire. The device may eventually ACK.

Table 11 Read Command Data Packet; MR11[3] = '1'

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr[1]	1	0	1	0	HID			W=0	A	
	MemReg	Blk Addr [0]	Address [5:0]						A	
	0	0	0	0	Blk Addr [4:1][2]				A	
Sr	1	0	1	0	HID			R=1	A[3]	
	Data								A	
	…								A	
	Data								N	Sr or P

1. In I2C mode, start or repeat start operation followed by 7'h7E with W=0 is only allowed for the purpose of issuing CCCs that are allowed in I²C mode. Any other operation including another repeat start is considered an illegal operation.
2. The memory size of SPD5 Hub device is limited to 1024 Bytes. SPD5 Hub device ignores Blk Addr [4] bit.
3. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times as possible it may desire. The device may eventually ACK.

4.9.3. Default Read Address Pointer Mode

During normal operation of the DDR5 DIMM, the host periodically may poll critical information from the same location. An example may be the SPD5 Hub device's temperature readout. To help improve the efficiency of the I²C bus protocol, the SPD5 Hub offers a

Copyright © 2024, NOVOSENSE

default read address pointer mode so that whenever the SPD5 Hub device sees the stop operation on its HSCL and HSDA bus, its read address pointer always resets to default address. The default read pointer address mode is enabled through register **MR18**[4] and default starting address for read operation is selectable through register **MR18**[3:2]. This allows Host to read the read command data packet as shown in Table 12 compared to as shown in Table 11. The default read address pointer reduces the packet overhead by 3 bytes. The host typically enables this mode when the normal operation of the DDR5 DIMM begins. The default read address pointer mode is only applicable to volatile register space (i.e., MemReg = '0').

Table 12 Read Command Data Packet w/ Default Address Pointer Mode

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr	1	0	1	0	HID			R=1	A	
	Data								A	
	…								A	
	Data								N	Sr or P

Copyright © 2024, NOVOSENSE

4.10. I²C Target Protocol - Host to Local Device Through SPD5 Hub Device

4.10.1. Write Operation - Data Packet

Table 13 Write Command Data Packet (e.g., TS)

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr[1]	0	0	1	0	HID			W=0	A	
	Address [7:0]								A	
	Data								A	
	Data								A	
	…								A	
	Data								A	Sr or P

1. In I2C mode, start or repeat start operation followed by 7'h7E with W=0 is only allowed for the purpose of issuing CCCs that are allowed in I²C mode. Any other operation including another repeat start is considered an illegal operation.

Table 14 Write Command Data Packet (e.g., PMIC)

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr[1]	1	0	0	1	HID			W=0	A	
	Address [7:0]								A	
	Data								A	
	Data								A	
	…								A	
	Data								A	Sr or P

1. In I2C mode, start or repeat start operation followed by 7'h7E with W=0 is only allowed for the purpose of issuing CCCs that are allowed in I²C mode. Any other operation including another repeat start is considered an illegal operation.

When host makes any write request to the RCD device, the RCD protocol has PEC check in the last byte. The SPD simply treats the last byte as data like any other bytes as shown in **Table 16**. The RCD device requires valid stable input clock (DCK_t, DCK_c), Reset_n and DCS_n to allow any read or write access on its I²C interface.

Copyright © 2024, NOVOSENSE

Table 15 Write Command Data Packet (e.g., RCD; PEC Disabled)

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr[1]	1	0	1	1	HID			W=0	A	
	Data (I^2C Bus Command)								A	
	Data (Byte Count=8)								A	
	Data (Reserved; 0x00)								A	
	Data (Dev/Channel Num)								A	
	Data (Page Num [7:0])								A	
	Data (Reg Num [7:0])								A	
	Data (Wr Data [31:24])								A	
	Data (Wr Data [23:16])								A	
	Data (Wr Data [15:8])								A	
	Data (Wr Data [7:0])								A	Sr or P

1. In I2C mode, start or repeat start operation followed by 7'h7E with W=0 is only allowed for the purpose of issuing CCCs that are allowed in I^2C mode. Any other operation including another repeat start is considered an illegal operation.

Table 16 Write Command Data Packet (e.g., RCD; PEC Enabled)

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr[1]	1	0	1	1	HID			W=0	A	
	Data (I^2C Bus Command)								A	
	Data (Byte Count=8)								A	
	Data (Reserved; 0x00)								A	
	Data (Dev/Channel Num)								A	
	Data (Page Num [7:0])								A	
	Data (Reg Num [7:0])								A	
	Data (Wr Data [31:24])								A	
	Data (Wr Data [23:16])								A	
	Data (Wr Data [15:8])								A	
	Data (Wr Data [7:0])								A	
	Data (PEC [7:0])								A	Sr or P

1. In I2C mode, start or repeat start operation followed by 7'h7E with W=0 is only allowed for the purpose of issuing CCCs that are allowed in I^2C mode. Any other operation including another repeat start is considered an illegal operation.

4.10.2. Read Operation - Data Packet

When host makes any read request to the RCD device, the RCD protocol has PEC check for the RCD address information as well as data returned by the RCD. The SPD5 Hub simply treats the PEC information as data byte like any other data byte as shown in Table 20. The SPD5 Hub device does not check for the PEC. The RCD device requires valid stable input clock (DCK_t, DCK_c), Reset_n and DCS_n to allow any read or write access on its I^2C interface.

Copyright © 2024, NOVOSENSE

Table 17 Read Command Data Packet (e.g., TS)

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr[1]	0	0	1	0	HID			W=0	A	
	Address [7:0]								A	
Sr	0	0	1	0	HID			R=1	A[2]	
	Data								A	
	Data								A	
	…								A	
	Data								N	Sr or P

1. In I2C mode, start or repeat start operation followed by 7'h7E with W=0 is only allowed for the purpose of issuing CCCs that are allowed in I²C mode. Any other operation including another repeat start is considered an illegal operation.
2. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times it may desire.

Table 18 Read Command Data Packet (e.g., PMIC)

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr[1]	1	0	0	1	HID			W=0	A	
	Address [7:0]								A	
Sr	1	0	0	1	HID			R=1	A[2]	
	Data								A	
	Data								A	
	…								A	
	Data								N	Sr or P

1. In I2C mode, start or repeat start operation followed by 7'h7E with W=0 is only allowed for the purpose of issuing CCCs that are allowed in I²C mode. Any other operation including another repeat start is considered an illegal operation.
2. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times as possible it may desire.

Copyright © 2024, NOVOSENSE

Table 19 Read Command Data Packet (e.g., RCD; PEC Disabled; Legacy Format)

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr[1]	1	0	1	1	HID			W=0	A	
	Data (I^2C Bus Command)								A	
	Data (Byte Count=4)								A	
	Data (Reserved; 0x00)								A	
	Data (Dev/Channel Num)								A	
	Data (Page Num [7:0])								A	
	Data (Reg Num [7:0])								A	P
S	1	0	1	1	HID			W=0	A	
	Data (I2C Bus Command)								A	
Sr	1	0	1	1	HID			R=1	A[2]	
	Data (Byte Count)								A	
	Data (Status)								A	
	Data (Rd Data [31:24])								A	
	Data (Rd Data [23:16])								A	
	Data (Rd Data [15:8])								A	
	Data (Rd Data [7:0])								A	Sr or P

1. In I2C mode, start or repeat start operation followed by 7'h7E with W=0 is only allowed for the purpose of issuing CCCs that are allowed in I^2C mode. Any other operation including another repeat start is considered an illegal operation.
2. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times as possible it may desire.

Copyright © 2024, NOVOSENSE

Table 20 Read Command Data Packet (e.g., RCD; PEC Enabled; Legacy Format)

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr[1]	1	0	1	1	HID			W=0	A	
	Data (I^2C Bus Command)								A	
	Data (Byte Count=4)								A	
	Data (Reserved; 0x00)								A	
	Data (Dev/Channel Num)								A	
	Data (Page Num [7:0])								A	
	Data (Reg Num [7:0])								A	
	Data (PEC [7:0])								A	P
S	1	0	1	1	HID			W=0	A	
	Data (I^2C Bus Command)								A	
Sr	1	0	1	1	HID			R=1	A[2]	
	Data (Byte Count)								A	
	Data (Status)								A	
	Data (Rd Data [31:24])								A	
	Data (Rd Data [23:16])								A	
	Data (Rd Data [15:8])								A	
	Data (Rd Data [7:0])								A	
	Data (PEC [7:0])								A	Sr or P

1. In I2C mode, start or repeat start operation followed by 7'h7E with W=0 is only allowed for the purpose of issuing CCCs that are allowed in I^2C mode. Any other operation including another repeat start is considered an illegal operation.
2. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times as possible it may desire.

Copyright © 2024, NOVOSENSE

Table 21 Read Command Data Packet (e.g., RCD; PEC Disabled; Optimized Format)

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N	Stop
S or Sr[1]	1	0	1	1	HID			W=0	A	
	Data (I^2C Bus Command)								A	
	Data (Byte Count=4)								A	
	Data (Reserved; 0x00)								A	
	Data (Dev/Channel Num)								A	
	Data (Page Num [7:0])								A	
	Data (Reg Num [7:0])								A	P
Sr	1	0	1	1	HID			R=1	A[2]	
	Data (Byte Count)								A	
	Data (Status)								A	
	Data (Rd Data [31:24])								A	
	Data (Rd Data [23:16])								A	
	Data (Rd Data [15:8])								A	
	Data (Rd Data [7:0])								A	Sr or P

1. In I2C mode, start or repeat start operation followed by 7'h7E with W=0 is only allowed for the purpose of issuing CCCs that are allowed in I^2C mode. Any other operation including another repeat start is considered an illegal operation.
2. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times it may desire.

4.11. I3C Basic Target Protocol - Host to SPD5 Hub Device

4.11.1. Write Operation Data Packet

The SPD5 Hub devices operate on a standard I3C serial interface. Transactions where the SPD5 Hub device is the targeted Target device begin with the host issuing a start condition followed by a 7-bit SPD5 Hub device address then a write bit, R/W. All data are transmitted with the most significant bit MSB first. During the address followed by R/W bit transmission, the SPD5 Hub device typically replies with an ACK unless there are conditions when it may passively assert a NACK. The 'T' bit carries Parity information from the host for each byte.

The SPD5 Hub device does not allow any read or write operations to its non-volatile memory when it is busy with internal write operation to non-volatile memory. The host should either check **MR48**[3] register status or ensure that Write time (t_W) parameter is satisfied prior to performing another write or read operations to non-volatile memory. If host violates this and performs either write or read operation to non-volatile memory, the SPD5 Hub device always ACKs the address byte. In a subsequent byte, the device ignores the write operation if MemReg = '1' and sets the **MR52**[7] = '1' and requests IBI if IBI function is enabled. Subsequently, if Host continues to do the read operation with repeat start, the SPD5 Hub device NACKs.

The SPD5 Hub device does allow any read or write operation to its volatile memory (i.e., MemReg = '0') when it is busy with internal write operation to non-volatile memory.

The Packet Error Check (PEC) function is disabled by default when the SPD5 Hub device is put in I3C Basic mode.

The host may enable this function through DEVCTRL CCC (RegMod = '0'). If enabled, the PEC is appended at the end of all transactions. If PEC is enabled, the host must complete the burst length as indicated in CMD field. In other words, the host must not interrupt the burst length pre-maturely for Write operation.

The MemReg bit determines if the target of the transaction is an NVM location (MemReg = '1') or an internal register (MemReg = '0'). When MemReg = '0', there is no concept of "Block Address"; Block Address bits are treated simply Upper Address bits.

Copyright © 2024, NOVOSENSE

Table 22 Write Command Data Packet; PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	0	1	0	HID			W=0	A[1,2,3]	
	MemReg	Blk Addr [0]	Address [5:0]						T	
	0	0	0	0	Blk Addr [4:1][4]				T	
	Data								T	
	…								T	
	Data								T	Sr[5] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (MemReg bit).
2. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
3. The SPD5 Hub device does not check for parity error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The SPD5 Hub device ignores the entire packet until stop or next repeat start operation.
4. The memory size of SPD5 Hub device is limited to 1024 Bytes. SPD5 Hub device ignores Blk Addr [4] bit.
5. repeat start or repeat start with 7’h7E.

Table 23 Write Command Data Packet; PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	0	1	0	HID			W=0	A[1,2,3]	
	MemReg	Blk Addr [0]	Address [5:0]						T	
	CMD			W=0	Blk Addr [4:1][4]				T	
	Data								T	
	…								T	
	Data								T	
	PEC								T	Sr[5] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (MemReg bit).
2. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
3. The SPD5 Hub device does not check for parity or PEC error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The SPD5 Hub device ignores the entire packet until stop or next repeat start operation.
4. The memory size of SPD5 Hub device is limited to 1024 Bytes. SPD5 Hub device ignores Blk Addr [4] bit.
5. Repeat start or repeat start with 7’h7E.

The host may optionally allow SPD5 Hub device to request IBI. For this case, the transactions to the SPD5 Hub device begin with the I3C Basic Host issuing a start condition followed by 7’h7E and then write bit. If SPD5 Hub device has a pending IBI, it transmits its 7-bit device select code followed by R=1. If SPD5 Hub device has no pending IBI, there is no action taken by SPD5 Hub. The Table 24 and Table 25 show the I3C Basic bus write command data packet with optional IBI header for PEC disabled and PEC enabled cases respectively. Note that in Table 25, PEC calculation does not include IBI header byte (7’h7E followed by W=0).

Copyright © 2024, NOVOSENSE

Table 24 Write Command Data Packet with IBI Header; PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1,3]	
Sr	1	0	1	0	HID			W=0	A[2,3,4]	
	MemReg	Blk Addr [0]	Address [5:0]						T	
	0	0	0	0	Blk Addr [4:1][5]				T	
	Data								T	
	…								T	
	Data								T	Sr[6] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (repeat start).
2. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK) and Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (MemReg bit).
3. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
4. The SPD5 Hub device does not check for parity error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The SPD5 Hub device ignores the entire packet until stop or next repeat start operation.
5. The memory size of SPD5 Hub device is limited to 1024 Bytes. SPD5 Hub device ignores Blk Addr [4] bit.
6. Repeat start or repeat start with 7'h7E.

Table 25 Write Command Data Packet with IBI Header; PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1,3]	
Sr	1	0	1	0	HID			W=0	A[2,3,4]	
	MemReg	Blk Addr [0]	Address [5:0]						T	
	CMD			W=0	Blk Addr [4:1][5]				T	
	Data								T	
	…								T	
	Data								T	
	PEC								T	Sr[6] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (repeat start).
2. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK) and Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (MemReg bit).
3. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
4. The SPD5 Hub device does not check for parity error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The SPD5 Hub device ignores the entire packet until stop or next repeat start operation.
5. The memory size of SPD5 Hub device is limited to 1024 Bytes. SPD5 Hub device ignores Blk Addr [4] bit.
6. Repeat start or repeat start with 7'h7E.

Copyright © 2024, NOVOSENSE

4.11.2. Read Operation Data Packet

The transactions to SPD5 Hub target device begin with the I3C Basic Host issuing a start condition followed by a 7-bit SPD5 Hub device type identifier then a write bit. All I3C Basic bus data are transmitted with MSB first. During select code transmission, the SPD5 Hub device typically replies with an ACK unless there are exceptional conditions when it may passively assert a NACK. See Table 26. The 'T' bit carries Parity information from the host prior to repeated start. After repeated start, 'T' bit carries information from SPD5 Hub device to Host indicating Continuous ('1') or Stop ('0') whether it is transmitting the last byte or not.

The packet error check (PEC) function is disabled by default when SPD5 Hub device is put in I3C Basic mode. The host may enable this function through DEVCTRL CCC (RegMod = '0'). If enabled, the PEC is appended as shown in Table 27. If PEC is enabled, the host must complete the burst length as indicated in CMD field. In other words, the host must not interrupt the burst length pre-maturely for read operation.

The MemReg bit determines if the target of the transaction is an NVM location (MemReg = '1') or an internal register (MemReg = '0'). When MemReg = '0', there is no concept of "Block Address"; Block Address bits are treated simply upper address bits.

Table 26 Read Command Data Packet; PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	0	1	0	HID			W=0	A[1,2,3]	
	MemReg	Blk Addr [0]	Address [5:0]						T	
	0	0	0	0	Blk Addr [4:1][4]				T	
S or Sr	1	0	1	0	HID			R=1	A/N[5,6]	
	Data								T=1	
	…								T=1	
	Data								T=1[7,8]	Sr[9] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (MemReg bit).
2. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
3. The SPD5 Hub device does not check for parity error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The SPD5 Hub device ignores the entire packet until stop or next repeat start operation.
4. The memory size of SPD5 Hub device is limited to 1024 Bytes. SPD5 Hub device ignores Blk Addr [4] bit.
5. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times as possible it may desire. If target device NACKs due to parity error in previous bytes, it will always NACK regardless of how many times the host tries repeat start. If there were no parity errors the SPD5 Hub may eventually ACK.
6. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK).
7. See Figure 13 to see how host ends target device operation.
8. For NVM read memory access (i.e. MemReg = '1'), when last byte is reached (1024 Byte) or for volatile memory access (i.e. MemReg = '0'), when last byte (i.e. MR255) is reached (extreme rare case), the target device sends T = '0'. See Figure 14 to see how target device ends the operation followed by host stop operation.
9. Repeat start or repeat start with 7'h7E.

Copyright © 2024, NOVOSENSE

Table 27 Read Command Data Packet; PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	0	1	0	HID			W=0	A[1,2,3]	
	MemReg	Blk Addr [0]	Address [5:0]						T	
	CMD			R=1	Blk Addr [4:1][4]				T	
	PEC								T	
S or Sr	1	0	1	0	HID			R=1	A/N[5,6]	
	Data								T=1	
	…								T=1	
	Data								T=1	
	PEC								T=0[7]	Sr[8] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (MemReg bit).
2. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
3. The SPD5 Hub device does not check for parity or PEC error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The SPD5 Hub device ignores the entire packet until stop or next repeat start operation.
4. The memory size of SPD5 Hub device is limited to 1024 Bytes. SPD5 Hub device ignores Blk Addr [4] bit.
5. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times as possible it may desire. If target device NACKs due to PEC error parity error in previous bytes, it will always NACK regardless of how many times the host tries repeat start. If there were no parity or PEC errors, the SPD5 Hub may eventually ACK. The PEC calculation by the target device only includes device select code of the ACK response of the repeat start operation. In other words, if there are more than one repeat start operation, the target device includes device select code of only the last repeat start from the host when it ACKs in PEC calculation and all other NACK responses of the device select code of the repeat start are not included in PEC calculation.
6. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK).
7. See Figure 14 to see how target device ends the operation followed by host stop operation.
8. Repeat start or repeat start with 7'h7E.

The host may optionally allow SPD5 Hub device to request IBI. For this case, the transactions to the SPD5 Hub device begin with the I3C Basic Host issuing a start condition followed by 7'h7E and then write bit. If SPD5 Hub device has a pending IBI, it transmits its 7-bit device select code followed by R=1. If SPD5 Hub device has no pending IBI, there is no action taken by SPD5 Hub. The Table 28 and Table 29 show the I3C Basic bus read command data packet with optional IBI header for PEC disabled and PEC enabled cases respectively. Note that in Table 29, PEC calculation (from Host to SPD5 Hub) does not include IBI header byte (7'h7E followed by W=0).

Copyright © 2024, NOVOSENSE

Table 28 Read Command Data Packet with IBI Header; PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1,3]	
Sr	1	0	1	0	HID			W=0	A[2,3,4]	
	MemReg	Blk Addr [0]	Address [5:0]						T	
	0	0	0	0	Blk Addr [4:1][5]				T	
Sr	1	0	1	0	HID			R=1	A/N[6,7]	
	Data								T=1	
	…								T=1	
	Data								T=1[8,9]	Sr[10] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (repeat start).
2. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK) and Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (MemReg bit).
3. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
4. The SPD5 Hub does not check for parity error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The SPD5 Hub ignores the entire packet until stop or next repeat start operation.
5. The memory size of SPD5 Hub device is limited to 1024 Bytes. SPD5 Hub device ignores Blk Addr [4] bit.
6. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK).
7. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times as possible it may desire. If target device NACKs due to parity error in previous bytes, it will always NACK regardless of how many times the host tries repeat start. If there were no parity errors, the SPD5 Hub may eventually ACK.
8. See Figure 13 to see how host ends target device operation.
9. For NVM read memory access (i.e., MemReg = ‘1’), when last byte is reached (1024 Byte) or for volatile memory access (i.e., MemReg = ‘0’), when last byte (i.e. MR255) is reached (extreme rare case), the target device sends T = ‘0’. See Figure 14 to see how target device ends the operation followed by host stop operation.
10. Repeat start or repeat start with 7’h7E.

Copyright © 2024, NOVOSENSE

Table 29 Read Command Data Packet with IBI Header; PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1,3]	
Sr	1	0	1	0	HID			W=0	A[2,3,4]	
	MemReg	Blk Addr [0]	Address [5:0]						T	
	CMD			R=1	Blk Addr [4:1][5]				T	
	PEC								T	
Sr	1	0	1	0	HID			R=1	A/N[6,7]	
	Data								T=1	
	…								T=1	
	Data								T=1	
	PEC								T=0[8]	Sr[9] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (repeat start).
2. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK) and See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (MemReg bit).
3. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
4. The SPD5 Hub does not check for parity or PEC error in subsequent bytes when it determines 7-bit device select code issued by the host does not match its own device code. The SPD5 Hub ignores the entire packet until stop or next repeat start operation.
5. The memory size of SPD5 Hub device is limited to 1024 Bytes. SPD5 Hub device ignores Blk Addr [4] bit.
6. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK).
7. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do repeat start as many times as possible it may desire. If target device NACKs due to PEC error or parity error in previous bytes, it will always NACK regardless of how many times the host tries repeat start. If there was no parity or PEC error, the SPD5 Hub may eventually ACK. The PEC calculation by the target device only includes device select code of the ACK response of the repeat start operation. If there are more than one repeat start operation, the target device includes device select of only the last repeat start from the host when it ACKs in PEC calculation and all other NACK responses of the device select code of the repeat start are not included in PEC calculation.
8. See Figure 14 to see how target device ends the operation followed by host stop operation.
9. Repeat start or repeat start with 7’h7E.

4.11.3. Default Read Address Pointer Mode

This mode works the same exact way as explained in **Section 4.9.3**. Table 30 and Table 31 show the read command data packet for PEC function disabled and enabled respectively. When PEC function is enabled, **MR18**[1] sets the number of bytes that SPD5 Hub device sends out followed by the PEC calculation. If PEC is enabled, the host must complete the burst length as indicated in **MR18**[1] register. In other words, the host must not interrupt the burst length pre-maturely for default address pointer read operation. The default read address pointer mode is only applicable to volatile register space (i.e., MemReg = ‘0’).

Copyright © 2024, NOVOSENSE

Table 30 Read Command Data Packet with Read Address Pointer Mode; PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	0	1	0	HID			R=1	A[1]	
	Data								T=1	
	…								T=1	
	Data								T=1[2,3]	Sr[4] or P

1. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
2. See Figure 13 to see how host ends target device operation.
3. When last byte (i.e., MR255) is reached (extreme rare case), the target device sends T = '0'. See Figure 14 to see how target device ends the operation followed by host stop operation.
4. Repeat start or repeat start with 7'h7E.

Table 31 Read Command Data Packet with Read Address Pointer Mode; PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	0	1	0	HID			R=1	A[1]	
	Data								T=1	
	…								T=1	
	Data								T=1	
	PEC								T=0[2]	Sr[3] or P

1. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
2. See Figure 14 to see how target device ends the operation followed by stop operation.
3. Repeat start or repeat start with 7'h7E.

Table 32 Read CMD Data Packet with Read Address Pointer Mode and IBI Header; PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	0	1	1	1	1	0	W=0	A[1,2]	
Sr	1	0	1	0	HID			R=1	A[2,3]	
	Data								T=1	
	…								T=1	
	Data								T=1[4,5]	Sr[6] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (repeat start).
2. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
3. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK).
4. See Figure 13 to see how host ends target device operation.
5. When last byte (i.e., MR255) is reached (extreme rare case), the target device sends T = '0'. See Figure 14 to see how target device ends the operation followed by host stop operation.
6. Repeat start or repeat start with 7'h7E.

Copyright © 2024, NOVOSENSE

Table 33 Read CMD Data Packet with Read Address Pointer Mode and IBI Header; PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1,2]	
Sr	1	0	1	0	HID			R=1	A[2,3]	
	Data								T=1	
	…								T=1	
	Data								T=1	
	PEC								T=0[4]	Sr[5] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (repeat start).
2. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
3. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK).
4. See Figure 14 to see how target device ends the operation followed by stop operation.
5. Repeat start or repeat start with 7’h7E.

Copyright © 2024, NOVOSENSE

4.12. I3C Basic Target Protocol - Host to Local Device (Through SPD5 Hub Device)

4.12.1. Write Operation - Data Packet

Table 34 to Table 39 shows examples of write command data packet for different types of local devices behind the SPD5 Hub. These examples do not show the optional IBI header byte that host may choose to use. All local devices behind SPD5 Hub device also support the IBI header byte similar to as shown in Table 24 and Table 25.

