

# **PJ71MXX Datasheet**

## **I2C/I3C Intelligent Multiple Port Hub Devices**

**Version: Rev.1.0**

Release Date: 2026-02-26

## Contents

1 Description .....	1
2 Features .....	2
3 Applications .....	2
4 Pin Configurations (Top View) .....	3
5 Pin Description .....	4
6 Function Block .....	6
7 Ordering Information .....	8
8 Specification .....	9
8.1 Absolute Maximum Ratings .....	9
8.2 Supply and Operating Conditions .....	9
8.3 Electrical Characteristics .....	10
8.3.1 Power Good Threshold Test Setup .....	12
8.4 Timing Parameters – Master Side Port On-Chip Slave Interface .....	13
8.5 Timing Parameters – Slave Side Port Bus Agent .....	14
8.6 Timing Parameters – Device .....	14
8.7 Timing Parameters – Hub Network .....	14
8.8 Timing Waveforms and Timing .....	15
8.8.1 I3C/I2C Bus Waveform .....	15
8.8.2 I3C Read Timing .....	15
8.8.3 Hub Push/Pull Re-driving Mode Propagation Delay .....	15
8.8.4 Hand-off Timing Diagram .....	16
9 Function Description .....	18
9.1 Definitions .....	18
9.2 Master Side Ports .....	19
9.2.1 Overview .....	19
9.2.2 IO Operating Mode and Signaling Level .....	19
9.2.3 On-Chip Slaves .....	20
9.3 Master Multiplexer .....	28
9.3.1 Multiplexer Selection .....	29
9.3.2 Multiplexer Port Switching .....	29
9.4 Hub Network .....	30

9.4.1 Hub Control Logic .....	30
9.4.2 Open Drain Mode .....	31
9.4.3 Push/Pull Mode .....	31
9.4.4 Hub Network Transparency .....	32
9.5 Slave Side Ports .....	32
9.5.1 IO Operating Mode and Signaling Level .....	32
9.5.2 Slave Side Port Features and Configuration .....	33
9.5.3 Port Enable .....	33
9.5.4 Pull-up Resistor .....	34
9.5.5 GPIO Mode .....	34
9.5.6 Agenting .....	34
9.5.7 Hub Network Connection .....	34
9.6 Bus Agent .....	35
9.6.1 Bus Agent Overview .....	35
9.6.2 SMBus Agent Data Buffer .....	36
9.6.3 SMBus Master Agent .....	37
9.6.4 SMBus Slave Agent .....	39
9.6.5 SMBus Agent Related Registers .....	40
9.6.6 Example SMBus Transaction Descriptor Mapping .....	40
9.7 IO Voltage Detector .....	44
10 Registers .....	45
10.1 Register Space and Attribute .....	45
10.2 Register Map .....	45
10.2.1 On-Chip Slave Register Space Map .....	45
10.2.2 Two On-Chip Slaves' Register Space and Sharing/Selection Map .....	48
10.2.3 Paged Register Map .....	51
10.2.4 Master Side Port Message Registers .....	52
10.2.5 Device Level Registers .....	52
10.2.6 PID Registers .....	52
10.3 Register Definition Details .....	53
10.3.1 Device Information Registers .....	53
10.3.2 Device Configuration Registers .....	54

10.3.3 Device Status and IBI Related Registers .....	62
10.3.4 Master-Master Generic Scratch Board Registers .....	64
10.3.5 Master Multiplexer Selection Control and Status Registers .....	65
10.3.6 Slave Dynamic Address Assignment Flag Registers .....	65
10.3.7 Slave Side Port Control Registers .....	66
10.3.8 Slave Side Port Status .....	69
10.3.9 SMBus Agent Status and Configuration Registers .....	70
10.3.10 Special Registers .....	71
10.4 Part Number Specific Configurations .....	74
11 Supported CCC and Frame Mapping .....	75
11.1 I3C Basic 1.1 CCC .....	75
11.1.1 SETBUSCON CCC .....	75
11.2 JEDEC SPD Bus CCCs .....	75
11.2.1 GETSTATUS CCC (JEDEC and I3C Basic 1.0) .....	75
11.2.2 SETAASA CCC (JEDEC and I3C Basic 1.0) .....	76
11.2.3 DEVCTRL CCC (JEDEC SPD Bus) .....	76
11.2.4 ENEC CCC (JEDEC and I3C Basic 1.0) .....	78
11.2.5 DISEC CCC (JEDEC and I3C Basic 1.0) .....	79
11.2.6 RSTDAA CCC (JEDEC and I3C Basic 1.0) .....	81
11.2.7 DEVCAP CCC (JEDEC SPD Bus) .....	82
11.2.8 SETHID CCC (JEDEC SPD Bus) .....	83
11.3 I3C Basic 1.0 standard CCCs .....	83
11.3.1 Supported I3C Basic 1.0 standard CCCs .....	83
11.3.2 PJ71MXX BCR .....	83
11.3.3 PJ71MXX DCR .....	84
12 Theory of Operation .....	85
12.1 Power Supply .....	85
12.2 Initialization .....	86
12.3 Device Reset .....	86
12.4 I3C Interface Reset .....	87
12.5 Bus Reset .....	87
12.6 Bus Clear .....	87
12.7 Bus Context .....	88