Table 34 Write Command Data Packet (e.g., TS); PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	0	0	1	0	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
	Data								T	
	Data								T	
	…								T	
	Data								T	Sr[4] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (1st bit of Addr; bit [7]).
2. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
3. The TS device does not check for parity error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The TS device ignores the entire packet until stop or next repeat start operation.
4. Repeat start or repeat start with 7'h7E.

Table 35 Write Command Data Packet (e.g., TS); PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	0	0	1	0	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
	CMD			W=0	0	0	0	0	T	
	Data								T	
	…								T	
	Data								T	
	PEC								T	Sr[4] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (1st bit of Addr; bit [7]).
2. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
3. The TS device does not check for parity or PEC error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The TS device ignores the entire packet until stop or next repeat start operation.
4. Repeat start or repeat start with 7'h7E.

Copyright © 2024, NOVOSENSE

Table 36 Write Command Data Packet (e.g., PMIC); PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	0	0	1	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
	Data								T	
	Data								T	
	…								T	
	Data								T	Sr[4] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (1st bit of Addr; bit [7]).
2. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
3. The PMIC device does not check for parity error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The PMIC device ignores the entire packet until stop or next repeat start operation.
4. Repeat start or repeat start with 7'h7E.

Table 37 Write Command Data Packet (e.g., PMIC); PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	0	0	1	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
	CMD			W=0	0	0	0	0	T	
	Data								T	
	…								T	
	Data								T	
	PEC								T	Sr[4] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (1st bit of Addr; bit [7]).
2. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
3. The PMIC device does not check for parity or PEC error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The TS device ignores the entire packet until stop or next repeat start operation.
4. Repeat start or repeat start with 7'h7E.

The RCD device requires valid stable input clock (DCK_t, DCK_c), Reset_n and DCS_n to allow any read or write access on its I3C Basic interface.

Copyright © 2024, NOVOSENSE

Table 38 Write Command Data Packet (e.g., RCD); PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	0	1	1	HID			W=0	A[1,2,3]	
	Data (I3C Bus Command)								T	
	Data (Byte Count=8)								T	
	Data (Reserved; 0x00)								T	
	Data (Dev/Channel Num)								T	
	Data (Page Num [7:0])								T	
	Data (Reg Num [7:0])								T	
	Data (Wr Data [31:24])								T	
	Data (Wr Data [23:16])								T	
	Data (Wr Data [15:8])								T	
	Data (Wr Data [7:0])								T	Sr[4] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (1st bit of Data, bit [7]).
2. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
3. The RCD device does not check for parity error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The RCD device ignores the entire packet until stop or next repeat start operation.
4. Repeat start or repeat start with 7'h7E.

Table 39 Write Command Data Packet (e.g., RCD); PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	0	1	1	HID			W=0	A[1,2,3]	
	Data (I3C Bus Command)								T	
	Data (Byte Count=8)								T	
	Data (Reserved; 0x00)								T	
	Data (Dev/Channel Num)								T	
	Data (Page Num [7:0])								T	
	Data (Reg Num [7:0])								T	
	Data (Wr Data [31:24])								T	
	Data (Wr Data [23:16])								T	
	Data (Wr Data [15:8])								T	
	Data (Wr Data [7:0])								T	
	Data (PEC [7:0])								T	Sr[4] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (1st bit of Data, bit [7]).

Copyright © 2024, NOVOSENSE

2. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
3. The RCD device does not check for parity or PEC error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The RCD device ignores the entire packet until stop or next repeat start operation.
4. Repeat start or repeat start with 7'h7E.

4.12.2. Read Operation - Data Packet

Table 40 to Table 45 shows examples of read command data packet for different types of local devices behind the SPD5 Hub. These examples do not show the optional IBI header byte that host may choose to use. All local devices behind SPD5 Hub device also support the IBI header byte similar to as shown in Table 28 and Table 29.

Table 40 Read Command Data Packet (e.g., TS); PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	0	0	1	0	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
Sr	0	0	1	0	HID			R=1	A/N[4,5]	
	Data								T=1	
	…								T=1	
	Data								T=1[6,7]	Sr[8] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (1st bit of Addr, bit [7]).
2. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
3. The TS device does not check for parity error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The TS device ignores the entire packet until stop or next repeat start operation.
4. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times as possible it may desire. If target device NACKs due to parity error in previous bytes, it will always NACK regardless of how many times the host tries repeat start. If there were no parity errors, the device may eventually ACK.
5. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK).
6. See Figure 13 to see how host ends target device operation.
7. For volatile register access, when last byte (MR255) is reached (extreme rare case), the target device sends T = '0'. See Figure 14 to see how host ends target device operation.
8. Repeat start or repeat start with 7'h7E.

Table 41 Read Command Data Packet (e.g., TS); PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	0	0	1	0	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
Sr	0	0	1	0	HID			R=1	A/N[4,5]	
	Data								T=1	
	…								T=1	
	Data								T=1	
	PEC								T=0[6]	Sr[7] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (1st bit of Addr, bit [7]).
2. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
3. The TS device does not check for parity or PEC error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The TS device ignores the entire packet until stop or next repeat start operation.
4. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times as possible it may desire. If target device NACKs due to PEC error or parity error in previous bytes, it will always NACK regardless of how many times the host tries repeat start. If there were no parity or PEC errors, the device may eventually ACK. The PEC calculation by the target device only includes device select code of the ACK response of the repeat start operation. In other words, if there are more than one repeat start operation, the target device includes device select of only the last repeat start from the host when it ACKs in PEC calculation and all other NACK responses of the device select code of the repeat start are not included in PEC calculation.
5. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK).
6. See Figure 14 to see how target device ends the operation followed by host stop operation.
7. Repeat start or repeat start with 7'h7E.

Table 42 Read Command Data Packet (e.g., PMIC); PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	0	0	1	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
Sr	1	0	0	1	HID			R=1	A/N[4,5]	
	Data								T=1	
	…								T=1	
	Data								T=1[6,7]	Sr[8] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (1st bit of Addr, bit [7]).
2. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
3. The PMIC device does not check for parity error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The TS device ignores the entire packet until stop or next repeat start operation.
4. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times as possible it may desire. If target device NACKs due to parity error in previous bytes, it will always NACK regardless of how many times the host tries repeat start. If there were no parity errors, the device may eventually ACK.
5. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK).
6. See Figure 13 to see how host ends target device operation.
7. For volatile register access, when last byte is reached within the region (either Host region or DIMM Vendor region), it will continue to return the data but returned data is 0x00 if there is no valid password for DIMM vendor region or device vendor specific region. Once the address counter reaches MR255, it resets to address MR0, and it continues to return the data. Only Host can perform the stop operation.
8. Repeat start or repeat start with 7'h7E.

Copyright © 2024, NOVOSENSE

Table 43 Read Command Data Packet (e.g., PMIC); PEC Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	0	0	1	HID			W=0	A[1,2,3]	
	Address [7:0]								T	
Sr	1	0	0	1	HID			R=1	A/N[4,5]	
	Data								T=1	
	…								T=1	
	Data								T=1	
	PEC								T=0[6]	Sr[7] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (1st bit of Addr, bit [7]).
2. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
3. The PMIC device does not check for parity or PEC error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The TS device ignores the entire packet until stop or next repeat start operation.
4. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times as possible it may desire. If target device NACKs due to PEC error or parity error in previous bytes, it will always NACK regardless of how many times the host tries repeat start. If there were no parity or PEC errors, the device may eventually ACK. The PEC calculation by the target device only includes device select code of the ACK response of the repeat start operation. In other words, if there are more than one repeat start operation, the target device includes device select of only the last repeat start from the host when it ACKs in PEC calculation and all other NACK responses of the device select code of the repeat start are not included in PEC calculation.
5. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK).
6. See Figure 14 to see how target device ends the operation followed by host stop operation.
7. Repeat start or repeat start with 7’h7E.

The RCD device requires valid stable input clock (DCK_t, DCK_c), Reset _n and Chip Select input (DCS_n) to allow any read or write access on its I3C Basic interface.

Copyright © 2024, NOVOSENSE

Table 44 Read Command Data Packet (e.g., RCD); PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	0	1	1	HID			W=0	A[1,2,3]	
	Data (I^2C Bus Command)								T	
	Data (Byte Count=4)								T	
	Data (Reserved; 0x00)								T	
	Data (Dev/Channel Num)								T	
	Data (Page Num [7:0])								T	
	Data (Reg Num [7:0])								T	
Sr	1	0	1	1	HID			R=1	A/N[4,5]	
	Data (Byte Count)								T=1	
	Data (Status)								T=1	
	Data (Rd Data [31:24])								T=1	
	Data (Rd Data [23:16])								T=1	
	Data (Rd Data [15:8])								T=1	
	Data (Rd Data [7:0])								T=0[6]	Sr[7] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (1st bit of Data).
2. The RCD NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
3. The RCD device does not check for parity error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The RCD device ignores the entire packet until stop or next repeat start operation.
4. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times as possible it may desire. If target device NACKs due to parity error in previous bytes, it will always NACK regardless of how many times the host tries repeat start. If there were no parity errors, the device may eventually ACK.
5. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK).
6. See Figure 14 to see how target device ends the operation followed by host stop operation.
7. Repeat start or repeat start with 7'h7E.

Copyright © 2024, NOVOSENSE

Table 45 Read Command Data Packet (e.g., RCD); PEC Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	0	1	1	HID			W=0	A[1,2,3]	
	Data (I^2C Bus Command)								T	
	Data (Byte Count=4)								T	
	Data (Reserved; 0x00)								T	
	Data (Dev/Channel Num)								T	
	Data (Page Num [7:0])								T	
	Data (Reg Num [7:0])								T	
Sr	1	0	1	1	HID			R=1	A/N[4,5]	
	Data (Byte Count)								T=1	
	Data (Status)								T=1	
	Data (Rd Data [31:24])								T=1	
	Data (Rd Data [23:16])								T=1	
	Data (Rd Data [15:8])								T=1	
	Data (Rd Data [7:0])								T=1	
	Data (PEC [7:0])								T=0[6]	Sr[7] or P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation (1st bit of Data).
2. The RCD NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
3. The RCD device does not check for parity or PEC error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The RCD device ignores the entire packet until stop or next repeat start operation.
4. If target device NACKs during repeat start for any reason, the host may re-try repeat start again. The host can do the repeat start as many times as possible it may desire. If target device NACKs due to parity error in previous bytes, it will always NACK regardless of how many times the host tries repeat start. If there were no parity or PEC errors, the device may eventually ACK. The PEC calculation by the target device only includes device select code of the ACK response of the repeat start operation. In other words, if there are more than one repeat start operation, the target device includes device select of only the last repeat start from the host when it ACKs in PEC calculation and all other NACK responses of the device select code of the repeat start are not included in PEC calculation.
5. See Figure 12 to see how the transition occurs from host push-pull operation to target open-drain (ACK).
6. See Figure 14 to see how target device ends the operation followed by host stop operation.
7. Repeat start or repeat start with 7'h7E.

Copyright © 2024, NOVOSENSE

Figure 10 target open-drain to host push-pull Hand Off Operation

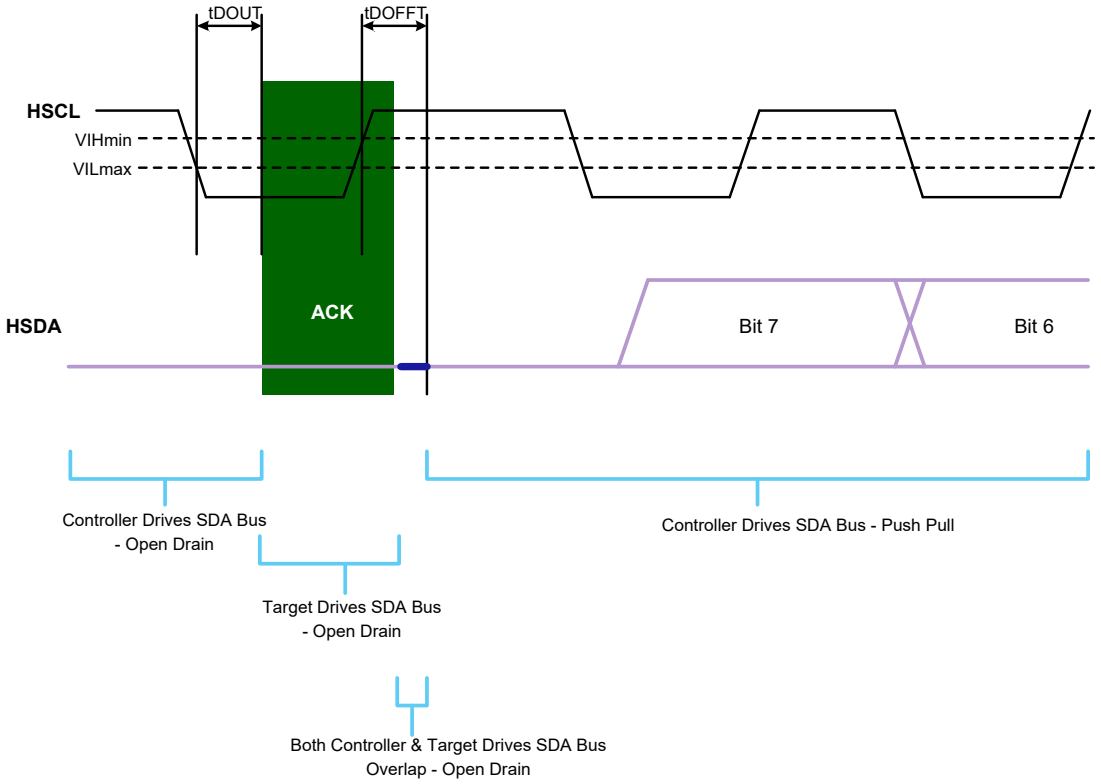
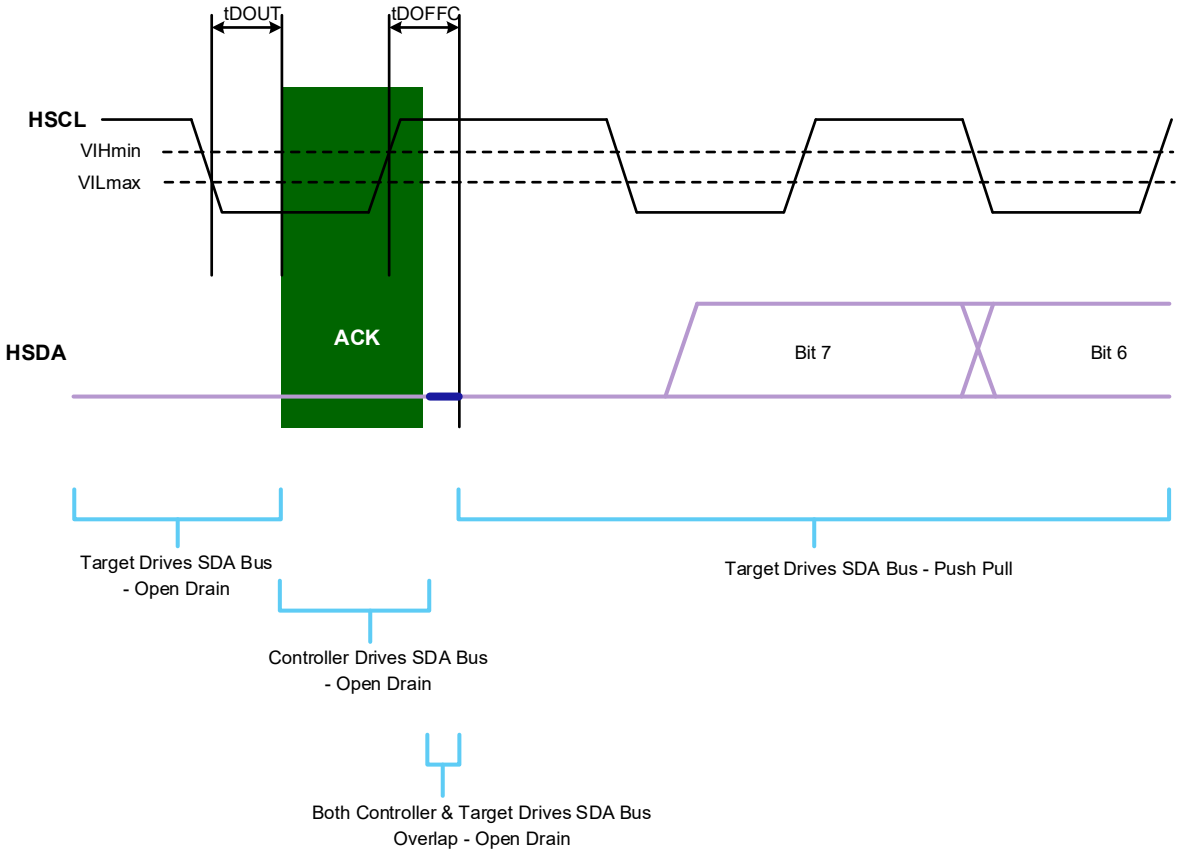
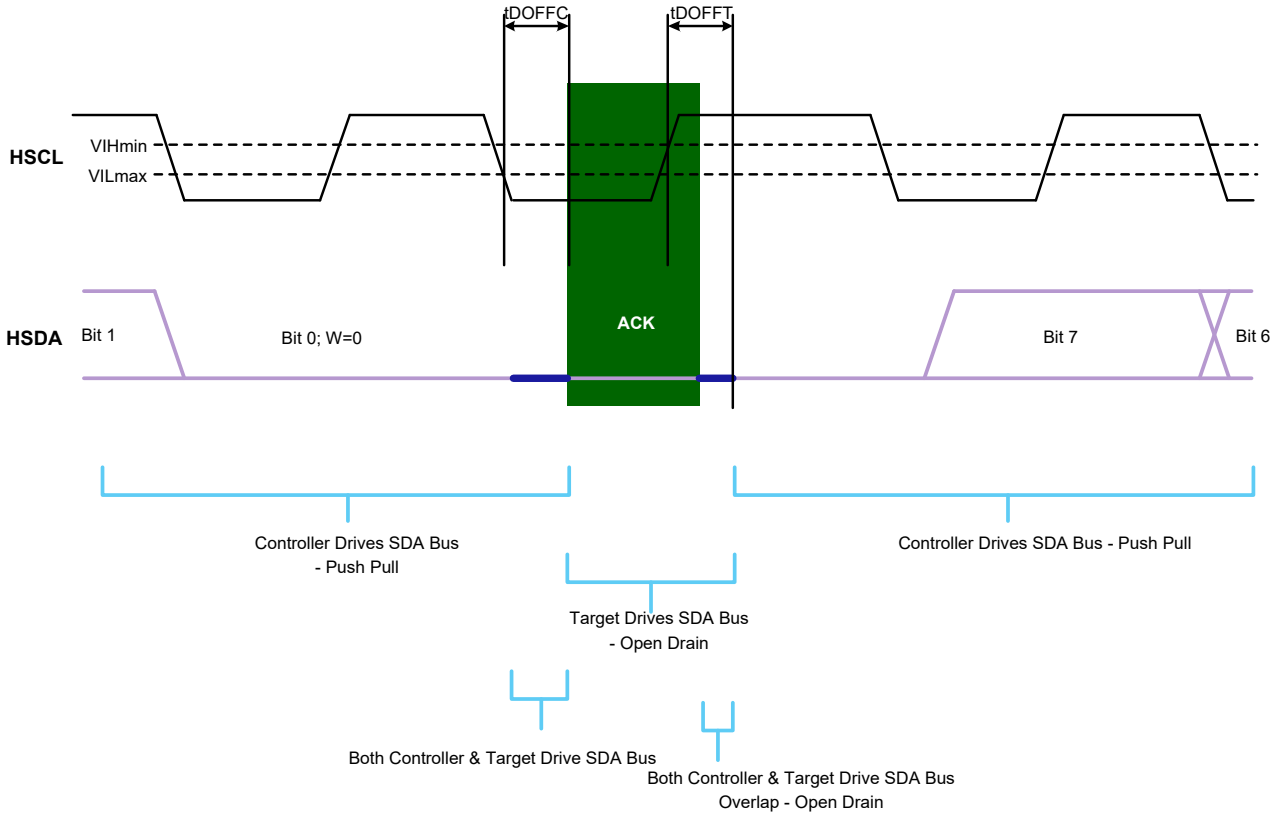


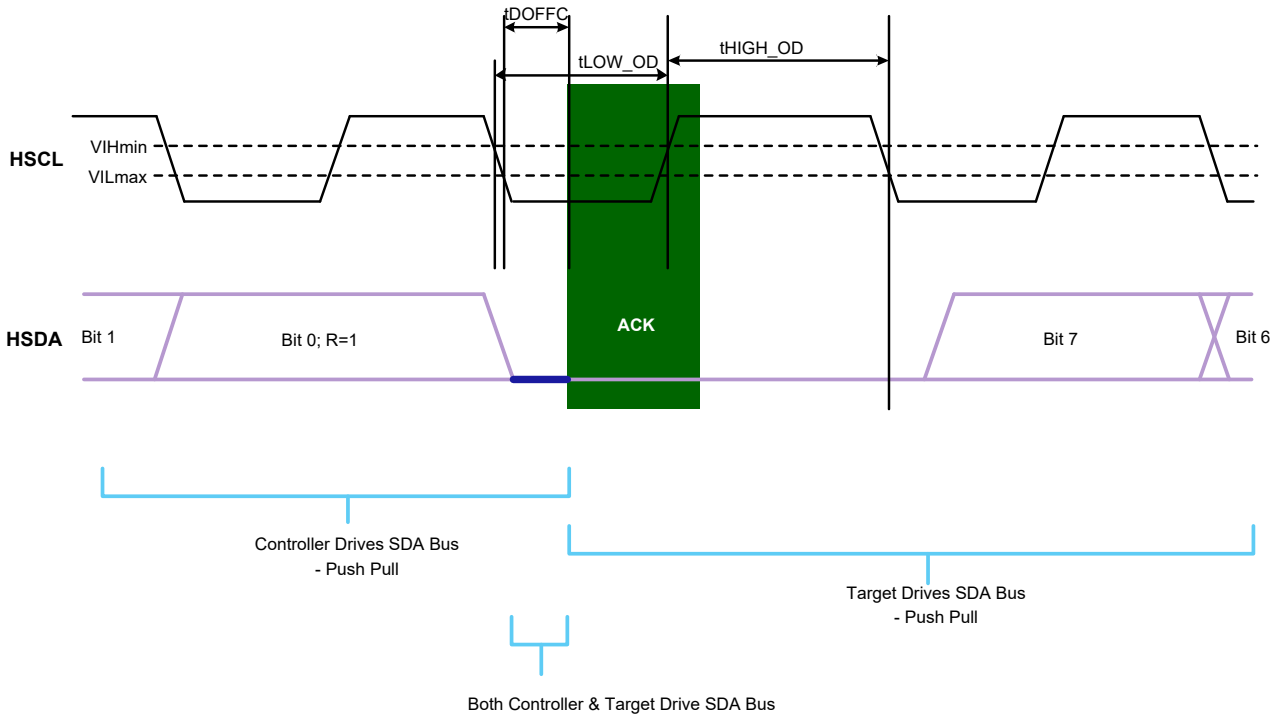
Figure 11 Controller open-drain (ACK) to Target push-pull Hand Off Operation



Copyright © 2024, NOVOSENSE

Figure 12 Controller push-pull to target open-drain Hand Off Operation





Copyright © 2024, NOVOSENSE

Figure 13 T = 1; Host Ends Read with Repeated start and STOP Waveform

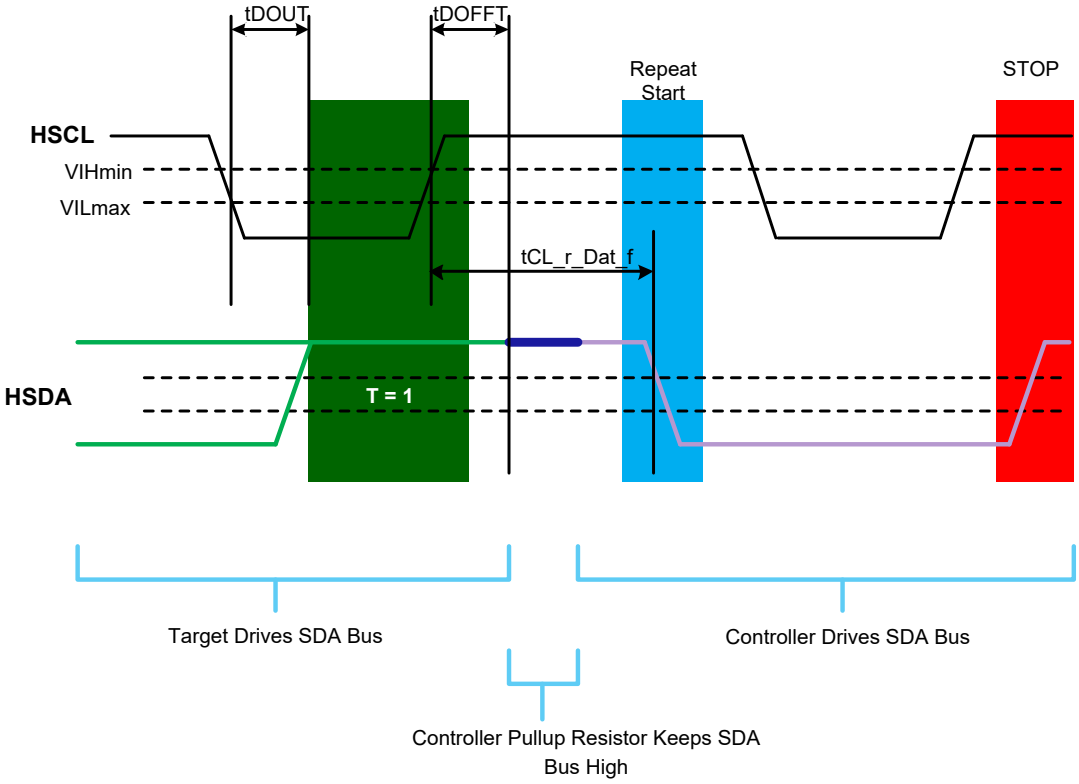
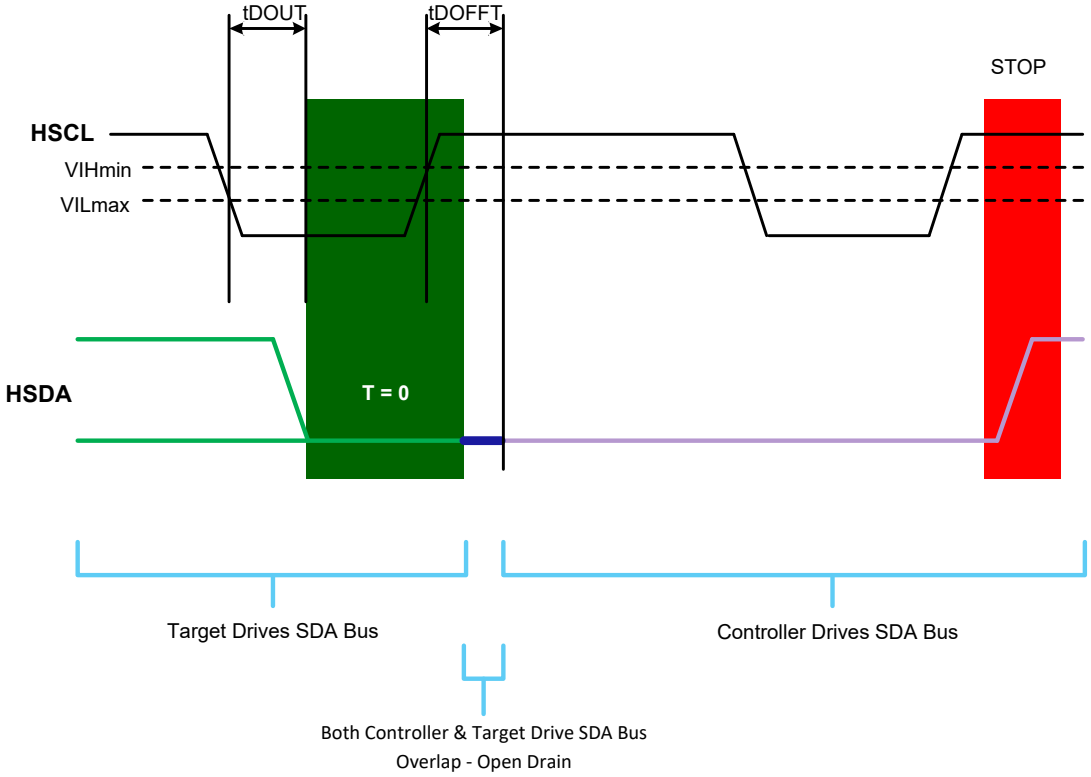


Figure 14 T = 0; Target Ends Read; Host Generates STOP



Copyright © 2024, NOVOSENSE

4.13. I3C Basic Common Command Codes (CCC)

The I3C Basic specification lists large number of common command codes (CCC). Not all CCC are required to be supported. The SPD5 Hub device NACKs for all unsupported CCC. The SPD5 Hub supports CCC as listed in Table 46 below.

The SPD5 Hub device requires stop operation in between when switching from CCC operation to private device specific Write or Read or Default Read Address Pointer mode operation and vice versa. In other words, any CCC operation must be followed by stop operation before continuing to any device specific Write or Read or Default Read Address Pointer mode operation. Similarly, any device specific Write or Read or Default Read Address Pointer mode operation must be followed by stop operation before continuing to any CCC operation. The SPD5 Hub device also requires stop operation between any direct CCC to broadcast CCC.

The SPD5 Hub device does allow repeat start operation between any direct CCC to any other direct CCC or between any broadcast CCC to any other broadcast CCC or between any private Write or Read or Default Read Address Pointer mode operation to any other private Write or Read or Default Read Address Pointer mode operation.

Table 46 SPD5 Hub CCC Support Requirement

CCC	Mode	Code	Description
ENEC	Broadcast	0x00	Enable Event Interrupts
	Direct	0x80	
DISEC	Broadcast	0x01	Disable Event Interrupts
	Direct	0x81	
RSTDAA	Broadcast	0x06	Put the device in I^2C Mode (aka: Reset Dynamic Address Assignment)
SETAASA	Broadcast	0x29	Put the device in I3C Basic Mode (aka: Set All Addresses to Static Address)
GETSTATUS	Direct	0x90	Get Device Status
DEVCAP[1]	Direct	0xE0	Get Device Capability
SETHID[1]	Broadcast	0x61	SPD5 Hub updates 3-bit HID field, updates 'T' bit with updated parity calculation for all devices behind Hub and stops 3-bit HID translation.
DEVCTRL[1]	Broadcast	0x62	Configure SPD5 Hub and all devices behind Hub

1. JEDEC specific CCC.

Copyright © 2024, NOVOSENSE

4.13.1. ENEC CCC

The ENEC CCC is only supported after device is put in I3C Basic mode. In I^2C mode, it is illegal for Host to issue this CCC. When ENEC CCC is registered by the SPD5 Hub, it updates **MR27**[4] = '1' and it takes in effect at the next Start operation (i.e., after STOP condition). Table 47 to Table 50 shows an example of a single ENEC CCC. Table 51 shows the encoding definition for ENEC CCC.

If PEC function is enabled, the PEC calculation starts with start or repeat start operation but does not include 7'h7E with W=0 byte in PEC calculation.

Table 47 ENEC CCC - Broadcast

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x00 (Broadcast)								T	
	7'h00							ENINT	T	Sr[2] or P

1. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
2. Repeat start or repeat start with 7'h7E.

Table 48 ENEC CCC - Broadcast with PEC

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x00 (Broadcast)								T	
	7'h00							ENINT	T	
	PEC								T	Sr[2] or P

1. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
2. Repeat start or repeat start with 7'h7E.

Table 49 ENEC CCC - Direct

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x80 (Direct)								T	
Sr	DevID [6:0]							W=0	A[1,2]	
	7'h00							ENINT	T	Sr[3] or P

1. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
2. The SPD5 Hub device does not check for parity error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The SPD5 Hub device ignores the entire packet until stop or next repeat start operation.
3. Repeat start or repeat start with 7'h7E.

Copyright © 2024, NOVOSENSE

Table 50 ENEC CCC - Direct with PEC

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x80 (Direct)								T	
	PEC								T	
Sr	DevID [6:0]							W=0	A[1,2]	
	7'h00							ENINT	T	
	PEC								T	Sr[3] or P

1. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
2. The SPD5 Hub device does not check for parity or PEC error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The SPD5 Hub device ignores the entire packet until stop or next repeat start operation.
3. Repeat start or repeat start with 7'h7E.

Table 51 ENEC CCC Byte Encoding

Bit	Encoding	Notes
ENINT	0 = No Action 1 = Enable IBI Interrupt	It is illegal for Host to issue ENEC CCC with ENINT bit = '0'

4.13.2. DISEC CCC

The DISEC CCC is only supported after device is put in I3C Basic mode. In I^2C mode, it is illegal for Host to issue this CCC. When DISEC CCC is registered by the SPD5 Hub, it updates **MR27**[4] = '0' and it takes in effect at the next Start operation (i.e., after STOP condition). Table 52 to Table 55 shows an example of a single DISEC CCC. Table 56 shows the encoding definition for DISEC CCC.

If PEC function is enabled, the PEC calculation starts with start or repeat start operation but does not include 7'h7E with W=0 byte in PEC calculation.

Table 52 DISEC CCC - Broadcast

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x01 (Broadcast)								T	
	7'h00							DISINT	T	Sr[2] or P

1. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
2. Repeat start or repeat start with 7'h7E.

Table 53 DISEC CCC - Broadcast with PEC

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x01 (Broadcast)								T	
	7'h00							DISINT	T	

Copyright © 2024, NOVOSENSE

	PEC	T	Sr[2] or P

1. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
2. Repeat start or repeat start with 7'h7E.

Table 54 DISEC CCC - Direct

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x81 (Direct)								T	
Sr	DevID [6:0]							W=0	A[1,2]	
	7'h00							ENINT	T	Sr[3] or P

1. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
2. The SPD5 Hub device does not check for parity error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The SPD5 Hub device ignores the entire packet until stop or next repeat start operation.
3. Repeat start or repeat start with 7'h7E.

Table 55 DISEC CCC - Direct with PEC

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x81 (Direct)								T	
	PEC								T	
Sr	DevID [6:0]							W=0	A[1,2]	
	7'h00							DISINT	T	
	PEC								T	Sr[3] or P

1. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
2. The SPD5 Hub device does not check for parity or PEC error in subsequent bytes when it determines 7-bit device select code issued by the host does not match with its own device code. The SPD5 Hub device ignores the entire packet until stop or next repeat start operation.
3. Repeat start or repeat start with 7'h7E.

Table 56 DISEC CCC Byte Encoding

Bit	Encoding	Notes
DISINT	0 = No Action 1 = Disable IBI Interrupt	It is illegal for Host to issue DISEC CCC with DISINT bit = '0'

4.13.3. RSTDAA CCC

The RSTDAA CCC is only supported after device is put in I3C Basic mode. In I^2C mode, this CCC is ignored. When RSTDAA CCC is registered by the SPD5 Hub, it updates MR18[5] = '0' and it takes in effect at the next Start operation (i.e., after STOP condition).

Copyright © 2024, NOVOSENSE

Further it disables IBI and PEC function (**MR27**[4] = '0', **MR18**[7] = '0' respectively) and clears parity function (MR18[6] = '0'). Table 57 to Table 58 shows an example of a single RSTDAA CCC.

If PEC function is enabled, the PEC calculation starts with start or repeat start operation but does not include 7'h7E with W=0 byte in PEC calculation.

Table 57 RSTDAA CCC - Broadcast

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x06 (Broadcast)								T	Sr[2] or P

1. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
2. Repeat start or repeat start with 7'h7E.

Table 58 RSTDAA CCC - Broadcast with PEC

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x06 (Broadcast)								T	
	PEC								T	Sr[2] or P

1. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
2. Repeat start or repeat start with 7'h7E.

4.13.4. SETAASA CCC

The SETAASA CCC is only supported when device is in I^2C mode; however, it still follows I3C SDR timings compliant to CCC definitions. In I^2C mode, when the host issues this CCC, to guarantee that this CCC is registered by the device without any error, the host shall limit the maximum speed operation for this CCC to 1 MHz. In I3C Basic mode, this CCC is ignored. When SETAASA CCC is registered by the SPD5 Hub, it updates **MR18**[5] = '1'and it takes in effect at the next Start operation (i.e., after STOP condition). Table 59 shows an example of a single SETAASA CCC.

SETAASA CCC does not support PEC function as device is in I^2C mode and there is no PEC function in I^2C mode.

Table 59 SETAASA CCC - Broadcast

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A	
	0x29 (Broadcast)								T	P

4.13.5. GETSTATUS CCC

The GETSTATUS CCC is supported in I3C Basic mode. In I^2C mode, this CCC is ignored (i.e., it is not executed internally and GETSTATUS CCC code is not acknowledged, and Host must do stop operation). Table 60 to Table 61 shows an example of a single GETSTATUS CCC. Table 62 shows the encoding definition for GETSTATUS CCC.

If PEC function is enabled, the PEC calculation starts with start or repeat start operation but does not include 7'h7E with W=0 byte in PEC calculation.

When the SPD5 Hub device responds to GETSTATUS CCC, after it completes the response, the PEC_Err, P_Err and Pending Interrupt Bits [3:0] do not automatically get cleared. The host must explicitly clear the appropriate status register through Clear command by writing '1' to corresponding register or by issuing Global Clear command. Once the SPD5 Hub device clears the appropriate status register, the PEC_Err, P_err and Pending Interrupt Bits [3:0] get cleared.

Copyright © 2024, NOVOSENSE

After Host issues Clear command, if the condition is still present, the device will again set the appropriate status register, sets the IBI status register to '1' and Pending Interrupt Bits [3:0] to '0001'.

Table 60 GETSTATUS CCC - Direct

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x90 (Direct)								T	
Sr	DevID [6:0]							R=1	A[1]	
	PEC_Err	0	0	0	0	0	0	0	T=1	
	0	0	P_Err	0	Pending Interrupt				T=0	Sr[2] or P

1. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
2. Repeat start or repeat start with 7'h7E.

Table 61 GETSTATUS CCC - Direct with PEC[1]

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[2]	
	0x90 (Direct)								T	
	PEC								T	
Sr	DevID [6:0]							R=1	A[1]	
	PEC_Err	0	0	0	0	0	0	0	T=1	
	0	0	P_Err	0	Pending Interrupt				T=1	
	PEC								T=0	Sr[3] or P

1. GETSTATUS CCC with PEC check is only supported in I3C Basic mode.
2. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
3. Repeat start or repeat start with 7'h7E.

Table 62 GETSTATUS CCC Byte Encoding

Bit	Encoding	Notes
PEC_Err	0 = No Error 1 = PEC Error occurred	This register is cleared when host issues clear command to **MR20**[1] for PEC error.
P_Err	0 = No Error 1 = Protocol Error; Parity Error occurred	This register is cleared when host issues clear command to **MR20**[0] for Parity error.
Pending Interrupt	0000 = No Pending Interrupt 0001 = Pending Interrupt All other encodings are reserved	This register is cleared when host issues clear command to any appropriate device status register that causes IBI status register to get cleared.

Copyright © 2024, NOVOSENSE

4.13.6. DEVCAP CCC

The DEVCAP CCC is only supported after device is put in I3C Basic mode. In I^2C mode, it is illegal for Host to issue this CCC. Table 63 to Table 64 shows an example of a single DEVCAP CCC. Table 65 shows the encoding definition for DEVCAP CCC.

If PEC function is enabled, the PEC calculation starts with start or repeat start operation but does not include 7'h7E with W=0 byte in PEC calculation.

Table 63 DEVCAP CCC - Direct

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0xE0 (Direct)								T	
Sr	DevID [6:0]							R=1	A[1]	
	MSB (Each bit defines capability)								T=1	
	LSB (Each bit defines capability)								T=0	Sr[2] or P

1. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
2. Repeat start or repeat start with 7'h7E.

Table 64 DEVCAP CCC - Direct with PEC

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x90 (Direct)								T	
	PEC								T	
Sr	DevID[6:0]							R=1	A[1]	
	MSB (Each bit defines capability)								T=1	
	LSB (Each bit defines capability)								T=1	
	PEC								T=0	Sr[3] or P

1. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
2. Repeat start or repeat start with 7'h7E.

Table 65 DEVCAP CCC Byte Encoding

Bit	Encoding	Notes
MSB [7]	RFU	Coded as '0'
MSB [6]	RFU	Coded as '0'
MSB [5]	RFU	Coded as '0'
MSB [4]	RFU	Coded as '0'
MSB [3]	RFU	Coded as '0'
MSB [2]	0 = No Support for Timer based Reset 1 = Supports Timer based Reset	SPD5 Hub hard codes to '1'

Copyright © 2024, NOVOSENSE

MSB [1:0]	RFU	Coded as '0'
LSB [7:0]	RFU	Coded as '0'

4.13.7. SETHID CCC

The SETHID CCC is supported only when device is in I²C mode. In I²C mode, when the host issues this CCC, to guarantee that this CCC is registered by the device without any error, the host shall limit the maximum speed operation for this CCC to 1 MHz. In I3C Basic mode, it is illegal for Host to issue this CCC. When SETHID CCC is registered by the SPD5 Hub, it stops 3-bit HID translation. Table 66 shows an example of a single SETHID CCC. The host must send all '0' in the data byte followed by 'T' bit. The SPD5 Hub device forwards bits [7:4, 0] to local devices behind the SPD5 Hub as it receives from the host. The SPD5 Hub device substitutes its own 3-bit HID code in bits [3:1] and forwards it to the local devices behind the SPD5 Hub. The SPD5 Hub device also re-calculates the parity information and forwards the updated parity information in 'T' bit. As the device is in I²C mode when SETHID CCC is issued, the PEC function is not supported.

The host may issue SETHID CCC more than one time.

Table 66 SETHID CCC - Broadcast

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A	
	0x61 (Broadcast)								T	
	0	0	0	0	0	0	0	0	T	P

4.13.8. DEVCTRL CCC

On a typical I3C Basic bus there can be up to 120 devices. For DDR5 DIMM application environment, there are up to 8 SPD5 Hub devices and behind each SPD5 Hub device, there are 4 local target devices totaling up to 40 or more devices on I3C Basic bus. For certain operation such as enable or disable functions that are common to all devices (i.e., Packet Error Check), the host must go through one device at a time which takes significant amount of time at initial power up. Further, it requires additional complexity on the host because it must speak different protocol depending on how it may access the device until all devices are configured identically.

To help expedite this configuration operation and to simplify the host complexity, the device supports the DEVCTRL CCC. The DEVCTRL CCC is supported either in I²C mode or I3C Basic mode of operation. In I²C mode, when the host issues this CCC, to guarantee that this CCC is registered by the device without any error, the host shall limit the maximum speed operation for this CCC to 1 MHz. Table 67 to Table 68 shows an example of a single DEVCTRL CCC.

If PEC function is enabled, the PEC calculation starts with start or repeat start operation but does not include 7'h7E with W=0 byte in PEC calculation.

The host shall pay attention to DEVCTRL CCC. The DEVCTRL CCC is limited to SPD5 Hub device's volatile register space only and should not be used for SPD5 Hub's NVM operation. If DEVCTRL CCC is used to access device specific registers (e.g., RegMod = '1'), the host shall still follow any device specific register restriction. For example, if device specific register requires stop operation for device to take in the effect of the setting, the host must also use stop operation when using DEVCTRL CCC to access device specific register.

In I²C mode, DEVCTRL CCC must be limited to 1 byte addressing mode for SPD5 Hub device (i.e., **MR11**[3] = '0').

Copyright © 2024, NOVOSENSE

Table 67 DEVCTRL CCC - Broadcast

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	AddrMask [2:0]			StartOffset [1:0]		PEC BL [1:0]		RegMod	T	
	DevID [6:0]							0	T[2]	
	Byte 0 Data Payload								T	
	Byte 1 Data Payload								T	
	Byte 2 Data Payload								T	
	Byte 3 Data Payload								T	Sr[3] or P

1. The SPD5 Hub NACKs if there is a parity error in a previous transaction when host performs consecutive transactions with repeat start.
2. An exception is made for DEVCTRL CCC where device does report a parity error when it determines 7-bit device select code issued by the host does not match with its own device code. If 7-bit device select code does not match but if parity is still valid, the device does not check for parity error in subsequent bytes; ignores the entire packet and waits until stop or next repeat start operation.
3. Repeat start or repeat start with 7'h7E.

Table 68 DEVCTRL CCC - Broadcast with PEC[1]

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S or Sr	1	1	1	1	1	1	0	W=0	A[2]	
	0x62 (Broadcast)								T	
	AddrMask [2:0]			StartOffset [1:0]		PEC BL [1:0]		RegMod	T	
	DevID [6:0]							0	T[3]	
	Byte 0 Data Payload								T	
	Byte 1 Data Payload								T	
	Byte 2 Data Payload								T	
	Byte 3 Data Payload								T	
	PEC								T	Sr[4] or P

1. DEVCTRL CCC with PEC check is only supported in I3C Basic mode.
2. The SPD5 Hub NACKs if there is a parity or PEC error in a previous transaction when host performs consecutive transactions with repeat start.
3. An exception is made for DEVCTRL CCC where device does report a parity error when it determines 7-bit device select code issued by the host does not match with its own device code. The device does not check for PEC as all subsequent bytes are discarded due to parity error. If 7-bit device select code does not match but if parity is still valid, the device does not check for parity error in subsequent bytes; ignores the entire packet and waits until stop or next repeat start operation.
4. Repeat start or repeat start with 7'h7E.

Copyright © 2024, NOVOSENSE

Table 69 DEVCTRL CCC Command Definition

Parameter	Definition
AddrMask [2:0]	Broadcast, Unicast or Multicast Command Selection 000 = Unicast Command; SPD5 Hub device responds if DevID [6:0] field matches with SPD5 Hub device's own 7-bit address (4-bit LID + 3-bit HID) 011 = Multicast Command; SPD5 Hub device and possible other device respond if DevID [6:3] field matches with SPD5 Hub device's own 4-bit LID address 111 = Broadcast Command; All devices respond to this command All other encodings are reserved.
StartOffset [1:0]	Only applicable if RegMod = '0' Identifies the starting Byte (Byte 0 or Byte 1 or Byte 2 or Byte 3) for DEVCTRL CCC. Host can start at any Byte (from Byte 0 to Byte 3) and has continuous access to next byte until stop operation. If Byte 3 is reached, the host is responsible for applying stop operation. 00 = Byte 0 01 = Byte 1 10 = Byte 2 11 = Byte 3
PEC BL [1:0]	Only applicable if RegMod = '0' and PEC function is enabled. Identifies the burst length just for this DEVCTRL CCC. The device uses the setting in this field to know when the PEC byte is expected after the data bytes. 00 = 1 Byte 01 = 2 Byte 10 = 3 Byte 11 = 4 Byte
RegMod	Identifies if DEVCTRL is going to be used for General Registers as identified in Byte 0 to Byte 3 or device specific address offset register. 0 = Access to General Registers in Byte 0 to Byte 3 (i.e. StartOffset [1:0] = Valid) 1 = Device Specific Offset Address (i.e. StartOffset [1:0] and PECBL [1:0] is a don't care and does not apply). The host shall NOT use RegMod = '1' with Broadcast Command if there are different types of devices on the I3C Basic bus.
DevID[6:0]	Identifies 7-bit device address. Device responds to DEVCTRL CCC data packet depending on AddrMask[2:0]. If AddrMask[2:0] = '111', DevID [6:0] is a don't care and device always responds. If AddrMask[2:0] = '000', DevID [6:0] must match for device to respond. If AddrMask[2:0] = '011', DevID [6:3] must match for device to respond. DevID [2:0] is don't care. For any other codes for AddrMask [2:0], the device always NACKs.

Copyright © 2024, NOVOSENSE

Table 70 DEVCTRL CCC Data Payload Definition

Byte	Bit	Function	Definition	Comment
Byte 0	[7]	PEC Enable	0 = Disable 1 = Enable	**MR18** [7] is updated
	[6]	Parity Disable	0 = Disable 1 = Enable	**MR18** [6] is updated
	[5:2]	RFU	RFU	
	[1]	RSVD	0 = RSVD 1 = RSVD	SPD5 Hub device always ignores this bit.
	[0]	RFU	RFU	
Byte 1	[7:4]	RFU	RFU	
	[3]	Global and IBI Clear	0 = No Action 1 = Clear All Event and pending IBI[1]	**MR27** [7] is updated.
	[2:0]	RFU	RFU	
Byte 2	[7:0]	RFU	RFU	
Byte 3	[7:0]	RFU	RFU	

1. After Target device clears the event, the device can still have certain registers set to '1' if the event is still present in which case, the device will generate an IBI again at the next opportunity.

DEVCTRL CCC Examples - RegMod = '0'

Table 71 shows an example of DEVCTRL CCC data packet. It assumes that all devices on the bus are already in I3C Basic mode with PEC function disabled and parity function enabled. In this example, the host uses DEVCTRL CCC as Multicast command. Host sends Multicast command to all devices with 4-bit LID code of '1001' on I3C Basic bus to do VR Enable followed by all devices with 4-bit LID code of '0110' to disable parity function. The host sends AddrMask = '011' to indicate multicast command with DevID [6:3] match; StartOffset = '00' to indicate starting Byte 0 and RegMod = '0' to indicate general register. Upon receiving this command, all devices with DevID [6:3] that matches to '1001' will do the VR Enable command and DevID [6:3] that matches to '0110' with disable the parity function.

Table 71 DEVCTRL CCC Example - Multicast Command to '1001' and '0110' Devices

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	011			00		00		0	T	
	1001 000							0	T	
	0000 0010								T	
Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	011			00		00		0	T	
	0110 000							0	T	

Copyright © 2024, NOVOSENSE

	0100 0000	T	P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation.

Table 72 shows an example of DEVCTRL CCC data packet. It assumes that all devices on the bus are already in I3C Basic mode with PEC function disabled and parity function enabled. In this example, the host uses DEVCTRL CCC as broadcast command to enable PEC function. The host sends AddrMask = '111' to indicate Broadcast command; StartOffset = '00' to indicate starting Byte 0 and RegMod = '0' to indicate general register. Upon receiving this command, all devices will enable PEC function.

Table 72 DEVCTRL CCC Example - Broadcast Command to all Devices

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	111			00		00		0	T	
	0000 000							0	T	
	1000 0000								T	P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation.