12.7.1 Context Setting Command .....	88
12.7.2 JEDEC SPD/Module Management Bus Operation .....	88
12.7.3 Generic I3C Basic 1.0 Bus Operation .....	89
12.7.4 Comparison of the two Bus Context's Operation .....	90
12.8 Two Masters Operation .....	93
12.8.1 I3C Network Management .....	93
12.8.2 Master Selection .....	94
12.8.3 Shared Device Level Register Access .....	94
12.8.4 Dedicated Port Level Register Access .....	94
12.9 IBI Generation and Arbitration .....	94
12.10 Master-Master Communication .....	95
12.10.1 IBI Message Buffer .....	95
12.10.2 Shared Scratch Board Buffers .....	95
12.11 Transparent Hub Network Operation .....	95
12.11.1 I2C/SMBus Open-Drain Only Operation .....	96
12.11.2 I2C/I3C Compatible Mode Operation .....	96
12.11.3 Mixed I2C/I3C Slave support .....	99
12.11.4 Bus Partition .....	99
12.11.5 Fixed Analog Switch Mode .....	99
12.12 Bus Agent Operation .....	100
12.12.1 Topology .....	100
12.12.2 Bus Agent Communication to the I3C Masters .....	101
12.12.3 Bus Agent Communication to the Downstream Devices .....	101
12.12.4 Signal Integrity .....	101
12.12.5 IO Drive Strength Control in Re-Driving Mode .....	101
12.12.6 Dynamic Analog Switch Mode .....	101
12.13 Hot Join .....	102
12.13.1 Hot Join on Slave Side Ports .....	102
12.13.2 Hot Join on Master Side Ports .....	102
12.14 Transaction Rules .....	103
12.14.1 Device Configuration Transaction Rules .....	103
12.14.2 CCC Transaction Rules .....	103
12.15 Device Addressing and Identification .....	103

---

12.15.1 I2C/SMBus Static Addresses .....	103
12.15.2 I3C Dynamic Addresses .....	103
12.16 SMBus Only Operation .....	105
13 Package Outline Dimensions and Recommend Land Pattern Layout .....	106
13.1 Package Outline Dimensions .....	106
13.2 Recommended Land Pattern Layout .....	107
14 Packing information .....	108
15 Version History .....	109
Disclaimer .....	110

## 1 Description

The PJ71MXX is a family of I2C/I3C Multiple Port Hub devices featuring two I2C/I3C Master Side Ports and up to eight I2C/IC Slave Side Ports. A PJ71MXX device provides connectivity to up to two masters (Controllers), and up to eight slaves (Targets). By cascading and duplicating the PJ71MXX, the user builds a PJ71MXX Hub Network to provide connectivity to a number of devices, with extended reach distance, and running with different level of I2C, I3C, and SMBus protocols.

In the PJ71MXX device, each of Master Side Port is associated with an on-chip I2C/I3C slave interface. Each I2C/I3C interface accesses on-chip registers. Master-Master communication is achieved with shared registers and intra-port IBI channels.

One of the two Master Side Ports is selected to connect to a 1:N Hub network, which allows the selected master to get access to the enabled Slave Side Ports in the Hub network. The master selection multiplexer allows the two masters to share the downstream I2C/I3C network.

The 1:N I2C/I3C Hub network allows the management of the I2C/I3C hierarchy with expansion to up to eight Slave Side Ports. The expanded ports allow the system to reduce the load that the selected master sees at any moment. The Hub network maintains software level transparently. All devices connected to the Hub Slave Side Ports are accessed the same way as if all ports directly connected as if the Hub network does not exist. The Hub network also allows physical segmentation of the I2C/I3C hierarchy employing on the fly connecting and disconnecting to any of the expanded ports.