Table 73 shows an example of DEVCTRL CCC data packet. It assumes that all devices on the bus are already in I3C Basic mode with PEC function disabled and parity function enabled. In this example, the host uses DEVCTRL CCC as unitcast command to enable VR on DIMM5. The host sends AddrMask = '000' to indicate Unicast command; StartOffset = '00' to indicate starting Byte 0 and RegMod = '0' to indicate general register. Upon receiving this command, PMIC on DIMM5 will enable its regulator.

Table 73 DEVCTRL CCC Example - Unicast Command to PMIC on DIMM5

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	000			00		00		0	T	
	1001 101							0	T	
	0000 0010								T	P

1. See Figure 10 to see how the transition occurs from target open-drain (ACK) to host push-pull operation.

DEVCTRL CCC Examples - RegMod = '1'

Table 74 shows an example of DEVCTRL CCC data packet for the purpose of configuring device specific address offset register. It assumes that all devices on the bus are already in I3C Basic mode with PEC function enabled and parity function enabled. In this example, the host sends Multicast command to all devices with 4-bit LID code of '0010' on the I3C Basic bus to write to address offset of 0x1C and 0x1D with data 0xFF and 0x55 respectively followed by all devices with 4-bit LID of '1001' on the I3C Basic bus to write to address offset of 0x15 with data 0x78.

The PEC calculation starts with start or repeat start operation but does not include 7'h7E with W=0 byte in PEC calculation.

Copyright © 2024, NOVOSENSE

Table 74 DEVCTRL CCC Example - Multicast Command to '0010' and '1001' Devices

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	011			00		00		1	T	
	0010 000							0	T	
	0001 1100 (address offset 0x1C)								T	
	0010 0000 (CMD field = 2 bytes of data)								T	
	1111 1111 (data)								T	
	0101 0101 (data)								T	
	PEC								T	
Sr	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	011			00		00		1	T	
	1001 000							0	T	
	0001 0101 (address offset 0x15)								T	
	0000 0000 (CMD field = 1 byte of data)								T	
	0111 1000 (data)								T	
	PEC								T	P

Table 75 shows an example of DEVCTRL CCC data packet for the purpose of configuring device specific address offset register. It assumes that all devices on the bus are already in I3C Basic mode with PEC function disabled and parity function enabled. In this example, the host sends Multicast command to all devices with 4-bit LID code of '1001' on the I3C Basic bus to write to address offset of 0x13 with data 0xFF and it continues to write data 0x01 to the next address.

Table 75 DEVCTRL CCC Example - Multicast Command to '1001' Devices

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	1	1	1	1	1	0	W=0	A[1]	
	0x62 (Broadcast)								T	
	011			00		00		1	T	
	1001 000							0	T	
	0001 0011 (address offset 0x13)								T	
	1111 1111 (data)								T	
	0000 0001 (data)								T	P

1. See **Figure 10** to see how the transition occurs from target open-drain (ACK) to host push-pull operation.

Copyright © 2024, NOVOSENSE

5. In Band Interrupt (IBI)

In I²C mode, in band interrupt function is not supported. Only I3C Basic mode supports in band interrupt function.

5.1. Enabling and Disabling In Band Event Interrupt Function

By default, all interrupt sources are disabled (i.e., set to '0'). The host may enable following interrupts in the SPD5 Hub device. Once enabled, the SPD5 Hub device sends an IBI when that event occurs.

1. Error Interrupt Enable in **MR27**[4]:

- When **MR27** [4] = '1', the device sends the IBI at next available opportunity when any of the register bit in **MR52** [7:5, 1:0] is set to '1' and sets **MR48** [7] = '1' and updates Pending Interrupt Bits [3:0] = '0001' for GETSTATUS CCC.
- When **MR27** [4] = '0', the device does not send the IBI regardless of the register bit status in **MR52** [7:5, 1:0]. However, the device does set **MR48** [7] = '1' and updates Pending Interrupt Bits [3:0] = '0001' for GETSTATUS CCC.

2. Temperature Sensor Interrupt Enable in **MR27** [3:0]: The host can set any combination of register bits to '1':

- When any of the register bits in **MR27** [3:0] = '1' and if **MR27** [4] = '1', the device sends the IBI at next available opportunity when the corresponding register bit in **MR51** [3:0] is set to '1' and sets **MR48** [7] = '1' and updates Pending Interrupt Bits [3:0] = '0001' for GETSTATUS CCC.
- When any of the register bits in **MR27** [3:0] = '0' or **MR27** [4] = '0', the device does not send the interrupt regardless of the corresponding register bit status in **MR51** [3:0]. However, the device does set **MR48** [7] = '1' and updates Pending Interrupt Bits [3:0] = '0001' for GETSTATUS CCC if any of the bits in **MR27** [3:0] = '1' and **MR27** [4] = '0'.

5.2. Mechanics of Interrupt Generation - SPD5 Hub Device

Event interrupts may be generated by the SPD5 Hub device if IBI is enabled. When there is a pending interrupt (i.e. **MR48** [7] = '1 and **MR27** [4] = '1'), the SPD5 Hub requests an interrupt after detecting start condition by transmitting its 7-bit binary address (LID bits followed by HID bits) followed by R/W = '1' on the SDA bus serially (synchronized by SCL falling transitions).

If the SPD5 Hub detects no START condition but if the host to the SPD5 Hub device bus (HSDA and HSCL) has been inactive (no edges seen) for t_{AVAL} period, then the SPD5 Hub device may assert HSDA low by t_{IBI_ISSUE} time to request an interrupt. When the SPD5 Hub device requests an interrupt, the host toggles the HSCL. The SPD5 Hub device transmits its 7-bit binary address, '1010' followed by 3 HID bits, and then sets the R/W bit = '1'.

When the SPD5 Hub device requests an interrupt, the host may take one of the two actions below.

- The host sends ACK on 9th bit to accept the interrupt request. At this point, if the SPD5 Hub confirms that it has won the arbitration, the SPD5 Hub device transmits the IBI payload as shown in Table 76 and Table 77 for PEC disabled and PEC enabled configuration, respectively. See Figure 15. It just shows only first two data bits of the MDB byte to illustrate the timing. The interrupt payload contains MDB followed by 8-bit register contents of **MR51** and **MR52** in order. The host then issues the STOP command. Note the timing waveform in Figure 15. The host then accepts the IBI payload if it sends an ACK on 9th bit to accept the interrupt request. The host can interrupt the IBI payload at 'T'. If host stops the IBI payload at 'T' bit in the middle of payload, the SPD5 Hub device retains the IBI status flag **MR48** [7] = '1' and Pending Interrupt Bits [3:0] internally and waits for the next opportunity to request an interrupt. If the SPD5 Hub device successfully transmits the entire IBI payload, it then clears IBI status flag **MR48** [7] = '0' and Pending Interrupt Bits [3:0] = '0000' on its own and does not request for an IBI again unless there is another different event occurs; for another same event, the device does not request for an IBI.
- The host sends NACK on the 9th bit as shown in Figure 16 followed by a STOP command. In this case, the SPD5 Hub device does not transmit the IBI payload and waits for the next opportunity to request an interrupt. At this point, though Host sent a NACK, it does have a knowledge of which SPD5 Hub device sent the IBI request. The SPD5 Hub retains the IBI status flag **MR48** [7] = '1' and Pending Interrupt Bits [3:0] = '0001'.

Table 76 Hub IBI Payload Packet; PEC is Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	0	1	0	HID			R=1	A[1]	
	MDB = 0x00								T=1	
	MR51 [7:0]								T=1	

Copyright © 2024, NOVOSENSE

	MR52 [7:0]	T=0[2]	P

1. Figure 11 shows the transition occurs from host open-drain (ACK) to target push-pull operation (1st bit of MDB, bit [7]).
2. See Figure 14 to see how target device ends the operation followed by host stop operation.

Table 77 Hub IBI Payload Packet; PEC is Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	1	0	1	0	HID			R=1	A[1]	
	MDB = 0x00								T=1	
	MR51 [7:0]								T=1	
	MR52 [7:0]								T=1	
	PEC								T=0[2]	P

1. Figure 11 shows the transition occurs from host open-drain (ACK) to target push-pull operation (1st bit of MDB, bit [7]).
2. See Figure 14 to see how target device ends the operation followed by host stop operation.

Figure 15 SPD5 Hub Interrupt; Host Ack Followed by SPD5 Hub Device IBI Payload

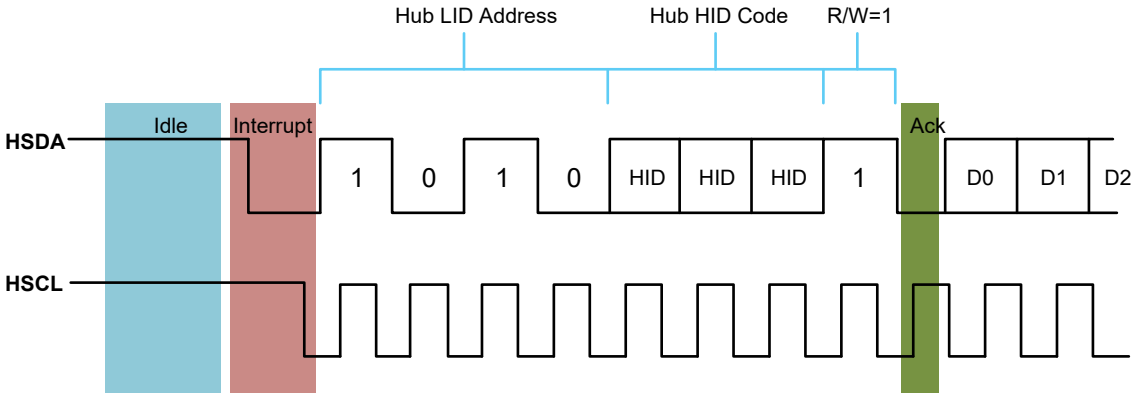
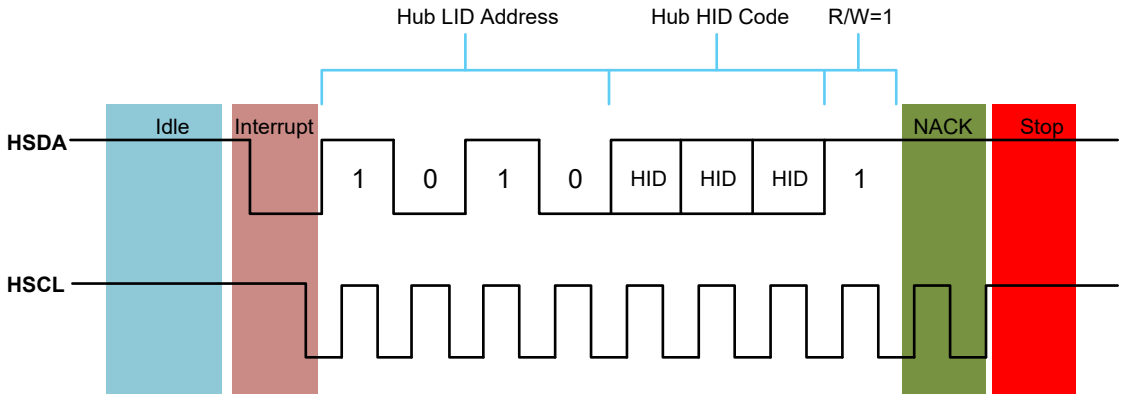


Figure 16 SPD5 Hub Interrupt; Host NACK followed by STOP



5.3. Mechanics of Interrupt Generation - Local Target Device

Event interrupts may be generated by the local device if IBI is enabled. When there is a pending interrupt in any local device and if IBI is enabled, the local device requests an interrupt after detecting start condition by transmitting its 7-bit binary address (LID bits followed by HID bits) followed by R/W = '1' on the SDA bus serially (synchronized by SCL falling transitions).

If the local device detects no START condition but if the host to the local target device (through the SPD5 Hub device) bus (LSDA and LSCL) has been inactive (no edges seen) for t_{AVAL} period, then any local device may assert LSDA low by t_{IBI_ISSUE} time to request an interrupt. When the local device requests an interrupt, the SPD5 Hub device propagates the LSDA to HSDA to host. The host toggles

Copyright © 2024, NOVOSENSE

the HSCL. The local device transmits its 7-bit binary address (LID bits followed by HID bits) followed by R/W bit = '1' to the SPD5 Hub device. This is shown in Figure 17 top waveform.

The SPD5 Hub device forwards LID bits it receives from the local device to the host. The SPD5 Hub device substitutes its own HID code in place of the HID bits it receives from the local device and sends to the host if Host has not issued SETHID CCC previously. This is shown in Figure 17 middle waveform. The SPD5 Hub device forwards HID bits it receives from the local device to the host if Host has issued SETHID CCC previously. This is shown in Figure 17 bottom waveform. The SPD5 Hub device forwards the R/W bit = '1' to the host.

When the local device requests an interrupt, the host may take one of the two actions below.

- The host sends ACK on 9th bit to accept the interrupt request. At this point, if the local device confirms that it has won the arbitration, the local device transmits the IBI payload as shown in Table 78 and Table 79 for PEC disabled and PEC enabled configuration respectively. See Figure 17. It just shows only first two data bits of MDB byte to illustrate the timing. The interrupt payload contains MDB followed by appropriate Target device error register contents in order. The host then issues the STOP command. Note the timing waveform in Figure 17. The host then accepts the IBI payload if it sends an ACK on 9th bit to accept the interrupt request. The host can interrupt the IBI payload at 'T' bit. If host stops the IBI payload at 'T' bit in the middle of payload, the local device retains the IBI status flag and Pending Interrupt Bits [3:0] internally and waits for the next opportunity to request an interrupt. If the local device successfully transmits the entire IBI payload, it then clears IBI status flag and Pending Interrupt Bits [3:0] = '0000' on its own and does not request for an IBI again unless there is another different event occurs; for another same event, the device does not request for an IBI.
- The host sends NACK on the 9th bit as shown in Figure 18 followed by a STOP command. In this case, the local device does not transmit the IBI payload and waits for the next opportunity to request an interrupt. At this point, though Host sent a NACK, it does have a knowledge of which local device sent the IBI request. The local device retains the IBI status flag and Pending Interrupt Bits [3:0] = '0001'.

Table 78 Target Device IBI Payload Packet; PEC is Disabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	LID				HID			R=1	A[1]	
	MDB = 0x00								T=1	
	First Error Code Byte								T=1	
	Second Error Code Byte								T=1	
	…								T=0[2]	P

1. Figure 11 shows the transition occurs from host open-drain (ACK) to target push-pull operation (1st bit of MDB, bit [7]).
2. See Figure 14 to see how target device ends the operation followed by host stop operation.

Table 79 Target Device IBI Payload Packet; PEC is Enabled

Start	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	A/N/T	Stop
S	LID				HID			R=1	A[1]	
	MDB = 0x00								T=1	
	First Error Code Byte								T=1	
	Second Error Code Byte								T=1	
	…								T=1	
	PEC								T=0[2]	P

1. Figure 11 shows how the transition occurs from host open-drain (ACK) to target push-pull operation (1st bit of MDB, bit [7]).
2. See Figure 14 to see how target device ends the operation followed by host stop operation.

Copyright © 2024, NOVOSENSE

Figure 17 Local Device Interrupt, Host Ack followed by Target Device IBI Payload

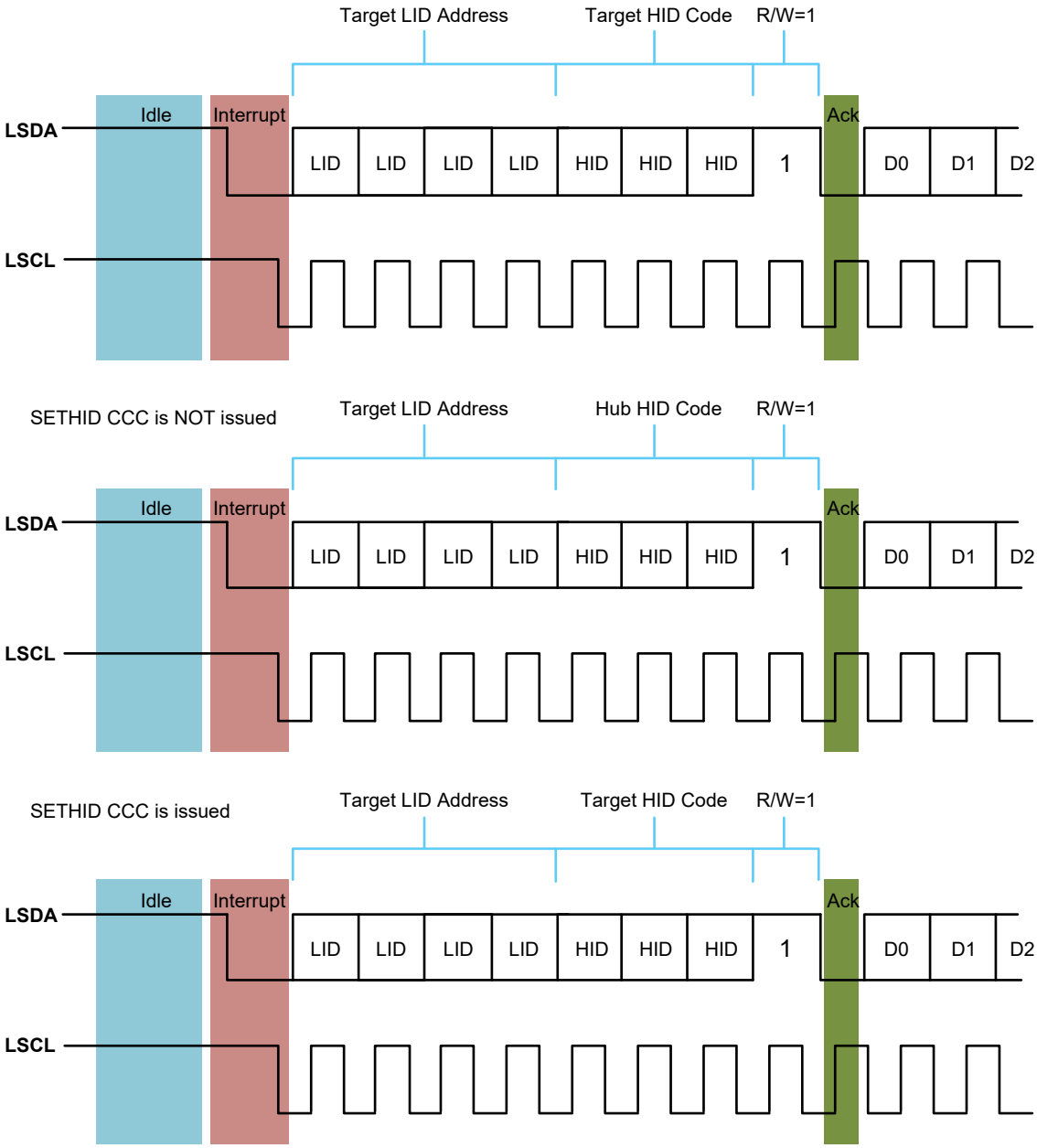
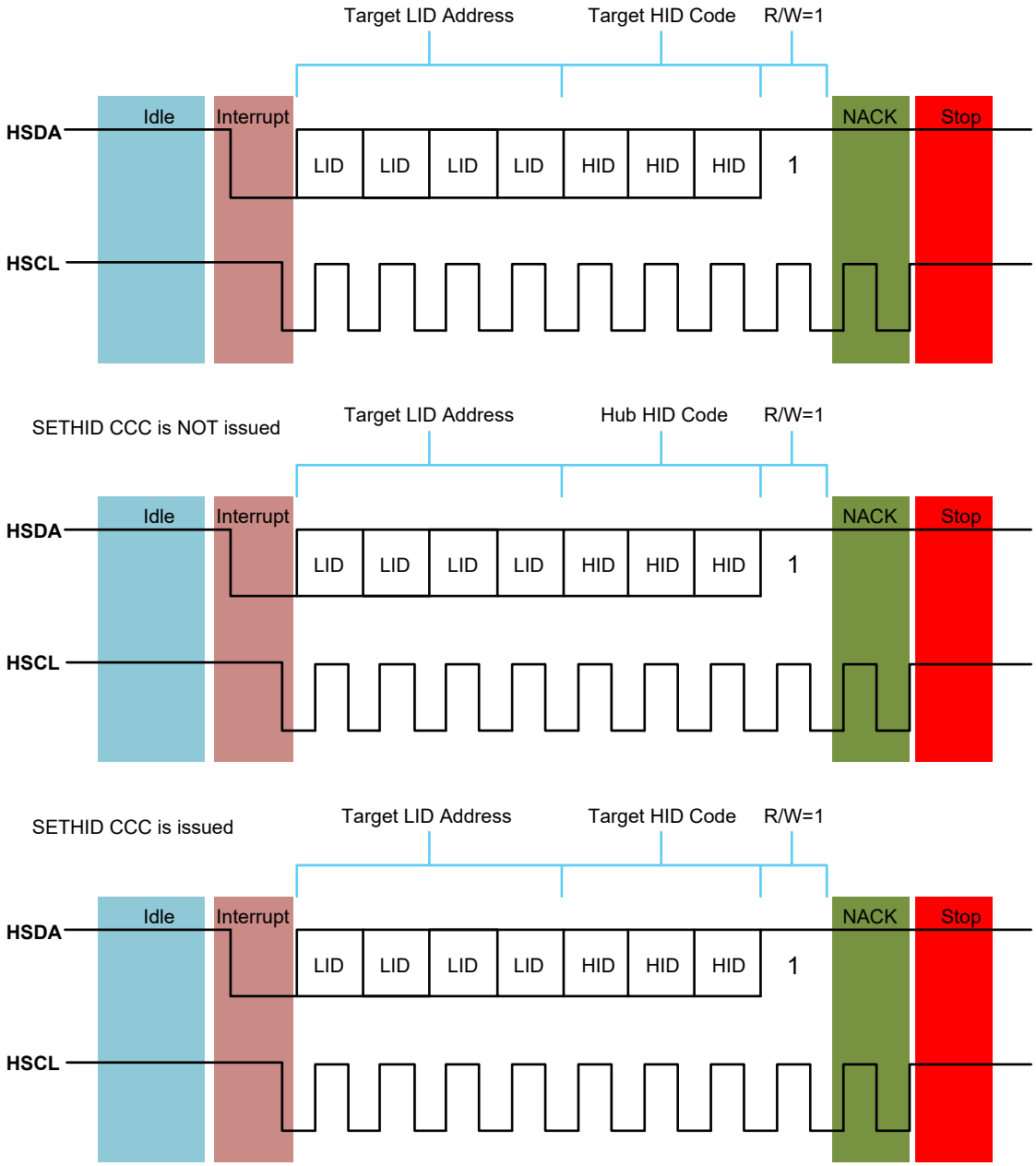


Figure 18 Local Device Interrupt; Host NACK followed by STOP



5.4. Interrupt Arbitration; SETHIDD CCC is Not Issued by Host

As there are multiple local target devices behind the SPD5 Hub and there are multiple SPD5 Hub devices on I3C Basic bus, multiple devices may request an interrupt when the I3C Basic bus is inactive for t_{AVAL} period. Arbitration process is required.

5.4.1. Interrupt Arbitration - Among SPD5 Hub Devices

This section explains the interrupt arbitration among the SPD5 Hub devices only. There are up to 8 SPD5 Hub devices on the I3C Basic bus.

As all SPD5 Hub devices have the same 4-bit LID code of '1010', the arbitration is always won by the lower HID code. For example, if one SPD5 Hub device has HID code of '000' and other SPD5 Hub device has a HID code of '011', through the arbitration process, the HID code of '000' wins. The SPD5 Hub device with a HID code of '011 must release the bus and wait for next opportunity to request an interrupt. Table 80 shows the arbitration priority based on the HID code for the SPD5 Hub device.

Copyright © 2024, NOVOSENSE

Table 80 Interrupt Arbitration - Among SPD5 Hub Devices

Hub Device LID Code	Hub Device HID Code	Arbitration Priority
1010	000	1
1010	001	2
1010	010	3
1010	011	4
1010	100	5
1010	101	6
1010	110	7
1010	111	8

5.4.2. Interrupt Arbitration - Among Local Target Devices behind one SPD5 Hub Devices

This section explains the interrupt arbitration among the local devices only behind the SPD5 Hub. There are up to 13 local devices behind the SPD5 Hub.

As all local target devices behind the SPD5 Hub devices have the same 3-bit HID code, hence the arbitration is always won by the 4-bit LID code. For example, if one local target device has LID code of '1001' and other local target device has a LID code of '0010', through the arbitration process, the LID code of '0010' wins. The local device with a LID code of '1001' must release the bus and wait for next opportunity to request an interrupt. Table 81 shows the arbitration priority based on the LID code for the local devices. The Green color cells in Table 81 are the likely devices that will be on a standard DDR5 RDIMM or DDR5 LRDIMM.

During the interrupt arbitration phase, the SPD5 Hub device forwards the winning 4-bit LID code one digit at a time that it receives from the local target devices to the host.

The SPD5 Hub device discards the 3-bit '111' HID code received from the local target device. Instead, the SPD5 Hub device forwards its own unique 3-bit HID code one digit at a time to the host. As a result, the host can identify the winning device based on the 4-bit LID code (Local Target device) and 3-bit HID code (the DIMM). Also, when the SPD5 Hub device substitutes its own unique 3-bit HID code to the host, its own receiver will see that at the input, and it compares each 3 bits one at a time as it if the host sent those 3 bits. If there is a match, the SPD5 Hub device forwards '1' to the local device interface and if there is a mismatch, the SPD5 Hub device forwards '0' to the local device interface. Because the local target device sends '111' as its 3 bits HID code, the local device knows that it won the arbitration.

Copyright © 2024, NOVOSENSE

Table 81 Interrupt Arbitration - Among Local Target Devices

Device	Target Device LID Code	Target Device HID Code	Arbitration Priority
N/A	0000	N/A	N/A
RFU	0001	111	1
TS0	0010	111	2
RFU	0011	111	3
RFU	0100	111	4
RFU	0101	111	5
TS1	0110	111	6
RFU	0111	111	7
PMIC1	1000	111	8
PMIC0	1001	111	9
SPD Hub	1010	HID	N/A
RCD	1011	111	10
PMIC2	1100	111	11
RFU	1101	111	12
RFU	1110	111	13
N/A	1111	N/A	N/A

5.4.3. Interrupt Arbitration - Between SPD5 Hub Device and Local Target Devices behind the Hub

This section explains the interrupt arbitration among the SPD5 Hub device and all local devices behind that SPD5 Hub. There are up to 13 local devices behind each SPD5 Hub.

As the SPD5 Hub device LID code of '1010', any local target devices with a lower LID code always wins the arbitration process as shown in Table 81. The local target devices with a higher LID code than '1010', the SPD5 Hub device wins the arbitration process. As an example, for a typical DDR5 RDIMM/LRDIMM, the RCD always has the lowest priority for winning the interrupt arbitration.

5.4.4. Interrupt Arbitration - Among Local Target Devices behind Different SPD5 Hub Devices

This section explains the interrupt arbitration among the SPD5 Hub devices and all local devices behind any SPD5 Hub devices. There are up to 8 SPD5 Hub devices on the I3C bus and up to 13 local devices behind each SPD5 Hub.

The arbitration process is hybrid of **Section 5.4.1** to **Section 5.4.3**. The device with the lowest 4-bit LID code across all local target devices and across all SPD5 Hub devices always wins the arbitration process. The HID code for that lowest LID code represents the SPD5 Hub device code.

Table 82 shows four examples. In each example, the target LID code column represents local target device behind Hub which has HID code value of '111'; the Hub HID code column represents the SPD5 Hub device which has LID code of '1010' followed by its own unique HID code; the winning device column represent the final winner among all devices (either SPD Hub device or local target devices) during the arbitration phase.

Example 1:

There are total of 5 devices (4 local target devices and 1 Hub device) that are requesting an interrupt. These 5 devices are shown in the RED color text. The devices that are in the black color text are not requesting an interrupt. The winning device is the local device LID code of '0010' and HID code of '111'. This is because the LID code of '0010' is the lowest among three other local target device code and its HID code is '111'.

Copyright © 2024, NOVOSENSE

Example 2:

There are total of 4 devices (2 local target devices and 2 Hub devices) that are requesting an interrupt. These 4 devices are shown in the RED color text. The devices that are in the black color text are not requesting an interrupt. The winning device is the local device LID code of '1001' and HID code of '100'. This is because the LID code of '1001' is the lowest among one other local target device code and its HID code is '100'.

Example 3:

There are total of 3 devices (2 local target devices and 1 Hub device) that are requesting an interrupt. These 3 devices are shown in the RED color text. The devices that are in black color text are not requesting an interrupt. The wining device is the Hub device on DIMM0 with LID code of '1010' and HID code of '000'. This is because LID code '1010' is lower than two other local target device code and its HID code is '000'.

Example 4:

There are total of 2 devices (2 local target devices) that are requesting an interrupt. These 2 devices are shown in RED color text. The devices that are in black color text are not requesting an interrupt. This example is unique as two identical Target devices across two different DIMM devices are requesting an interrupt. The winning device is the local target device on DIMM 0 with LID code of '0010' and DIMM 0 HID code of '000'. This is because DIMM0 HID code '000' is lower than DIMM2 HID code of '010'. See **Section 5.4.2**.

5.4.5. Interrupt Arbitration - Between Host and Any SPD5 Hub or Any Local Target Devices behind Hub

When the bus is idle for t_{AVAL} time, any SPD5 Hub device or any local device behind the Hub can request an interrupt by pulling the SDA bus low.

In an uncommon but possible scenario would be that at the exact same time as when the SPD5 Hub or local target devices are requesting an interrupt, the host is starting an operation to the Hub or local target devices. When this happens, Host also gets involved in the arbitration process along with the Hub or the local target devices. During the arbitration phase, there will be always only one winning device, and it could be either the Hub or the local target device or the host.

If the host wins during the arbitration phase, it continues with normal operation. The losing Hub or local target device waits for next opportunity to send an interrupt.

If the host loses during the arbitration phase, it must let go of the bus. When the host loses during the arbitration, the host must let the Hub or local target device finish sending its 4-bit LID code followed by 3-bit HID code followed by R/W = '1'. At this point, during the 9th bit, the host has two options to take the action as noted below:

- Host sends an ACK to accept the interrupt and hence accepts the IBI payload from the winning SDP5 Hub or local target device. After the IBI payload, the host issues stop operation.
- Host sends a NACK followed by stop operation.

In a rare but still possible scenario would be that at the exact same time as when the Hub or local target device is requesting an interrupt, the host is starting an operation to that same exact Hub or local target device. When this happens, neither Host nor the Hub or local target device knows it is a winner until the 8th bit and Host always wins. This is because, the Hub or local target device sends R=1 (8th bit) during the interrupt. The host sets W=0 (8th bit) during the operation. As a result, the host wins and the Hub or local target device must let go of the bus and wait for the next opportunity to send an interrupt. This is shown as example 3 in Table 83.

Table 83 shows three examples. In each example, the target LID code column represents local target device behind Hub which has HID code value of '111'; the Hub HID code column represents the SPD5 Hub device which has LID code of '1010' followed by its own unique HID code; the winning device column represents the final winner among all devices (either Host or SPD5 Hub device or local target devices) during the arbitration phase.

In example 1, there are total of 5 devices (1 Hub device and 4 local target devices) are requesting an interrupt at exactly same time as when the host is starting an operation to hub device on DIMM 3 (1010 011). The winning device is local target device '0010 111' because it has the lower 4-bit LID code.

In example 2, there are total of 4 devices (2 Hub devices 2 local target devices) are requesting an interrupt at exactly same time as when the host is requesting an operation to the local target device on DIMM5 (0110 101). The host is the winner because its intended target device has the lower 4-bit LID code than devices that are requesting an interrupt.

In example 3, there is one 1 hub device on DIMM 2 is requesting an interrupt at exactly the same time as when host is requesting an operation to the same exact Hub device on DIMM2. In this case, the host is the winner because the 8-bit will be driven low by the host (W=0) while the Hub device drives it high (R=1) during the interrupt.

Copyright © 2024, NOVOSENSE

In an extreme rare but still possible scenario would be that at the exact same time as when the Hub or local target device is requesting an interrupt, the host is requesting a read operation with default read address pointer mode to the same exact Hub or local target device. When this happens, there is no winning device. This is the only time there is no winning device. This is because, the Hub or local target device sends R=1 (8th bit) during the interrupt and Host also sends R=1 for read request with default read address pointer mode. As a result, there is no winner because all devices, i.e., the host or the Hub or the local target devices, are waiting for other device to ACK. In this case, no devices will ACK. Since there is no ACK by any device, the host must time out and repeats the read request with repeat start. When it repeats the read request with repeat start, the Hub or local target device does not send an interrupt because of repeat start.

Table 82 Interrupt Arbitration - Among Local Target and SPD5 Hub Devices

Example 1			Example 2			Example 3			Example 4		
Target LID Code	Hub HID Code	Winning Device	Target LID Code	Hub HID Code	Winning Device	Target LID Code	Hub HID Code	Winning Device	Target LID Code	Hub HID Code	Winning Device
0010 0110 1001 1011	000		0010 0110 1001 1011	000		0010 0110 1001 1011	000	1010 000	0010 0110 1001 1011	000	0010 000
0010 0110 1001 1011	001		0010 0110 1001 1011	001		0010 0110 1001 1011	001		0010 0110 1001 1011	001	
0010 0110 1001 1011	010		0010 0110 1001 1011	010		0010 0110 1001 1011	010		0010 0110 1001 1011	010	
0010 0110 1001 1011	011		0010 0110 1001 1011	011		0010 0110 1001 1011	011		0010 0110 1001 1011	011	
0010 0110 1001 1011	100		0010 0110 1001 1011	100	1001 100	0010 0110 1001 1011	100		0010 0110 1001 1011	100	
0010 0110 1001 1011	101		0010 0110 1001 1011	101		0010 0110 1001 1011	101		0010 0110 1001 1011	101	
0010 0110 1001 1011	110		0010 0110 1001 1011	110		0010 0110 1001 1011	110		0010 0110 1001 1011	110	

Copyright © 2024, NOVOSENSE

0010 0110 1001 1011	111	0010 111	0010 0110 1001 1011	111		0010 0110 1001 1011	111		0010 0110 1001 1011	111	

Table 83 Interrupt Arbitration - Between Host and Local Target and SPD5 Hub Devices

Example 1				Example 2				Example 3			
Host Target Device	Target LID Code	Hub HID Code	Winning Device	Host Target Device	Target LID Code	Hub HID Code	Winning Device	Host Target Device	Target LID Code	Hub HID Code	Winning Device
1010 011	0010 0110 1001 1011	000		0110 101	0010 0110 1001 1011	000		1010 010	0010 0110 1001 1011	000	
	0010 0110 1001 1011	001			0010 0110 1001 1011	001			0010 0110 1001 1011	001	
	0010 0110 1001 1011	010			0010 0110 1001 1011	010			0010 0110 1001 1011	010	Host operation to 1010 010
	0010 0110 1001 1011	011			0010 0110 1001 1011	011			0010 0110 1001 1011	011	
	0010 0110 1001 1011	100			0010 0110 1001 1011	100			0010 0110 1001 1011	100	
	0010 0110 1001 1011	101			0010 0110 1001 1011	101	Host operation to 0110 101		0010 0110 1001 1011	101	
	0010 0110 1001 1011	110			0010 0110 1001 1011	110			0010 0110 1001 1011	110	
	0010	111	0011		0010	111			0010	111	

Copyright © 2024, NOVOSENSE

	0110 1001 1011		111		0110 1001 1011				0110 1001 1011		

5.5. Interrupt Arbitration; SETHID CCC is Issued by Host

The interrupt arbitration process works similar way as defined in **Section 5.4** but with a simplification. It is assumed that in this case, all target devices behind Hub across all DIMMs on I3C Basic bus have a unique 7-bit device select code. Further, it assumes that all devices on a DIMM have the same 3-bit HID code.

The SPD5 Hub device forwards the 4-bit LID code and 3-bit HID code one digit at a time that it receives from the local target devices to the host. The SPD5 Hub device's own receiver will see the same input and it forwards it back to the local device interface.

5.5.1. Interrupt Arbitration - Between SPD5 Hub Device and Local Target Devices behind Hub

On an any given DIMM, the arbitration is always won by the device that has the lowest 4-bit LID code since all devices on the DIMM has same 3-bit HID code.

5.5.2. Interrupt Arbitration - Between all SPD5 Hub Devices and all Local Target Devices behind Hub

Across multiple DIMMs, the arbitration is always won by the device that has the lowest 7-bit address (4-bit LID + 3-bit HID).

5.5.3. Interrupt Arbitration - Between Host and All Devices

In an uncommon but possible scenario would be that at the exact same time as when the target device is requesting an interrupt, the host is starting an operation to the target device. When this happens, Host also gets involved in the arbitration process along with the target devices. During the arbitration phase, there will be always only one winning device, and it could be either the host or the target device.

If the host wins during the arbitration phase, it continues with normal operation. The losing target device waits for next opportunity to send an interrupt.

If the host loses during the arbitration phase, it must let go of the bus. When Host loses during the arbitration, the host must let the target device finish sending their 4-bit LID code followed by 3-bit HID code followed by R/W = '1'. At this point, during the 9th bit, the host has two options to take the action as noted below:

- Host sends an ACK to accept the interrupt and hence accepts the IBI payload from the winning target device. After the IBI payload, the host issues stop operation.
- Host sends an NACK followed by stop operation.

In a rare but still possible scenario would be that at the exact same time as when the SPD5 Hub device is requesting an interrupt, the host is starting an operation to that same SPD5 Hub device. When this happens, neither Host or nor the SPD5 Hub device knows it is a winner until the 8th bit and Host always wins. This is because, the SPD5 hub device sends R=1 (8th bit) during the interrupt. The host sets W=0 (8th bit) during the operation. As a result, the host wins and the SPD5 Hub device must let go of the bus and wait for the next opportunity to send an interrupt.

In an extreme rare but still possible scenario would be that at the exact same time as when the SPD5 Hub device is requesting an interrupt, the host is requesting a read operation with default read address pointer mode to the SPD5 Hub device. When this happens, there is no winning device. This is the only time there is no winning device. This is because, the SPD5 Hub device sends R=1 (8th bit) during the interrupt and Host also sends R=1 for read request with default read address pointer mode. As a result, there is no winner because all devices, i.e., the host or the SPD5 Hub devices, are waiting for other device to ACK. In this case, no devices will ACK. Since there is no ACK by any device, the host must time out and repeats the read request with repeat start. When it repeats the read request with repeat start, the SPD5 Hub device does not send an interrupt because of repeat start.

5.6. Clearing Device Status and IBI Status Registers

The SPD5 Hub device provides the IBI status in **MR48**[7] by setting it to '1'. The SPD5 Hub device clears the IBI status register **MR48**[7] to '0' automatically when it sends a complete IBI (including payload and without interruption) and it also clears Pending Interrupt Bits [3:0] to '0000'. Once IBI status register is cleared, the SPD5 Hub does not request for an IBI again unless another event occurs.

The SPD5 Hub device provides the device status in **MR51** and **MR52** registers. The status information in **MR51** and **MR52** are latched and remains set even after the SPD5 Hub device sends IBI payload and clears the IBI status register **MR48**[7] to '0'. The host must explicitly clear the status register through clear command by writing '1' for appropriate status or by issuing a Global Clear command.

Copyright © 2024, NOVOSENSE

After Host issues Clear command, if the condition is no longer present, the SPD5 Hub device clears the appropriate status register, clears the IBI status register to '0' and Pending Interrupt Bits [3:0] to '0000' even if the SPD5 Hub device has not sent the IBI. After Host issues Clear command, if the condition is still present, the device will again set the appropriate status register, sets the IBI status register to '1' and Pending Interrupt Bits [3:0] to '0001' even if the device has already sent the IBI and entire IBI payload.

6. Error Check Function

6.1. Packet Error Check (PEC) Function

In I^2C mode, packet error checking is not supported. Only I3C Basic mode supports packet error checking.

The SPD5 Hub device implements an 8-bit Packet Error Check (PEC) which is appended at the end of all transactions if PEC is enabled through DEVCTRL CCC or by directly writing '1' to MR18 [7]. The PEC is a CRC-8 value calculated on all the message bytes except for start, repeated start, stop conditions or 'T' bits, ACK, and NACK and IBI header (7'h7E followed by W=0) bits.

The polynomial for CRC-8 calculations is:

$C(X) = X^8 + X^2 + X^1 + 1$

The seed value for PEC function is all zero.

When Host calculates PEC for SPD5 Hub device, it includes LID and HID bits followed by R/W bit.

6.2. Parity Error Check Function

In I^2C mode, parity error checking is not supported except for supported CCCs. Only I3C Basic mode supports parity error checking.

By default, when SPD5 Hub device is put in I3C Basic mode, parity function is automatically enabled. The host can disable the function after it is enabled. Host can disable the parity function with DEVCTRL CCC (RegMod = '0') or by directly writing '1' to MR18 [6]. When parity function is disabled, the SPD5 Hub device simply ignores the 'T' bit information from the host. The host may actually choose to compute the parity and send that information during 'T' bit or simply drive static low or high in 'T' bit.

The SPD5 Hub device implements ODD parity. If an odd number of bits in the byte are '1', the parity bit value is '0'.

If even number of bits in the byte are '1', the parity bit value is '1'. The host computes the parity and sends it during 'T' bit.

6.3. Packet Error Check and Parity Error Handling

There are two types of error checking done by the SPD5 Hub device: parity error checking and packet error checking. By default, the parity error checking is always enabled, and packet error checking is disabled when the SPD5 Hub device is put in I3C Basic mode. The host may enable the packet error checking at any time. The parity error is calculated for each byte. The host sends parity error information in 'T' bit.

I3C Basic defines TE0, TE1, TE2, TE3, TE4, TE5, TE6 error detection for target devices. Only TE1 and TE2 error detection is supported by the SPD5 Hub for parity checking. All other errors are not supported and not applicable.

6.3.1. Write Command Data Packet Error Handling - PEC Disabled

The SPD5 Hub device checks for the parity error for each byte in a packet except for the device select code byte that it receives from the host as shown in Table 22.

Write command - if no parity error:

- The SPD5 Hub device executes the command.

Write command - if parity error:

- The SPD5 Hub device discards the byte in the packet that had a parity error.
- The SPD5 Hub device discards all subsequent bytes in that packet until the stop operation. The SPD5 Hub device may or may not check parity for all sub-sequent bytes in that packet.
- Note that as the packet contains more than one byte, if first byte had no parity error but the second byte had a parity error, the SPD5 Hub device may or may not execute the first byte operation but second byte and all subsequent bytes operations are discarded.
- The SPD5 Hub device sets the **MR52**[0], **MR48**[7] and P_Err in GETSTATUS CCC to '1'; updates Pending Interrupt Bits [3:0] to '0001' and waits for the next opportunity to send an in band interrupt if IBI is enabled.

Copyright © 2024, NOVOSENSE

6.3.2. Read Command Data Packet Error Handling - PEC Disabled

The SPD5 Hub device checks for parity error for each byte in a packet except for the device select code byte that it receives from the host prior to repeat start as shown in Table 26.

The SPD5 Hub device does not compute the parity when it sends the data to the host. The host does not check for parity error for the bytes that SPD5 Hub device sends. The SPD5 Hub device sends Continuous (‘1’) or Stop (‘0’) information during ‘T’ bit.

Read command - If no parity error:

- The SPD5 Hub sends ACK back to the host when host performs start repeat operation.
- The SPD5 Hub device executes the command and sends the data as shown in Table 26.

Read Command - If parity error:

- The SPD5 Hub device discards the byte in the packet that had a parity error.
- The SPD5 Hub device discards second byte in that packet if the parity error occurred in first byte. The SPD5 Hub device may or may not check the parity for second byte in that packet.
- The SPD5 Hub sends NACK back to the host when host performs a Start Repeat operation. This is shown in the RED color cell in Table 26. The NACK represents either a parity error in one of the two bytes or that SPD5 Hub is not able to start the read operation. The host may re-try repeat start again. The host may do the repeat start as many times as it may desire. If the SPD5 Hub device NACKs due to parity error in a previous byte from the host, it will always NACK regardless of how many times Host tries repeat start.
- The SPD5 Hub does not send the data shown in Table 26 and instead expects Host to perform stop operation.
- The SPD5 Hub device sets **MR52**[0] and **MR48**[7] and P_Err in GETSTATUS CCC to ‘1’; updates Pending Interrupt Bits [3:0] in GETSTATUS to ‘0001’ and waits for the next opportunity to send an in band interrupt if IBI is enabled.

6.3.3. Write Command Data Packet Error Handling - PEC Enabled

The SPD5 Hub device checks for the parity error for each byte in a packet except for the device select code byte that it receives from the host as shown in Table 23. Further, the SPD5 Hub device checks for the packet error for the entire packet (from Start condition until last byte of Data) that receives from the host as shown in Table 23.

Write command - if no parity error:

- The SPD5 Hub device waits for the entire packet. If no error in packet, the SPD5 Hub device executes the command. If there is an error in the packet, the SPD5 Hub device discards the entire packet and does not execute that packet and waits for STOP, sets the **MR52**[1] and **MR48**[7] to ‘1’ and PEC_Err in GETSTATUS CCC to ‘1’ and updates Pending Interrupt Bits [3:0] in GETSTATUS CCC to ‘0001’ and waits for the next opportunity to send in band interrupt if IBI is enabled.

Write command - if parity error:

- The SPD5 Hub device discards that byte and the entire packet until stop operation.
- The SPD5 Hub device sets **MR52**[0] and **MR48**[7] and P_Err in GETSTATUS CCC to ‘1’; updates Pending Interrupt Bits [3:0] in GETSTATUS CCC to ‘0001’ and waits for the next opportunity to send in band interrupt if IBI is enabled.
- The SPD5 Hub device may or may not check the error for the packet. If the SPD5 Hub device checks for the packet error, likely it will detect an error in the packet and the device may also set **MR52**[1] and PEC_Err in GETSTATUS CCC to ‘1’ as well.

6.3.4. Read Command Data Packet Error Handling - PEC Enabled

The SPD5 Hub device checks for the parity error for each byte in a packet except for the device select code byte that it receives from the host prior to repeat start as shown in Table 27.

The SPD5 Hub device does not compute the parity when it sends the data to the host. It does not check for parity error for the bytes shown in Table 27. The SPD5 Hub device sends continuous (‘1’) or stop (‘0’) information during ‘T’ bit when SPD5 Hub device is sending the read data.

The SPD5 Hub device checks for the PEC error for a packet that it receives from the host from Start condition to repeat start condition (from first device select code followed by the address offset and CMD byte).

The SPD5 Hub device computes the PEC for the entire packet starting with repeat start (device select code and the data SPD5 Hub device transmits back to Host)

Read command - If no parity error and no PEC error:

- The SPD5 Hub sends ACK back to the host when host performs a start repeat operation.

Copyright © 2024, NOVOSENSE

- The SPD5 Hub device executes the command and sends the data as shown in Table 27.
- The SPD5 Hub computes PEC for the bytes (from start condition to PEC byte prior to repeat start) shown in Table 27.

Read command - if parity error or PEC error:

- The SPD5 Hub device discards the byte in the packet that had a parity error.
- The SPD5 Hub device discards second byte in that packet if a parity error occurred in first byte. The SPD5
- Hub device may or may not check parity for the second byte in that packet.
- The SPD5 Hub device discards the packet if there is a PEC error.
- The SPD5 Hub sends NACK back to the host when host performs start repeat operation. This is shown in the RED color cell in Table 27. The NACK represents either PEC error or a parity error in one of the three bytes or that SPD5 Hub is not able to start the read operation. The host may re-try repeat start again. The host may do the repeat start as many times as possible it may desire. The PEC calculation by SPD5 Hub device only includes device select code of the ACK responses of the repeat start operation. In other words, if there are more than one repeat start operation, the SPD5 Hub device includes the device select of only the last repeat start from the host when it ACKs in PEC calculation and other NACK responses of the device select codes of the repeat start are not included in PEC calculation. If the SPD5 Hub device NACKs due to PEC error or a parity error in previous bytes from Host, it will always NACK regardless of how many times Host tries repeat start.
- The SPD5 Hub does not send any data shown in Table 27 and instead expects Host to perform stop operation.
- The SPD5 Hub device sets **MR52**[0] and **MR48**[7] and P_Err in GETSTATUS CCC to '1' for parity error and **MR52**[1] and **MR48**[7] and PEC_Err in GETSTATUS CCC to '1' for PEC error. Further, the SPD5 Hub updates Pending Interrupt Bits [3:0] in GETSTATUS CCC to '0001' and waits for the next opportunity to send an in band interrupt if IBI is enabled.

6.3.5.CCC Packet Error Handling

Parity error and PEC error detected in a CCC packet are handled in the same way as described for normal read/write operations.

6.3.6.Error Reporting

All error conditions including PEC error check and parity error check detected by the SPD5 Hub devices are captured in MR51 and **MR52** registers.

There are three different possible ways error information can be communicated to the host.

1. The host makes the read request to **MR51** and **MR52** registers.
2. The host starts any transaction with 7'h7E IBI header (only applicable in I3C mode).
3. The SPD5 Hub device sends in band interrupt if enabled, when its SCL and SDA input has been idle for t_{AVAL} time (only applicable in I3C Basic mode).

Copyright © 2024, NOVOSENSE

7. SPD5 Hub Device - Write and Read

In both I²C and I3C mode, any read or write access to SPD5 Hub NVM memory must be followed by stop operation (i.e. not repeat start) before launching new access to SPD5 Hub volatile memory registers. Conversely, any read or write access to SPD5 Hub volatile memory registers must be followed by stop operation (i.e., not repeat start) before launching new access to SPD5 Hub NVM memory.

7.1. Command Truth Table

The command truth table as shown in Table 84 only applies in I3C Basic mode with PEC enabled. In I²C mode and I3C Basic mode with PEC disabled, the command truth table does not apply.