Each Slave Side Port is associated with a Bus Agent. This Agent independently receives or transmits legacy SMBus transactions and allows SCL Stretching within the SMBus segment behind the Slave Side Port. When the SMBus Agent is active, the Agent is engaged directly with the Slave Side Port and the Hub Network to the port is disconnected.

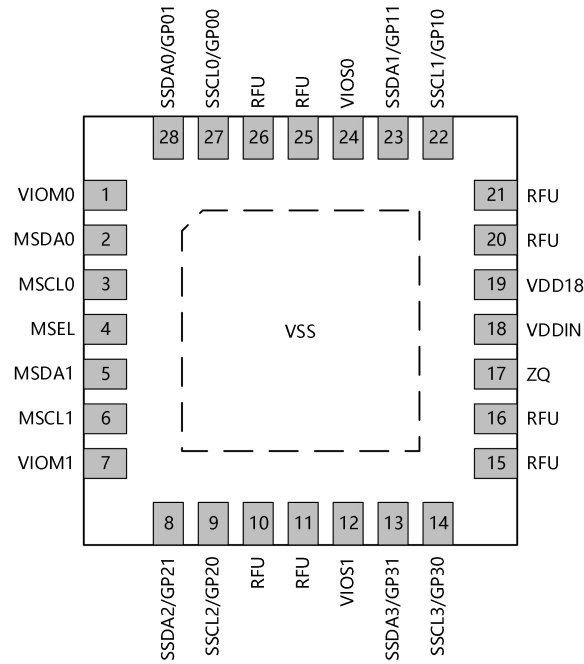
## 2 Features

- ◆ Two Master Side (upstream/controller) Ports
- ◆ Up to eight Slave Side (downstream/target) Ports
- ◆ On-Chip I2C/I3C Slave (target) Interface
- ◆ Master Side Port multiplexer
- ◆ Master switching via in-band I2C/I3C commands or pin selection
- ◆ Master-Master communication and messaging support
- ◆ Hot Join support
- ◆ Support both I2C and I3C Basic 1.0 Protocol
- ◆ All ports support 1.0 to 1.8V I2C/I3C compatible operation
- ◆ All ports support I2C Open-Drain only operation up to 3.3V
- ◆ JEDEC SPD/Module Management Bus Context support
- ◆ Master to Slave analog switch mode support
- ◆ Support analog switch mode
- ◆ Support re-driving mode
- ◆ Mixed I2C/I3C Bus support
- ◆ In-Band Interrupt (IBI) and IBI Optimization support
- ◆ Single 3.3V power supply
- ◆ Master-Slave Level shifting support
- ◆ On-Chip Voltage Regulators or direct platform connection of IO supplies
- ◆ SMBus transaction agent for SMBus compatibility
- ◆ SMBus SCL stretching support
- ◆ SMBus SCL stretching support in Open-Drain Only Operation
- ◆ Network partition support
- ◆ Support non-transparent bridging

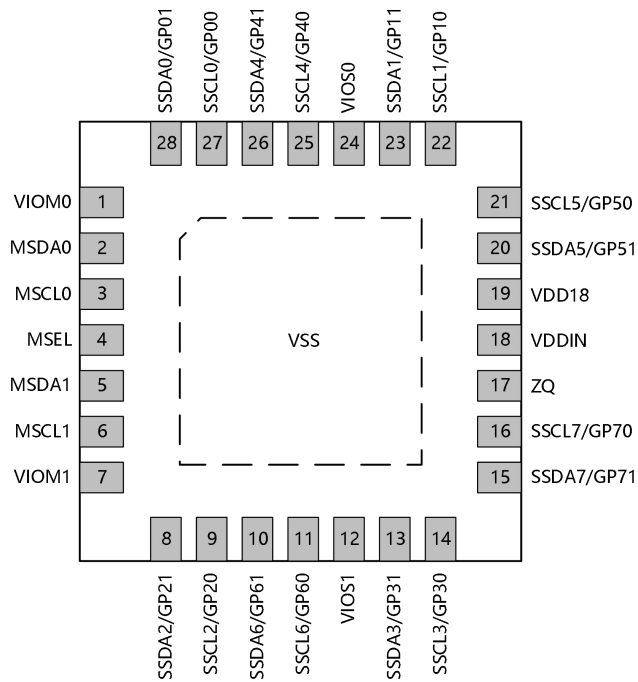
## 3 Applications

- ◆ DIMM I3C fan-out
- ◆ I3C/I2C/SMBus compatibility
- ◆ I3C Port Expansion and Level adaptation
- ◆ I3C Topology Management
- ◆ I3C Signal Integrity Management