Table 84 For I3C Basic Mode Only w/ PEC Enabled - Command Truth Table

SPD5 Command	Command Name	Command Code	RW	MemReg	Block Address	Address
		2nd Byte Bits [7:5]	2nd Byte Bit [4]	1st Byte Bit [7]	2nd Byte Bits [3:0] 1st Byte Bits [6]	1st Byte Bits [5:0]
Write 1 Byte to Register	W1R	000	0	0	V	V
Read 1 Byte from Register	R1R		1	0	V	V
Write 1 Byte to NVM	W1M		0	1	V	V
Read 1 Byte from NVM	R1M		1	1	V	V
Write 2 Byte to Register	W2R	001	0	0	V	V
Read 2 Byte from Register	R2R		1	0	V	V
Write 2 Byte to NVM	W2M		0	1	V	V
Read 2 Byte from NVM	R2M		1	1	V	V
Write 4 Byte to Register	W8R	010	0	0	V	V
Read 4 Byte from Register	R8R		1	0	V	V
Write 4 Byte to NVM	W8M		0	1	V	V
Read 4 Byte from NVM	R8M		1	1	V	V
Write 16 Byte to Register	W16R	011	0	0	V	V
Read 16 Byte from Register	R16R		1	0	V	V
Write 16 Byte to NVM	W16M		0	1	V	V
Read 16 Byte from NVM	R16M		1	1	V	V
Reserved	RSVD	011 to 111	RSVD	RSVD	RSVD	RSVD

Copyright © 2024, NOVOSENSE

7.2. Write and Read Access - NVM Memory

For I²C mode and I3 mode with PEC disabled, the Write access to NVM memory is done within 16-byte boundary in a block. Once the last address within 16-byte boundary in a block is reached, the device stops the operation. The write operation to remaining addresses is not executed by the SPD5 Hub device. The SPD5 Hub device does not loop to first address within the 16-byte boundary in that block. The SPD5 Hub device does not set any register to inform this to the host and does not generate any interrupt to the host.

For I3C mode with PEC enabled, if for a given Write access (W2M, W4M, W16M) to NVM memory reaches the last address within the 16-byte boundary in a block, (i.e., Byte 15, Byte 31, Byte 47, Byte 63), the device stops the operation. The write operation to remaining addresses is not executed by the SPD5 Hub device. The SPD5 Hub device does not loop to first address within the 16-byte boundary in that block. The SPD5 Hub device does not set any register to inform this to the host and does not generate any interrupt to the host.

Unlike Write access to NVM memory, any Read access to NVM memory does not impose 16-byte boundary or block boundary. The Read access to NVM memory is treated as continuous address space even if it crosses 16-byte boundary or block boundary in I²C mode or I3C Basic mode (with or without PEC enabled). The last byte for NVM memory is 1024th byte and when the address pointer reaches to the last byte, the SPD5 Hub device does not return any data. In I²C mode and I3C Basic mode with PEC disabled, the host must do the stop operation.

7.3. Write and Read Access - Register Memory

There is no concept of “Block Address”. The “Block Address” is treated is simply as upper address bit when MemReg = ‘0’ by the SPD5 Hub device.

For I²C mode and I3C Basic mode with PEC disabled or PEC enabled, any access to read or write to register memory is continuous address space even if it appears crossing 16-byte boundary or “Block Address” boundary. The last byte is MR255 and when the address pointer reaches to MR255, the SPD5 Hub device does not return any data.

In I²C mode and I3C Basic mode with PEC disabled, the host must do the stop operation.

7.4. Write Protection of Non-Volatile Memory

7.4.1. Normal Run Time Operation

In normal run time operation mode, by HSA Pin tied to GND via a resistor value, the SPD5 Hub device offers a write protection for its NVM memory. The **MR12**[7:0] and **MR13**[7:0] register contains a bit map for write protection status of each 64 byte block of NVM memory.

The **MR12**[7:0] and **MR13**[7:0] registers can be written to ‘1’ at any time. When any bit in **MR12** or **MR13** is set to ‘1’, further writes to that corresponding block of NVM are ignored and **MR52**[6] bit is set to ‘1’.

Once any bit is written to ‘1’ in **MR12** and **MR13**, clearing that bit in normal run time mode is not allowed. Any attempt to clear the bit in **MR12** and **MR13** is ignored and **MR52**[5] bit is set to ‘1’.

7.4.2. Offline Tester Operation

In offline tester operation mode, by HSA Pin tied directly to GND with no resistor value, the SPD5 Hub device allows to clear any bit in **MR12** and **MR13** registers.

Once cleared, the SPD5 Hub device allows to modify the corresponding block of NVM memory.

7.4.3. Suggested Steps to Program SPD5 Hub Devices

The recommended steps to program the SPD Hub devices are:

1. Connect HSA Pin directly to GND (without a resistor).
2. Power up the device. The device senses HSA pin. It sets **MR48**[2] = ‘1’ and enables write protection override.
3. Program **MR12** and **MR13** to enable desired NVM blocks to be written.
4. Program desired NVM blocks.
5. Program **MR12** and **MR13** to set the write protection as desired.

Copyright © 2024, NOVOSENSE

8. Registers

8.1. Register Attribute Definition

All volatile registers have base attributes as defined in Table 85. Some register attributes are further modified with attribute modifiers, as defined in Table 86.

The volatile register space has a continuous address. Unlike non-volatile memory in SPD5 Hub device, there is no concept of “Block” memory in volatile register space. When writing to and reading from volatile register space (i.e., MemReg = 0), the “Block Address bits” are treated simply as upper address bits.

Table 85 Register Base Attributes

Attribute	Abbreviation	Description
Read Only	R	This bit can be read by software. Writes have no effect.
Read/Write	RW	This bit can be read or written by software.
Write Only	W	This bit can only be written by software.
Reserved	RV	This bit is reserved for future expansion and its value must not be modified by software. The bit will return ‘0’ when read. When writing this bit, software must preserve the value read unless otherwise indicated.

Table 86 Register Attribute Modifier

Attribute	Abbreviation	Description
Write 1 Only	1O	This bit can only be set (i.e., write ‘1’) but not reset (i.e., write ‘0’)
Protected	P	This bit is protected by the password registers TBD. This bit cannot be written to unless the password code has been written into the password registers
Persistent	E	Persistent.

Copyright © 2024, NOVOSENSE

8.2. Register Map

Table 87 Register Map

Register Name	Register Address (Hex)	Attribute	Description
MR0	0x00	ROE	Device Type; Most Significant Byte
MR1	0x01	ROE	Device Type; Least Significant Byte
MR2	0x02	ROE	Device Revision
MR3	0x03	ROE	Vendor ID Byte 0
MR4	0x04	ROE	Vendor ID Byte 1
MR5	0x05	ROE	Device Capability
MR6	0x06	ROE	Device Write Recovery Time Capability
MR7 to MR10	0x07 to 0x0A	RV	Reserved
MR11	0x0B	RW	I^2C Legacy Mode Device Configuration
MR12	0x0C	RWE	Write Protection for NVM Blocks [7:0]
MR13	0x0D	RWE	Write Protection for NVM Blocks [15:8]
MR14	0x0E	RWE	Device Configuration - Host and Local Interface IO. DO NOT USE [4:0]
MR15	0x0F	RV	DO NOT USE
MR16 to MR17	0x10 to 0x11	RV	Reserved
MR18	0x12	RO, RW	Device Configuration
MR19	0x13	1O	Clear Register MR51 Temperature Status Command
MR20	0x14	1O	Clear Register MR52 Error Status Command
MR21 to MR25	0x15 to 0x19	RV	Reserved
MR26	0x1A	RW	TS Configuration
MR27	0x1B	1O, RO, RW	Interrupt Configurations
MR28	0x1C	RW	TS Temperature High Limit Configuration - Low Byte
MR29	0x1D	RW	TS Temperature High Limit Configuration - High Byte
MR30	0x1E	RW	TS Temperature Low Limit Configuration - Low Byte
MR31	0x1F	RW	TS Temperature Low Limit Configuration - High Byte
MR32	0x20	RW	TS Critical Temperature High Limit Configuration – Low Byte
MR33	0x21	RW	TS Critical Temperature High Limit Configuration – High Byte
MR34	0x22	RW	TS Critical Temperature Low Limit Configuration – Low Byte
MR35	0x23	RW	TS Critical Temperature Low Limit Configuration – High Byte
MR36 to MR47	0x24 to 0x2F	RV	Reserved for Device Configuration Type of Registers

Copyright © 2024, NOVOSENSE

MR48	0x30	RO	Device Status
MR49	0x31	RO	TS Current Sensed Temperature - Low Byte
MR50	0x32	RO	TS Current Sensed Temperature - High Byte
MR51	0x33	RO	TS Temperature Status
MR52	0x34	RO	Hub, Thermal and NVM Error Status
MR53 to MR127	0x35 to 0x7F	RV	Reserved
MR128 to MR255	0x80 to 0xFF	RV	Reserved for Vendor Specific Registers

8.3. Thermal Sensor Registers Read Out Mechanism

All thermal registers are 16-bit quantities stored in two consecutive registers: low byte first and then high byte. Five bits are reserved for future use. Reserved bits are read only bits and must be set to '0' when host writes to low and high byte. The device returns '0' in reserved bits when host reads from the low and high byte. Remaining eleven bits in these paired registers form a signed value of multiples of 0.25 ranging from -256.00 to + 255.75. Unit for all thermal registers is °C.

The format of each pair of thermal registers is shown in Table 88 below.

Table 88 Thermal Register - Low Byte and High Byte

Register		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
MRX	Low Byte	8	4	2	1	0.5	0.25	RSVD	RSVD
MRX+1	High Byte	RSVD	RSVD	RSVD	Sign	128	64	32	16

Bit 'Sign' decides whether the readout temperature is positive (Sign=0) or negative (Sign=1).

The examples (reserved bits in grey, sign bit highlighted in blue) is shown in Table 89 below.

Table 89 Thermal Register Examples

High Byte	Low Byte	Value	Unit
0000 0101	1111 0000	+ 95.00	°C
0000 0101	0101 0000	+ 85.00	°C
0000 0100	1011 0000	+ 75.00	°C
0000 0000	0001 0000	+ 1.00	°C
0000 0000	0000 1100	+ 0.75	°C
0000 0000	0000 1000	+ 0.50	°C
0000 0000	0000 0100	+ 0.25	°C
0000 0000	0000 0000	0.00	°C
0001 1111	1111 1100	- 0.25	°C
0001 1111	1111 1000	- 0.50	°C
0001 1111	1111 0100	- 0.75	°C
0001 1111	1111 0000	- 1.00	°C
0001 1101	1000 0000	- 40.00	°C

Copyright © 2024, NOVOSENSE

8.4. Register Description

Table 90 MR0

Bits	Attr	Default	Device Type; Most Significant Byte[1]
7:0	ROE	0x51	**MR0** [7:0]: MSB_DEV_TYPE Device Type - SPD5 Hub Device

1. The code in this register is used in conjunction with any device type in **MR1** register.

Table 91 MR1

Bits	Attr	Default	Device Type; Most Significant Byte[1]
7:0	ROE	0x18	**MR1** [7:0]: LSB_DEV_TYPE Device Type 0x18: with Temperature Sensor; SPD5 Hub Device

1. The code in this register is used in conjunction with any device type in **MR0** register.

Table 92 MR2

Bits	Attr	Default	Device Revision
7:6	RV	00	**MR2** [7:6]: Reserved
5:4	ROE	00	**MR2** [5:4]: DEV_REV_MAJOR Major Revision 00 = Revision 1 01 = Revision 2 10 = Revision 3 11 = Revision 4
3:1	ROE	101	**MR2** [3:1]: DEV_REV_MINOR Minor Revision 000 = Revision 0 001 = Revision 1 … 111 = Revision 7
0	RV	0	**MR2** [0]: Reserved

Copyright © 2024, NOVOSENSE

Table 93 MR3

Bits	Attr	Default	VENDOR ID
7:0	ROE	Ox0E	**MR3** [7:0]: VENDOR_ID_BYTE0 Vendor ID Byte 0

Table 94 MR4

Bits	Attr	Default	VENDOR ID
7:0	ROE	0xB5	**MR4** [7:0]: VENDOR_ID_BYTE1 Vendor ID Byte 1

Table 95 MR5

Bits	Attr	Default	Device Capability
7:2	RV	0	**MR5** [7:2]: Reserved
1	ROE	1	**MR5** [1]: TS_SUPPORT Internal Temperature Sensor Support 0 = Does not support Temperature Sensor 1 = Supports Temperature Sensor
0	ROE	1	**MR5** [0]: HUB_SUPPORT Hub Function Support 0 = Does not support Hub function 1 = Supports Hub function

Table 96 MR6

Bits	Attr	Default	Write Recovery Time
7:4	ROE	-	**MR6** [7:4]: WR_REC_UNIT Write Recovery Unit 0000 = 0 0001 = 1 0010 = 2 0011 = 3 0100 = 4 0101 = 5 0110 = 6 0111 = 7 1000 = 8 1001 = 9 1010 = 10 1011 = 50

Copyright © 2024, NOVOSENSE

			1100 = 100 1101 = 200 1110 = 500 1111 = Reserved
3:2	RV	0	**MR6** [3:2]: Reserved
1:0	ROE	-	**MR6** [1:0]: WR_REC_UNIT_TIME Write Recovery Time Unit 00 = ns 01 = µs 10 = ms 11 = Reserved

Table 97 MR11

Bits	*Attr*	*Default*	*I²C Legacy Mode Device Configuration*
7:4	RV	0	**MR11** [7:4]: Reserved
3	RW	0	**MR11** [3]: I²C_LEGACY_MODE_ADDR SPD5 Hub Device - I²C Legacy Mode Addressing 0 = 1 Byte Addressing for SPD5 Hub Device Memory 1 = 2 Bytes Addressing for SPD5 Hub Device Memory
2:0	RW	000	**MR11** [2:0]: I²C_LEGACY_MODE_ADDR_POINTER SPD5 Device - Non-Volatile Memory Address Page Pointer in I²C Legacy Mode[1,2,3] 000 = Page 0 (0x00 to 0x7F) 001 = Page 1 (0x80 to 0xFF) 010 = Page 2 (0x100 to 0x17F) 011 = Page 3 (0x180 to 0x1FF) 100 = Page 4 (0x200 to 0x27F) 101 = Page 5 (0x280 to 0x2FF) 110 = Page 6 (0x300 to 0x37F) 111 = Page 7 (0x380 to 0x3FF)

1. This register is only applicable if bit [3] = '0' and **MR18** [5] = '0'. The SPD5 Hub device does not incur any delay to switch from one page to another page.
2. This register only applies to non-volatile memory (1024 Bytes) access of SPD5 Hub device. For volatile memory access, this register must be programmed to '000'.
3. See **Section 7.2** for the NVM Write and Read operation when device reaches the last byte of the 16-byte block boundary.

Copyright © 2024, NOVOSENSE

Table 98 MR12

Bits	Attr	Default	NVM Protection Configuration For Blocks [7:0][1,2]
7	RWE	-	**MR12** [7]: WP_BLK_7 Write Protect - Block 7 0 = Not Protected 1 = Protected
6	RWE	-	**MR12** [6]: WP_BLK_6 Write Protect - Block 6 0 = Not Protected 1 = Protected
5	RWE	-	**MR12** [5]: WP_BLK_5 Write Protect - Block 5 0 = Not Protected 1 = Protected
4	RWE	-	**MR12** [4]: WP_BLK_4 Write Protect - Block 4 0 = Not Protected 1 = Protected
3	RWE	-	**MR12** [3]: WP_BLK_3 Write Protect - Block 3 0 = Not Protected 1 = Protected
2	RWE	-	**MR12** [2]: WP_BLK_2 Write Protect - Block 2 0 = Not Protected 1 = Protected
1	RWE	-	**MR12** [1]: WP_BLK_1 Write Protect - Block 1 0 = Not Protected 1 = Protected
0	RWE	-	**MR12** [0]: WP_BLK_0 Write Protect - Block 0 0 = Not Protected 1 = Protected

1. Once any register bit is set to '1', it can only be cleared when the SPD5 Hub device is in offline tester mode of operation.
2. The write (or update) transaction to this register must be followed by stop operation to allow SPD5 Hub device to update the setting.

Copyright © 2024, NOVOSENSE

Table 99 MR13

Bits	Attr	Default	NVM Protection Configuration For Blocks [15:8][1,2]
7	RWE	-	**MR13** [7]: WP_BLK_15 Write Protect - Block 15 0 = Not Protected 1 = Protected
6	RWE	-	**MR13** [6]: WP_BLK_14 Write Protect - Block 14 0 = Not Protected 1 = Protected
5	RWE	-	**MR13** [5]: WP_BLK_13 Write Protect - Block 13 0 = Not Protected 1 = Protected
4	RWE	-	**MR13** [4]: WP_BLK_12 Write Protect - Block 12 0 = Not Protected 1 = Protected
3	RWE	-	**MR13** [3]: WP_BLK_11 Write Protect - Block 11 0 = Not Protected 1 = Protected
2	RWE	-	**MR13** [2]: WP_BLK_10 Write Protect - Block 10 0 = Not Protected 1 = Protected
1	RWE	-	**MR13** [1]: WP_BLK_9 Write Protect - Block 9 0 = Not Protected 1 = Protected
0	RWE	-	**MR13** [0]: WP_BLK_8 Write Protect - Block 8 0 = Not Protected 1 = Protected

1. Once any register bit is set to '1', it can only be cleared when the SPD5 Hub device is in offline tester mode of operation.
2. The write (or update) transaction to this register must be followed by stop operation to allow SPD5 Hub device to update the setting.

Copyright © 2024, NOVOSENSE

Table 100 MR14

Bits	Attr	Default	Device Configuration - Local Interface[1,2]
7:6	RV	0	**MR14** [7:6]: Reserved
5	RWE	0	**MR14** [5]: LOCAL_INF_PULLUP_CONF Local Interface - Pull Up Resistor Configuration 0 = Internal (on die) Pullup Resistor[3] 1 = External (board) Pullup Resistor
4:0	RV	0	**MR14** [4:0]: DO NOT USE

1. DIMM Vendor configures this register during assembly based on the DIMM design. After SPD Hub device is powered up, the host can alter the setting through this register.
2. The write (or update) transaction to this register must be followed by stop operation to allow SPD5 Hub device to update the setting.
3. Typical value of on die pullup resistor is 1 kΩ. The minimum and maximum on die pullup resistor value is 750 Ω and 1.5 kΩ respectively.

Table 101 MR15

Bits	Attr	Default	DO NOT USE
7:0	RV	0	**MR15** [7:0]: DO NOT USE

Copyright © 2024, NOVOSENSE

Table 102 MR18

Bits	Attr	Default	Device Configuration[1]
7	RW	0	**MR18** [7]: PEC_EN PEC Enable[2,3] 0 = Disable 1 = Enable
6	RW	0	**MR18** [6]: PAR_DIS Parity (T bit) Disable[3,4] 0 = Enable 1 = Disable
5	RO	0	**MR18** [5]: INF_SEL Interface Selection 0 = I^2C Protocol (Max speed of 1 MHz) 1 = I3C Basic Protocol[5]
4	RW	0	**MR18** [4]: DEF_RD_ADDR_POINT_EN Default Read Address Pointer Enable 0 = Disable Default Read Address Pointer (Address pointer is set by the host)[6] 1 = Enable Default Read Address Pointer; Address selected by register bits [3:2]
3:2	RW	00	**MR18** [3:2]: DEF_RD_ADDR_POINT_START Default Read Pointer Starting Address[7] 00 = **MR49** 01 = Reserved 10 = Reserved 11 = Reserved
1	RW	0	**MR18** [1]: DEF_RD_ADDR_POINT_BL Burst Length for Read Pointer Address for PEC Calculation[8] 0 = 2 Bytes 1 = 4 Bytes
0	RV	0	**MR18** [0]: Reserved

1. The write (or update) transaction to this register must be followed by stop operation to allow the SPD5 Hub device to update the setting.
2. This register is only applicable if **MR18** [5] = '1'.
3. This register is updated when RSTDAA CCC is registered by SPD5 Hub device or when SPD5 Hub device goes through bus reset as described in **Section 2.4**.
4. This register is only applicable if **MR18** [5] = '1'. When Parity function is disabled, the SPD5 Hub device simply ignores the 'T' bit information from the host. The host may actually choose to compute the parity and send that information in 'T' bit or simply drive static low or high in 'T' bit.
5. This register is automatically updated when SETAASA CCC or RSTDAA CCC is registered by the SPD5 Hub device or when SPD5 Hub device goes through bus reset as described in **Section 2.4**. This register can be read by the host through normal read

Copyright © 2024, NOVOSENSE

operation, but it cannot be written with normal write operation either in I²C mode or I3C Basic mode of operation. When this register is updated, it takes in effect when there is a next START operation (i.e., after stop operation).

6. The setting in register **MR18** [3:1] is don't care.
7. This register is only applicable if **MR18** [4] = '1'.
8. This register is only applicable if **MR18** [7, 4] = '11'.

Table 103 MR19

Bits	Attr	Default	Clear Register Command[1]
7:4	RV	0	**MR19** [7:4]: Reserved
3	1O	0	**MR19** [3]: CLR_TS_CRIT_LOW Clear Temperature Sensor Critical Low Status 1 = Clear **MR51** [3] Register
2	1O	0	**MR19** [2]: CLR_TS_CRIT_HIGH Clear Temperature Sensor Critical High Status 1 = Clear **MR51** [2] Register
1	1O	0	**MR19** [1]: CLR_TS_LOW Clear Temperature Sensor Low Status 1 = Clear **MR51** [1] Register
0	1O	0	**MR19** [0]: CLR_TS_HIGH Clear Temperature Sensor High Status 1 = Clear **MR51** [0] Register

1. This entire register is self-clearing register after corresponding register is cleared.

Copyright © 2024, NOVOSENSE

Table 104 MR20

Bits	Attr	Default	Clear Register Command[1]
7	1O	0	**MR20** [7]: CLR_SPD_BUSY_ERROR Clear Write or Read Attempt while SPD Device Busy Error Status 1 = Clear **MR52** [7] Register
6	1O	0	**MR20** [6]: CLR_WR_NVM_BLK_ERROR Clear Write Attempt to Protected NVM Block Error Status 1 = Clear **MR52** [6] Register
5	1O	0	**MR20** [5]: CLR_WR_NVM_PRO_REG_ERROR Clear Write Attempt to NVM Protection Register Error Status 1 = Clear **MR52** [5] Register
4:2	RV	0	**MR20** [4:2]: Reserved
1	1O	0	**MR20** [1]: CLR_PEC_ERROR Clear Packet Error Status 1 = Clear **MR52**[1] Register
0	1O	0	**MR20** [0]: CLR_PAR_ERROR Clear Parity Error Status 1 = Clear **MR52** [0] Register

1. This entire register is self-clearing register after corresponding register is cleared.

Table 105 MR26

Bits	Attr	Default	Thermal Sensor Configuration
7:1	RV	0	**MR26** [7:1] Reserved
0	RW	0	**MR26** [0]: DIS_TS Disable Temperature Sensor[1,2] 0 = Enable thermal sensor 1 = Disable thermal sensor

1. This register is only applicable if **MR1** [7:0] programmed value is 0x18.
2. If this bit is set to '1' and then reset to '0', the host must wait minimum of t_{INIT} before accessing samples on the thermal sensor.

Copyright © 2024, NOVOSENSE

Table 106 MR27

Bits	Attr	Default	Interrupt Configuration
7	1O	0	**MR27** [7]: CLR_GLOBAL Global Clear Event Status and In Band Interrupt Status1,2 1 = Clear **MR48** [7], **MR51** [3:0] and **MR52** [7:5, 1:0] Register
6:5	RV	0	**MR27** [6:5]: Reserved
4	RO	0	**MR27** [4]: IBI_ERROR_EN In Band Error Interrupt Enable for **MR52** Error Log3 0 = Disable; Errors logged in **MR52** [7:5, 1:0] registers do not generate an IBI to Host 1 = Enable; Errors logged in **MR52** [7:5, 1:0] registers generate an IBI to Host
3	RW	0	**MR27** [3]: IBI_TS_CRIT_LOW_EN In Band Error Interrupt Enable for Temperature Sensor Critical Low 0 = Disable; **MR51** [3] = '1' does not generate an IBI to Host 1 = Enable; **MR51** [3] = '1' and **MR27** [4] = '1' generates an IBI to Host
2	RW	0	**MR27** [2]: IBI_TS_CRIT_HIGH_EN In Band Error Interrupt Enable for Temperature Sensor Critical High 0 = Disable; **MR51** [2] = '1' does not generate an IBI to Host 1 = Enable; **MR51** [2] = '1' and **MR27** [4] = '1' generates an IBI to Host
1	RW	0	**MR27** [1]: IBI_TS_LOW_EN In Band Error Interrupt Enable for Temperature Sensor Low 0 = Disable; **MR51** [1] = '1' does not generate an IBI to Host 1 = Enable; **MR51** [1] = '1' and **MR27** [4] = '1' generates an IBI to Host
0	RV	0	**MR27** [0]: IBI_TS_HIGH_EN In Band Error Interrupt Enable for Temperature Sensor High 0 = Disable; **MR51** [0] = '1' does not generate an IBI to Host 1 = Enable; **MR51** [0] = '1' and **MR27** [4] = '1' generates an IBI to Host

1. This register is a self-clearing register after corresponding registers are cleared. Writing '0' in this register has no effect.
2. After this command is issued, the device does not generate an IBI for any pending event. But if new event occurs, the device does generate an IBI.
3. This register is automatically updated when ENEC CCC or DISEC CCC or RSTDAA CCC is registered by the SPD5 Hub device or when SPD5 Hub device goes through bus reset as described in **Section 2.4**. This register can be read by the host through normal read operation but cannot be written with normal write operation either in I^2C mode or I3C Basic mode. When this register is updated, it takes effect when there is a next start operation (i.e., after stop operation).

Copyright © 2024, NOVOSENSE

Table 107 MR28

Bits	Attr	Default	Thermal Sensor High Limit Configuration - Low Byte[1,2,3]
7:0	RW	0x70	**MR28** [7:0]: TS_HIGH_LIMIT_LOW **MR28** and **MR29** - 16-bit thermal registers define the high limit for thermal sensor. See Table 88 Thermal Register - Low Byte and High Byte.

1. This entire register is only applicable if **MR1** [7:0] programmed value is 0x18.
2. Critical temperature high limit value must have a higher value than temperature high limit (**MR28** [7:0] and **MR29** [7:0]).
3. The reserved bits are read only bits. The host must write '0' in reserved bits when writing and device always returns '0' from reserved bits when reads this byte.

Table 108 MR29

Bits	Attr	Default	Thermal Sensor High Limit Configuration - High Byte[1,2,3]
7:0	RW	0x03	**MR29** [7:0]: TS_HIGH_LIMIT_HIGH **MR28** and **MR29** - 16-bit thermal registers define the high limit for thermal sensor. See Table 88 Thermal Register - Low Byte and High Byte.

1. This entire register is only applicable if **MR1** [7:0] programmed value is 0x18.
2. Critical temperature high limit value must have a higher value than temperature high limit (**MR28**[7:0] and **MR29**[7:0]).
3. The reserved bits are read only bits. The host must write '0' in reserved bits when writing and device always returns '0' from reserved bits when reads this byte.

Table 109 MR30

Bits	Attr	Default	Thermal Sensor Low Limit Configuration - Low Byte[1,2,3]
7:0	RW	0	**MR30** [7:0]: TS_LOW_LIMIT_LOW **MR30** and **MR31** - 16-bit thermal registers define the low limit for thermal sensor. See Table 88 Thermal Register - Low Byte and High Byte.

1. This entire register is only applicable if **MR1** [7:0] programmed value is 0x18.
2. Critical temperature low limit value must have a lower value than temperature low limit (**MR30** [7:0] and **MR31** [7:0]).
3. The reserved bits are read only bits. The host must write '0' in reserved bits when writing and device always returns '0' from reserved bits when reads this byte.

Copyright © 2024, NOVOSENSE

Table 110 MR31

Bits	Attr	Default	Thermal Sensor Low Limit Configuration - High Byte[1,2,3]
7:0	RW	0	**MR31** [7:0]: TS_LOW_LIMIT_HIGH **MR30** and **MR31** - 16-bit thermal registers define the low limit for thermal sensor. See Table 88 Thermal Register - Low Byte and High Byte.

1. This entire register is only applicable if **MR1** [7:0] programmed value is 0x18.
2. Critical temperature low limit value must have a lower value than temperature low limit (**MR30** [7:0] and **MR31** [7:0]).
3. The reserved bits are read only bits. The host must write '0' in reserved bits when writing and device always returns '0' from reserved bits when reads this byte.

Table 111 MR32

Bits	Attr	Default	Thermal Sensor Critical Temperature High Limit Configuration - Low Byte[1,2,3]
7:0	RW	0x50	**MR32**[7:0]: TS_CRIT_HIGH_LIMIT_LOW **MR32** and **MR33** - 16-bit thermal registers define the critical temperature high limit for thermal sensor. See Table 88 Thermal Register - Low Byte and High Byte.

1. This entire register is only applicable if **MR1** [7:0] programmed value is 0x18.
2. Critical temperature high limit value must have a higher value than temperature high limit (**MR28** [7:0] and **MR29** [7:0]).
3. The reserved bits are read only bits. The host must write '0' in reserved bits when writing and device always returns '0' from reserved bits when reads this byte.

Table 112 MR33

Bits	Attr	Default	Thermal Sensor Critical Temperature High Limit Configuration - High Byte[1,2,3]
7:0	RW	0x05	**MR33** [7:0]: TS_CRIT_HIGH_LIMIT_HIGH **MR32** and **MR33** - 16-bit thermal registers define the critical temperature high limit for thermal sensor. See Table 88 Thermal Register - Low Byte and High Byte.

1. This entire register is only applicable if **MR1** [7:0] programmed value is 0x18.
2. Critical temperature high limit value must have a higher value than temperature high limit (**MR28** [7:0] and **MR29** [7:0]).
3. The reserved bits are read only bits. The host must write '0' in reserved bits when writing and device always returns '0' from reserved bits when reads this byte.

Table 113 MR34

Bits	Attr	Default	Thermal Sensor Critical Temperature Low Limit Configuration - Low Byte[1,2,3]
7:0	RW	0	**MR34**[7:0]: TS_CRIT_LOW_LIMIT_LOW **MR34** and **MR35** - 16-bit thermal registers define the critical temperature low limit for thermal sensor. See Table 88 Thermal Register - Low Byte and High Byte.

1. This entire register is only applicable if **MR1** [7:0] programmed value is 0x18.
2. Critical temperature low limit value must have a lower value than temperature low limit (**MR30** [7:0] and **MR31** [7:0]).

Copyright © 2024, NOVOSENSE

3. The reserved bits are read only bits. The host must write '0' in reserved bits when writing and device always returns '0' from reserved bits when reads this byte.

Table 114 MR35

Bits	Attr	Default	Thermal Sensor Critical Temperature Low Limit Configuration - High Byte[1,2,3]
7:0	RW	0	**MR35** [7:0]: TS_CRIT_LOW_LIMIT_HIGH **MR34** and **MR35** - 16-bit thermal registers define the critical temperature low limit for thermal sensor. See Table 88 Thermal Register - Low Byte and High Byte.

1. This entire register is only applicable if **MR1** [7:0] programmed value is 0x18.
2. Critical temperature low limit value must have a lower value than temperature low limit (**MR30** [7:0] and **MR31** [7:0]).
3. The reserved bits are read only bits. The host must write '0' in reserved bits when writing and device always returns '0' from reserved bits when reads this byte.

Table 115 MR48

Bits	Attr	Default	Device Status
7	RO	0	**MR48** [7]: IBI_STATUS Device Event In Band Interrupt Status 0 = No pending IBI 1 = Pending IBI
6:4	RV	0	**MR48** [6:4]: Reserved
3	RO	0	**MR48** [3]: WR_OP_STATUS Write Operation Status 0 = No internal write operation is on-going. 1 = Internal write operation is on-going; Device ignores Host Write command if Host attempts to write when this bit is '1'. The device self clears this bit to '0' when it completes internal write operation
2	RO	-	**MR48** [2]: WP_OVERRIDE_STATUS Write Protect Override Status 0 = Override of write protect bits in **MR12** and **MR13** are blocked 1 = Override of write protect bits in **MR12** and **MR13** are allowed The default state of this register reflects the sensing of HSA pin during power on. This bit is set to '1' if HSA pin is directly tied to GND. This bit is set to '0' if HSA pin is connected to GND through a resistor.
1:0	RV	0	**MR48** [1:0]: Reserved

Copyright © 2024, NOVOSENSE

Table 116 MR49

Bits	Attr	Default	Current Sensed Temperature - Low Byte[1,2]
7:0	RO	0	**MR49** [7:0]: TS_SENSE_LOW **MR49** and **MR50** - 16-bit thermal registers return the most recent conversion of the thermal sensor. See Table 88 Thermal Register - Low Byte and High Byte.

1. This entire register is only applicable if **MR1** [7:0] programmed value is 0x18.
2. The device always returns '0' from reserved bits.

Table 117 MR50

Bits	Attr	Default	Current Sensed Temperature - High Byte[1,2]
7:0	RO	0	**MR50** [7:0]: TS_SENSE_HIGH MR49 and **MR50** - 16-bit thermal registers return the most recent conversion of the thermal sensor. See Table 88 Thermal Register - Low Byte and High Byte.

1. This entire register is only applicable if **MR1** [7:0] programmed value is 0x18.
2. The device always returns '0' from reserved bits.

Table 118 MR51

Bits	Attr	Default	Thermal Sensor Temperature Status[1]
7:4	RV	0	**MR51** [7:4]: Reserved
3	RO	0	**MR51 [3]: TS_CRIT_LOW_STATUS** Temperature Sensor Critical Low 0 = Temperature is above the limit set in **MR34** and **MR35** 1 = Temperature is below the limit set in **MR34** and **MR35**
2	RO	0	**MR51** [2]: TS_CRIT_HIGH_STATUS Temperature Sensor Critical High 0 = Temperature is below the limit set in **MR32** and **MR33** 1 = Temperature is above the limit set in **MR32** and **MR33**
1	RO	0	**MR51** [1]: TS_LOW_STATUS Temperature Sensor Low 0 = Temperature above limit set in registers **MR30** and **MR31** 1 = Temperature below limit set in registers **MR30** and **MR31**
0	RO	0	**MR51** [0]: TS_HIGH_STATUS Temperature Sensor High 0 = Temperature is below the limit set in registers **MR28** and **MR29** 1 = Temperature is above the limit set in registers **MR28** and **MR29**

1. This entire register is only applicable if **MR1** [7:0] programmed value is 0x18.

Copyright © 2024, NOVOSENSE

Table 119 MR52

Bits	Attr	Default	Hub and Thermal Sensor Error Status
7	RO	0	**MR52** [7]: BUSY_ERROR_STATUS Write or Read Attempt while SPD5 Hub Device was Busy (Write Recovery Time Violation)[1,2] 0 = No write or read attempt while SPD Hub device was busy 1 = Write or Read attempt while SPD5 Hub device was busy
6	RO	0	**MR52** [6]: WR_NVM_BLK_ERROR_STATUS Write Attempt to Protected NVM Block 0 = No write attempt 1 = Write attempt to protected NVM Block
5	RO	0	**MR52** [5]: WR_NVM_PRO_REG_ERROR_STATUS Write Attempt to NVM Protection Registers 0 = No attempt to modify write protect registers 1 = Write attempt to modify write protect registers
4:2	RO	0	**MR52** [4:2]: Reserved
1	RO	0	**MR52** [1]: PEC_ERROR_STATUS Packet Error[3,4] 0 = No PEC Error 1 = PEC Error in one or more packets
0	RO	0	**MR52** [0]: PAR_ERROR_STATUS Parity Check Error[4,5] 0 = No Parity Error 1 = Parity Error in one or more bytes

1. SPD5 Hub device busy status is only for accessing EEPROM memory. For any access to volatile register space, this bit definition does not apply.
2. When SPD5 Hub device is busy with EEPROM write/read, it sends NACK to the host requests within write recovery time.
3. Only applicable **MR18** [5] = '1' and if PEC function is enabled.
4. This register is updated when SPD5 Hub device goes through bus reset as described in **Section 2.4**.
5. Only applicable in **MR18** [5] = '1' and if parity function is not disabled or for supported CCC in I^2C mode.

Copyright © 2024, NOVOSENSE

9. Electrical Specifications

9.1. Absolute Maximum Ratings

Stressing the device above the rating listed in the absolute maximum ratings table may cause permanent damage to the device. These are stress ratings only and operation of the device at these or any other conditions above those indicated in the Operating sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 120 Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit
T_{STG}	Storage temperature	-65	+150	°C
V_{DDIO}	Supply voltage	-0.5	+2.1	V
V_{DDSPD}	Supply voltage	-0.5	+2.1	V
V_{HSA}	Voltage on HSA Pin	-0.5	+2.1	V
V_{IO}	Voltage on HSCL, HSDA, LSCL, LSDA Pins	-0.5	+3.6	V

9.2. ESD Ratings

Table 121 ESD Ratings

Symbol	*Ratings*	*Value*	*Unit*
$V_{ESD-HBM}$	Human body model (JESD22/A114) - 100pF, 1.5kΩ ● All pins	±6.0	kV
$V_{ESD-CDM}$	Charged device model (JESD22/C101): ● All pins	±2	kV

9.3. Operating Condition, Measurement Condition

Table 122 Operating Condition

Symbol	Parameter	Min	Typ	Max	Unit
V_{DDSPD}	Input Supply Voltage[1]	1.7	1.8	1.98	V
V_{DDIO}	Input Supply Voltage[2]	0.95	1.0	1.05	V
T_{CASE}	Case operating temperature	-40		125	°C
$T_{WRITEOK}$	Case temperature range for NVM Write operation Data writes outside this range may not meet retention requirements	-40		95	°C

1. For DDR5 DIMM application, the DDR5 PMIC VOUT_1.8V setting should be selected such that absolute min and max values for SPD Hub specification are not violated.
2. For DDR5 DIMM application, the DDR5 PMIC VOUT_1.0V setting should be selected such that absolute min and max values for SPD Hub specification are not violated.

Copyright © 2024, NOVOSENSE

Table 123 Write Endurance and Data Retention Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
D_R	Data Retention @ 25 °C	TBD			Years
N_{W1}	Write Endurance Cycles @ 25 °C	≥100000			Cycles
N_{W2}	Write Endurance Cycles @ 95 °C	100000			Cycles

Table 124 AC Measurement Conditions[1]

Symbol	Parameter	Min.	Max.	Unit
C_L	Load capacitance	40		pF
	Input rise and fall times - open-drain	-	TBD	ns
	Input rise and fall times - push-pull	-	TBD	ns
	Input signal swing levels	0.2 to 0.8		V
	Input levels for timing reference	0.3 to 0.7		V

1. This AC measurement condition (Table 124 and Figure 19) is only for the test purpose in lab.

Figure 19 AC Measurement Waveform

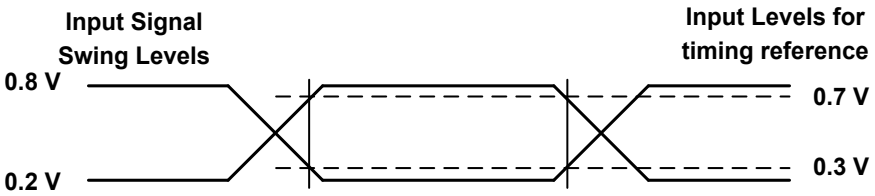


Table 125 Input Parameters

Symbol	Parameter[1,2]	Test Condition	Min	Max	Unit
C_{IN}	Input capacitance (HSDA, HSCL, LSDA)	-	-	5	pF
t_{SP}	Pulse width of spikes which must be suppressed by the input filter in I^2C mode	Single glitch, f ≤ 100 kHz	-	-	ns
		Single glitch, f > 100 kHz	-	50	ns

1. T_A = 25 °C, f = 400 kHz
2. Verified by design and characterization, not necessarily tested on all devices.

Table 126 Output Ron Specification

Symbol	Parameter[1]	Min	Max	Unit
R_{on_PUH}	HSDA Output Pullup Driver Impedance	10	45	Ω
R_{on_PDH}	HSDA Output Pulldown Driver Impedance	10	40	Ω
R_{on_Local}	LSCL, LSDA Output Pullup and Pulldown Driver Impedance	20	100	Ω

1. Pulldown $R_{on} = V_{out} / I_{out}$; Pullup $R_{on} = (V_{IO} - V_{out}) / I_{out}$

Copyright © 2024, NOVOSENSE

9.4. DC Characteristics

Table 127 DC Characteristics

Symbol	Parameter	Test Condition	Min	Max	Unit
I_{LI}	Input Leakage Current		-	± 5	μA
I_{LO}	Output Leakage Current		-	± 5	μA
I_{DDR}	Supply Current, Read Operation[1]	V_{DDSPD} = 1.8V, f_C = 12.5 MHz	-	2	mA
I_{DDW}	Supply Current, Write Operation[1]	V_{DDSPD} = 1.8V, f_C = 12.5 MHz	-	3	mA
I_{DD1}	Standby Supply Current	V_{IN} = V_{DDSPD} = 1.8V	-	TBD	μA
V_{IL}	Input Low Voltage		-0.35	0.3	V
V_{IH}	Input High Voltage		0.7	3.6	V
V_{OL}	Output Low Voltage[2]	3 mA sink current	-	0.3	V
V_{OH}	Output High Voltage	3 mA source current	0.75	-	V
I_{OL}	Output Low Current (HSDA, LSDA, LSCL)	V_{OL} = 0.3V	3		mA
I_{OH}	Output High Current (HSDA, LSDA, LSCL)	V_{OH} = V_{DDIO} - 0.3V		-3	mA
SR_R	Rising Output Slew Rate[3] (HSDA, LSDA, LSCL)		0.1	1.0	V/ns
SR_F	Falling Output Slew Rate[3] (HSDA, LSDA, LSCL)		0.1	3.0	V/ns
V_{PON}	Power On Reset Threshold	Monotonic rise between V_{PON} and $V_{DDSPD(min)}$ without ringback	1.6	-	V
V_{POFF}	Power Off Threshold for Warm Power On Cycle	No ringback above VP_{OFF}	-	0.3	V

1. Thermal sensor is active.
2. Example calculation for V_{OL}= R_{on} * ($I_{leakage}$ + I_{out}); where I_{out} = V_{DDIO}/ (R_{pu} + R_{on}); R_{pu} = Pullup resistor with typical value of 1 kΩ; Min and Max value of 750 Ω and 1.5 kΩ, respectively.
3. Output slew rate is guaranteed by design and / or characterization. The output slew rate reference load is shown in Figure 23 and Figure 24 shows the timing measurement points. For slew rate measurements, the V_{OH} level shown in Figure 24 is a function of R_{on} value; V_{OH} = {1.0 / (R_{on} + 50)} * 50.
4. Current into device pins is defined as positive. Current out of device pins is defined as negative.

Copyright © 2024, NOVOSENSE

9.5. AC Characteristics

Table 128 AC Characteristics

Symbol	Parameter	I²C Mode - open-drain		I3C Basic - Push-Pull[1]		Unit
		Min	Max	Min	Max	
f_{SCL}	Clock Frequency	0.01	1	0	12.5	MHz
t_{HIGH}	Clock Pulse Width High Time	260	-	35	-	ns
t_{LOW}	Clock Pulse Width Low Time	500	-	35	-	ns
$t_{TIMEOUT}$	Detect Clock Low Timeout	10	50	10	50	ms
t_R	SDA Rise Time[2,3]	-	120	-	5	ns
t_F	SDA Fall Time[2,3]	-	120	-	5	ns
$t_{SU:DAT}$	Data In Set-up Time[2]	50	-	8	-	ns
$t_{HD:DI}$	Data In Hold Time[2]	0	-	3	-	ns
$t_{SU:STA}$	Start Condition Setup Time[2]	260	-	12	-	ns
$t_{HD:STA}$	Start Condition Hold Time[2]	260	-	30	-	ns
$t_{SU:STO}$	Stop Condition Setup Time[2]	260	-	12	-	ns
t_{BUF}	Time between Stop Condition and next Start Condition[2,4]	500	-	500	-	ns
t_W	Write Time	-	5	-	5	ms
t_{POFF}	Warm Power Cycle Off Time	1	-	1	-	ms
t_{Sense_HSA}	Time from Valid 1.8V Supply to Sense HSA Pin for HID Code Assignment	-	5	-	5	ms
t_{INIT}	Time from Power On to First Command	10	-	10	-	ms
t_{AVAL}	Bus Available Time (No Edges Seen on HSDA and HSCL)	-	-	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 to Avoid IBI Generation; PEC Disabled	-	-	4	-	µs
	Time from Clear Register Status to any I3C Operation with Start Condition to Avoid IBI Generation; PEC Enabled	-	-	15	-	µs
t_{PDHL}	Propagation Delay, HSDA to LSDA and HSCL to LSCL[5]	-	N/A	-	6	ns
t_{PDLH}	Propagation Delay, LSDA to HSDA[5]	-	N/A	-	6	ns
$t_{HD:DAT}$	HSCL Falling Clock In to HSDA Data Out Hold Time[6]	0.5	350	N/A	N/A	ns
t_{DOUT}	HSCL Falling Clock In to HSDA Valid Data Out Time[7]	N/A	N/A	0.5	12	ns

Copyright © 2024, NOVOSENSE

Symbol	Parameter	I²C Mode - open-drain		I3C Basic - Push-Pull[1]		Unit
		Min	Max	Min	Max	
t_{DOFFT}	HSCL Rising Clock In to SDA Output Off[8]	N/A	N/A	0.5	12	ns
t_{DOFFC}	HSCL Rising Clock In to Controller SDA Output Off[9]	N/A	N/A	0.5	t_{HIGH}	ns
$t_{CL\_r\_DAT\_f}$	HSCL Rising Clock In to Controller Driving HSDA Low[10]	N/A	N/A	40	-	ns
$t_{DEVCTRLCCC\_DELAY\_PEC\_DIS}$	DEVCTRL CCC Followed by DEVCTRL CCC or Register Read/Write Command Delay[11,12,13]	3	-	3	-	µs
$t_{WR\_RD\_DELAY\_PEC\_EN}$	Register Write Command Followed by Register Read Command Delay in PEC Enabled Mode[14,15,16]	N/A	N/A	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	N/A	N/A	2.5	-	µs

1. I3C mode with open-drain operation follows timing values as shown in I²C Mode - open-drain column.
2. See Figure 20 for input timing parameter definition.
3. See Figure 25 for voltage threshold definition for rise and fall times.
4. If PEC is enabled, $t_{WR\_RD\_DELAY\_PEC\_EN}$ timing parameter also applies.
5. See Figure 29 for timing definition. See Figure 23 for output timing parameter measurement reference load.
6. See Figure 22 for output timing parameter definition.
7. The SPD5 Hub device must be in configured in I3C Basic mode to guarantee t_{DOUT} value. See Figure 21 for output timing parameter definition. See Figure 23 for output timing parameter measurement reference load.
8. The SPD5 Hub device must be configured in I3C Basic mode to guarantee t_{DOFFT} value, as shown in Figure 10. See Figure 23 for output timing parameter measurement reference load. Please refer to *MIPI I3C Basic Version 1.0-19 July 2018, section 5.1.2.3.2, Transition from Address ACK to Mandatory Byte during IBI*.
9. The SPD5 Hub device must be configured in I3C Basic mode. The host guarantees t_{DOFFC} value, as shown in Figure 11. See Figure 23 for output timing parameter measurement reference load.
10. See Figure 13.
11. From stop condition of DEVCTRL CCC to start condition for register read or register write command data packet delay.
12. The SPD5 Hub device may send ACK or NACK if Host does not satisfy $t_{DEVCTRLCCC\_DELAY\_PEC\_DIS}$ timing parameter.
13. This timing parameter restriction is only applicable when PEC function is disabled in SPD5 Hub. If PEC is enabled, this timing parameter does not apply.
14. From stop condition for register write command data packet to start condition for register read command data packet delay.
15. This timing parameter restriction is only applicable when PEC function is enabled in SPD5 Hub. If PEC is disabled, this timing parameter does not apply.
16. The SPD5 Hub device may send ACK or NACK if host does not satisfy $t_{WR\_RD\_DELAY\_PEC\_EN}$ timing parameter.

Copyright © 2024, NOVOSENSE

9.6. AC Timing Definition

9.6.1. I²C or I3C Basic Bus Timing

The SPD5 Hub device follows the I²C or I3C Basic bus timing requirements. The Figure 20 and Figure 21 show the timing diagram for data bus input and data output parameters.

Figure 20 I²C or I3C Basic Bus AC Input Timing Parameter Definition

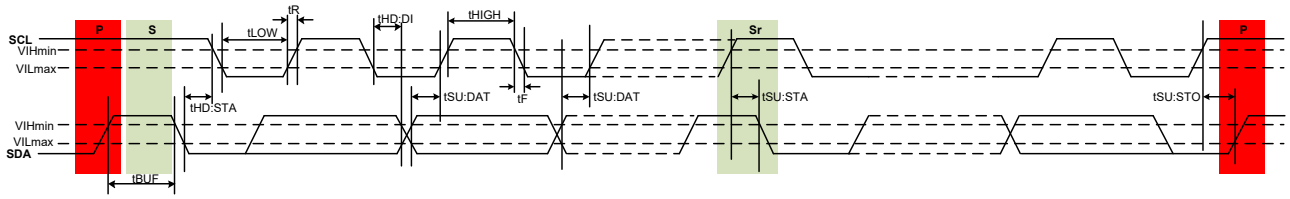


Figure 21 I3C Basic Bus AC Data Output Timing Parameter Definition

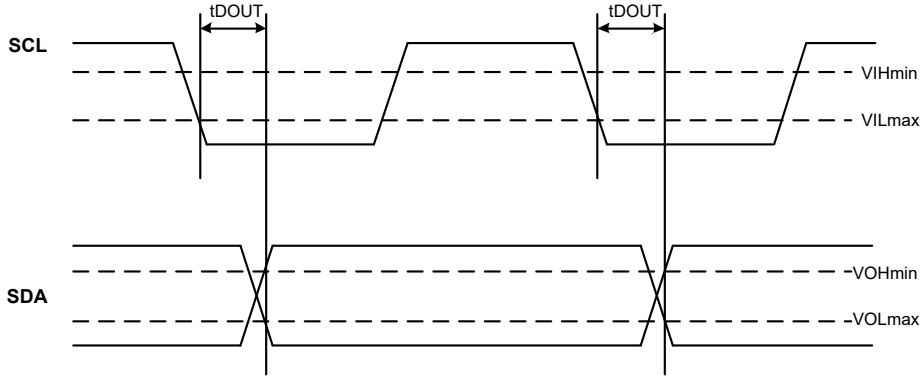


Figure 22 I²C Bus AC Data Output Timing Parameter Definition

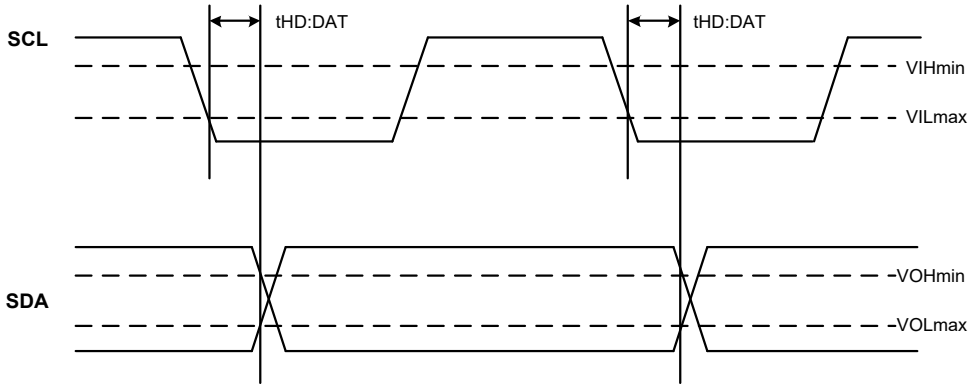
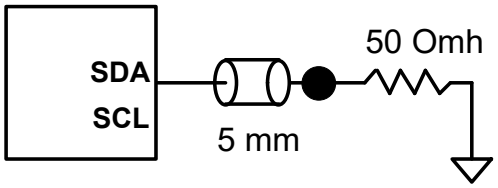


Figure 23 Output Slew Rate and Output Timing Reference Load



Copyright © 2024, NOVOSENSE

Figure 24 Output Slew Rate Measurement Points

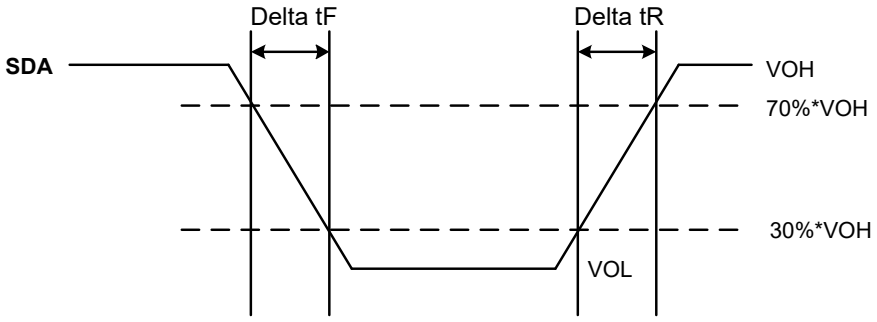
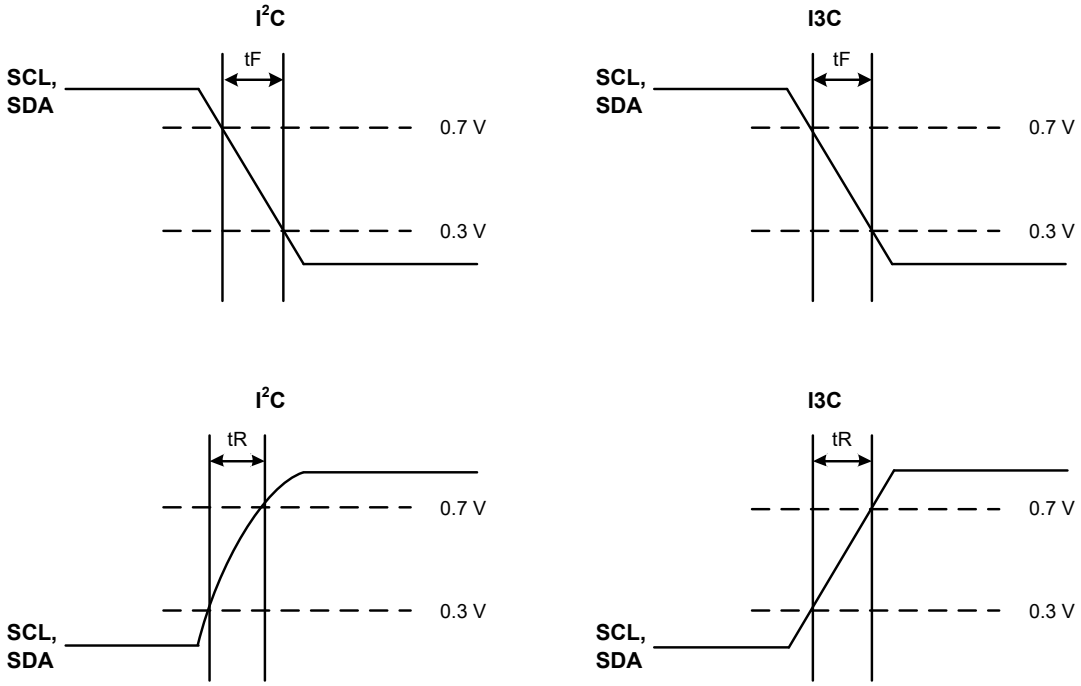


Figure 25 Rise and Fall Timing Parameter Definition



9.6.2.HSCL Non-Monotonicity

Due to non-deterministic loading (number of DIMMs populated) on an unterminated bus, there can be reflections on the bus, causing slope reversal on the HSCL signal on the input receiver of the SPD5 HUB.

The SPD5 Hub device must tolerate t_{SLPR} and V_{SLPR_PK2PK} slope reversal on HSCL in I3C mode as shown in Figure 26 to

Figure 28.

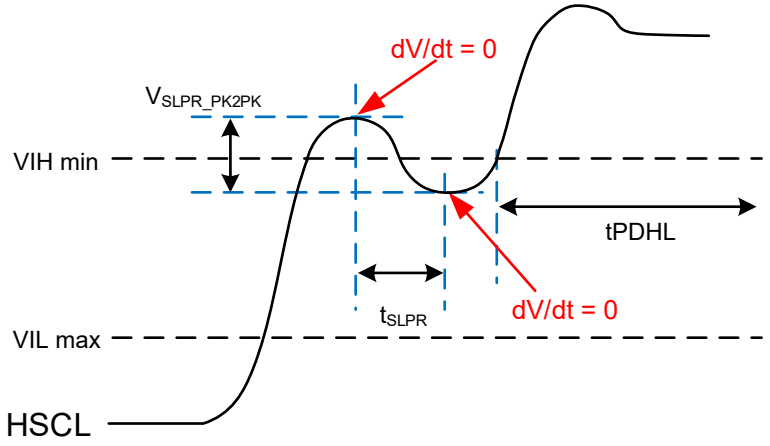
Table 129 HSCL Slope Reversal Parameters

Symbol	Parameter	Min	Max	Unit
t_{SLPR}	Pulse width of slope reversal which must be suppressed[1,2]	0	2.6	ns
V_{SLPR_PK2PK}	The peak-to-peak voltage of slope reversal which must be suppressed[1,2]	0	150	mV

1. Verified by design and characterization, not necessarily tested on all devices.
2. These parameters apply in I3C mode.

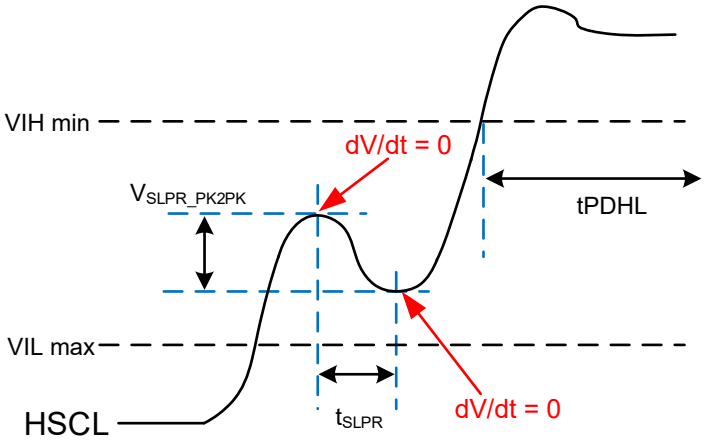
Copyright © 2024, NOVOSENSE

Figure 26 Slope Reversal on HSCL at VIH



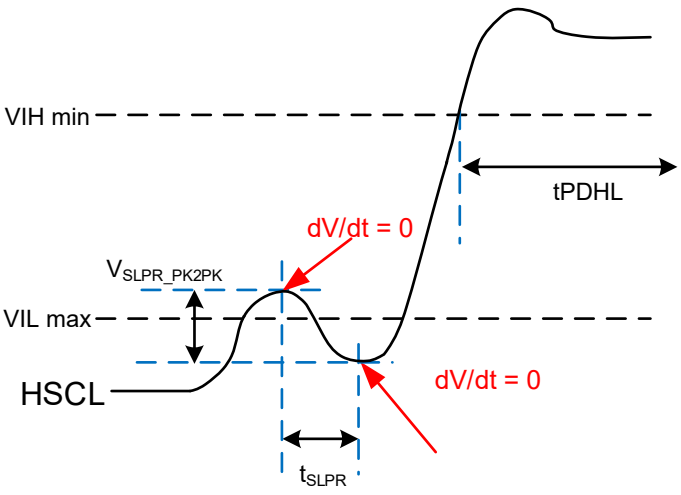
Note: t_{PDHL} is for reference only in this diagram; refer to measurement methodology for details on this parameter.

Figure 27 Slope Reversal on HSCL between VIL and VIH



Note: t_{PDHL} is for reference only in this diagram; refer to measurement methodology for details on this parameter.

Figure 28 Slope Reversal on HSCL at VIL



Note: t_{PDHL} is for reference only in this diagram; refer to measurement methodology for details on this parameter.

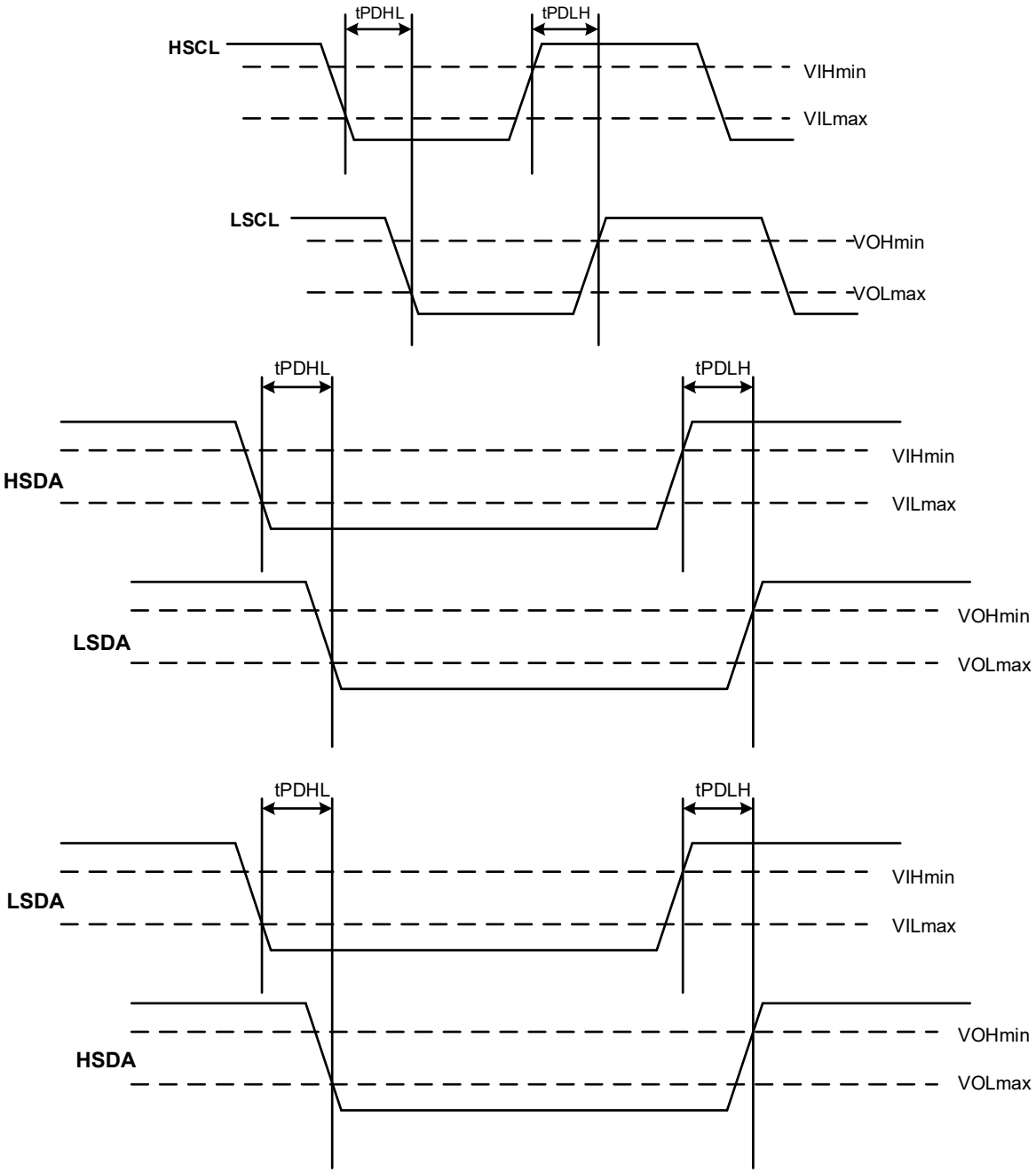
Copyright © 2024, NOVOSENSE

9.6.3. Hub Propagation Delay

The SPD5 Hub device has a propagation delay of t_{PDHL} / t_{PDLH} for its host interface HSCL/HSDA input signals to the local interface LSCL/LSDA signals respectively.

Similarly, the SPD5 Hub device has a propagation delay of t_{PDHL} / t_{PDLH} for its local interface LSDA input signal to the host interface HSDA signal.

Figure 29 Propagation Delay through the SPD5 Hub Device



9.7. Temperature Sensor Performance

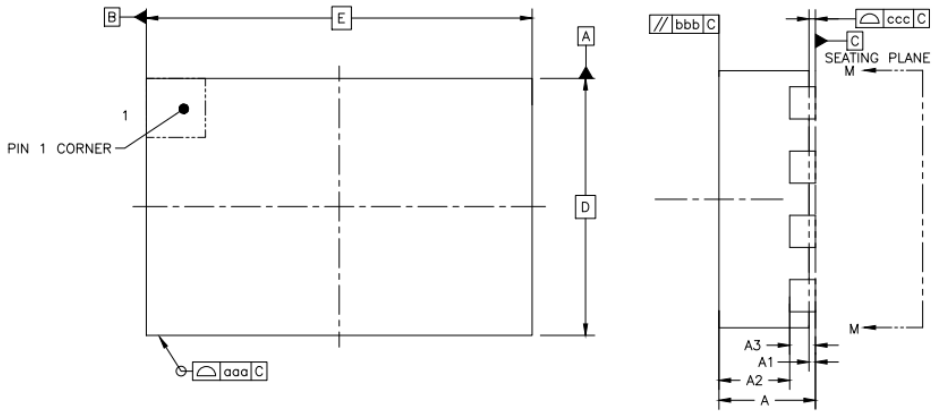
Table 130 Temperature Sensor Performance

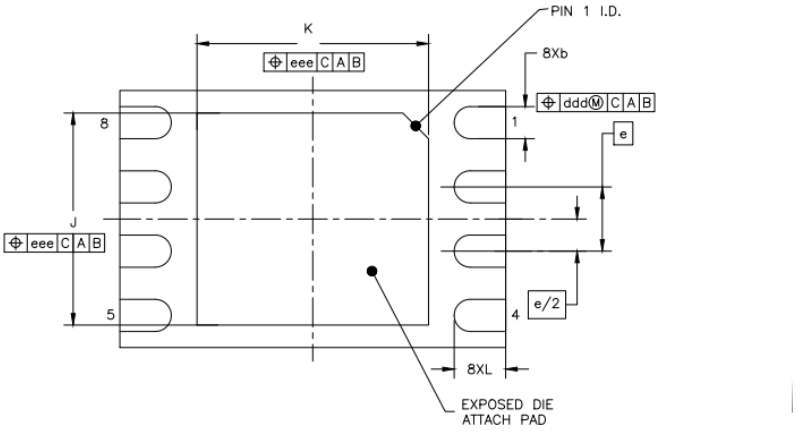
Symbol	Parameter	Test Condition	Min	Typ	Max	Unit
T_{ACC}	Temperature Sensor Accuracy (Active Range)	+75°C ≤ T_A ≤ +95°C	-	±0.5	±1.0	°C
	Temperature Sensor Accuracy (Monitor Range)	+40°C ≤ T_A ≤ +125°C	-	±1.0	±2.0	°C
	Temperature Sensor Accuracy (Industrial Temperature Range)	-40°C ≤ T_A ≤ +125°C	-	±2.0	±3.0	°C
R_{TS}	Temperature Sensor Resolution			0.25		°C
t_{CONV}	Conversion Time	Assumes 0.25 °C accuracy			68	ms
T_{HYST}	Hysteresis between temperature events		1	-	-	°C

Copyright © 2024, NOVOSENSE

10. Package Information

Figure 30 9-Pin 2.0 x 3.0mm DFN Package Outline





DESCRIPTION		SYMBOL	MILLIMETER		
			MIN	NOM	MAX
TOTAL THICKNESS		A	0.50	0.75	0.80
STAND OFF		A1	0.00	---	0.05
MOLD THICKNESS		A2	---	0.55	---
L/F THICKNESS		A3	0.203 REF		
LEAD WIDTH		b	0.20	0.25	0.30
BODY SIZE	X	D	1.95	2.00	2.05
	Y	E	2.95	3.00	3.05
LEAD PITCH		e	0.50 BSC		
EP SIZE	X	J	1.60	1.65	1.70
	Y	K	1.75	1.80	1.85
LEAD LENGTH		L	0.35	0.40	0.45
PACKAGE EDGE TOLERANCE		aaa	0.1		
MOLD FLATNESS		bbb	0.1		
COPLANARITY		ccc	0.08		
LEAD OFFSET		ddd	0.1		
EXPOSED PAD OFFSET		eee	0.1		

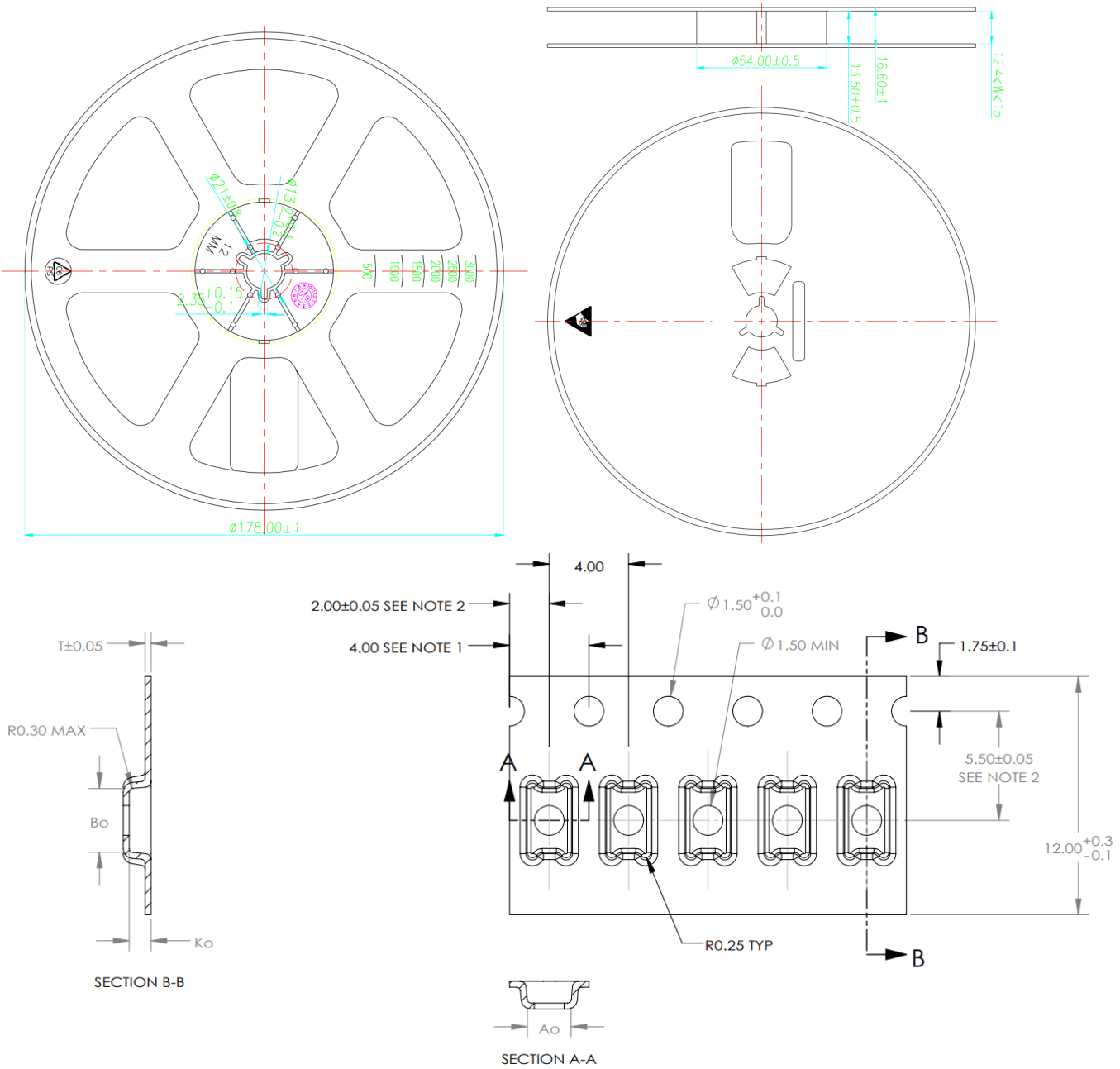
Copyright © 2024, NOVOSENSE

11. Ordering Information

Part Number	Temperature	MSL	Package type	Package Drawing	SPQ
NCA95118-DDAKR	-40°C to 125°C	3	DFN8	DFN8	3000

12. Tape and Reel Information

Figure 31 Tape and Reel Information of DFN8 in Millimeters



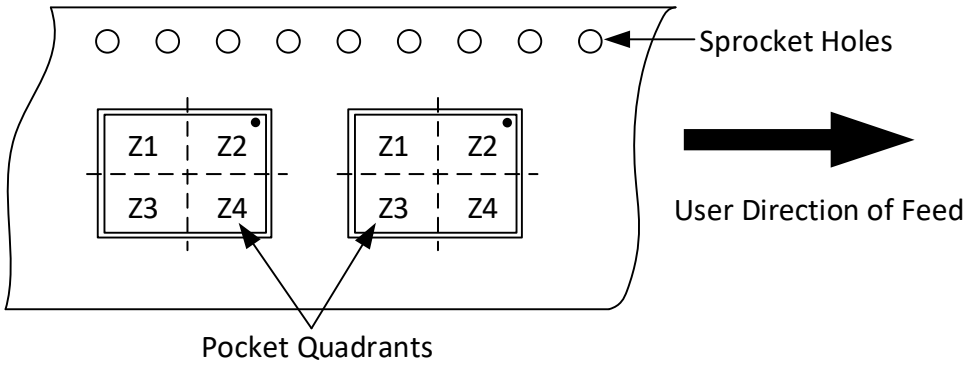
SCALE 1:1

PART#	T	MATERIAL	DRAWING#
ML0302-AC	0.30	PS+C	T103067BT
ML0302-ABB	0.25	PC TRI-LAM.	T119805BT

	DIM	±
Ao	2.20	0.10
Bo	3.20	0.10
Ko	1.10	0.10

NOTES:
1. 10 SPROCKET HOLE PITCH CUMULATIVE TOLERANCE ±0.2
2. POCKET POSITION RELATIVE TO SPROCKET HOLE MEASURED AS TRUE POSITION OF POCKET, NOT POCKET HOLE.
3. Ao AND Bo ARE MEASURED ON A PLANE AT A DISTANCE "R" ABOVE THE BOTTOM OF THE POCKET.

QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE



13. Revision History

Revision	Description	Date
1.0	Initial Version.	2024/8/15

Copyright © 2024, NOVOSENSE

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as any warranty or authorization of, express or implied, including but not limited to accuracy, completeness, merchantability, fitness for a particular purpose or infringement of any third party's intellectual property rights.

You are solely responsible for your use of Novosense' products and applications, and for the safety thereof. You shall comply with all laws, regulations and requirements related to Novosense's products and applications, although information or support related to any application may still be provided by Novosense.

The resources are intended only for skilled developers designing with Novosense' products. Novosense reserves the rights to make corrections, modifications, enhancements, improvements or other changes to the products and services provided. Novosense authorizes you to use these resources exclusively for the development of relevant applications designed to integrate Novosense's products. Using these resources for any other purpose, or any unauthorized reproduction or display of these resources is strictly prohibited. Novosense shall not be liable for any claims, damages, costs, losses or liabilities arising out of the use of these resources.

For further information on applications, products and technologies, please contact Novosense (www.novosns.com).

Suzhou Novosense Microelectronics Co., Ltd

Copyright © 2024, NOVOSENSE