## 4 Pin Configurations (Top View)



**Figure1.PJ71M4XX Pin Assignments**



**Figure2.PJ71M8XX Pin Assignments**

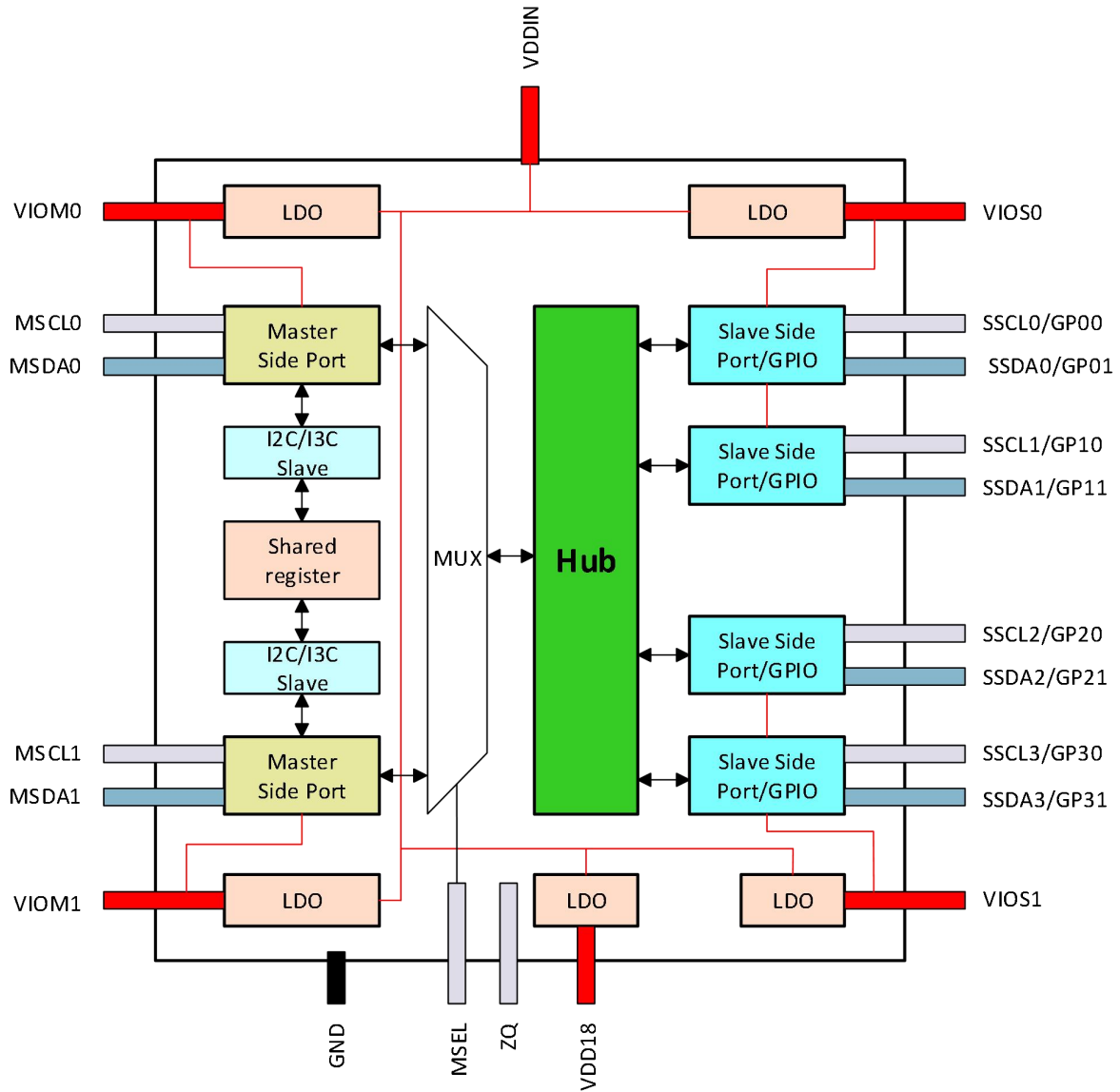
## 5 Pin Description

**Table1.Pin Description**

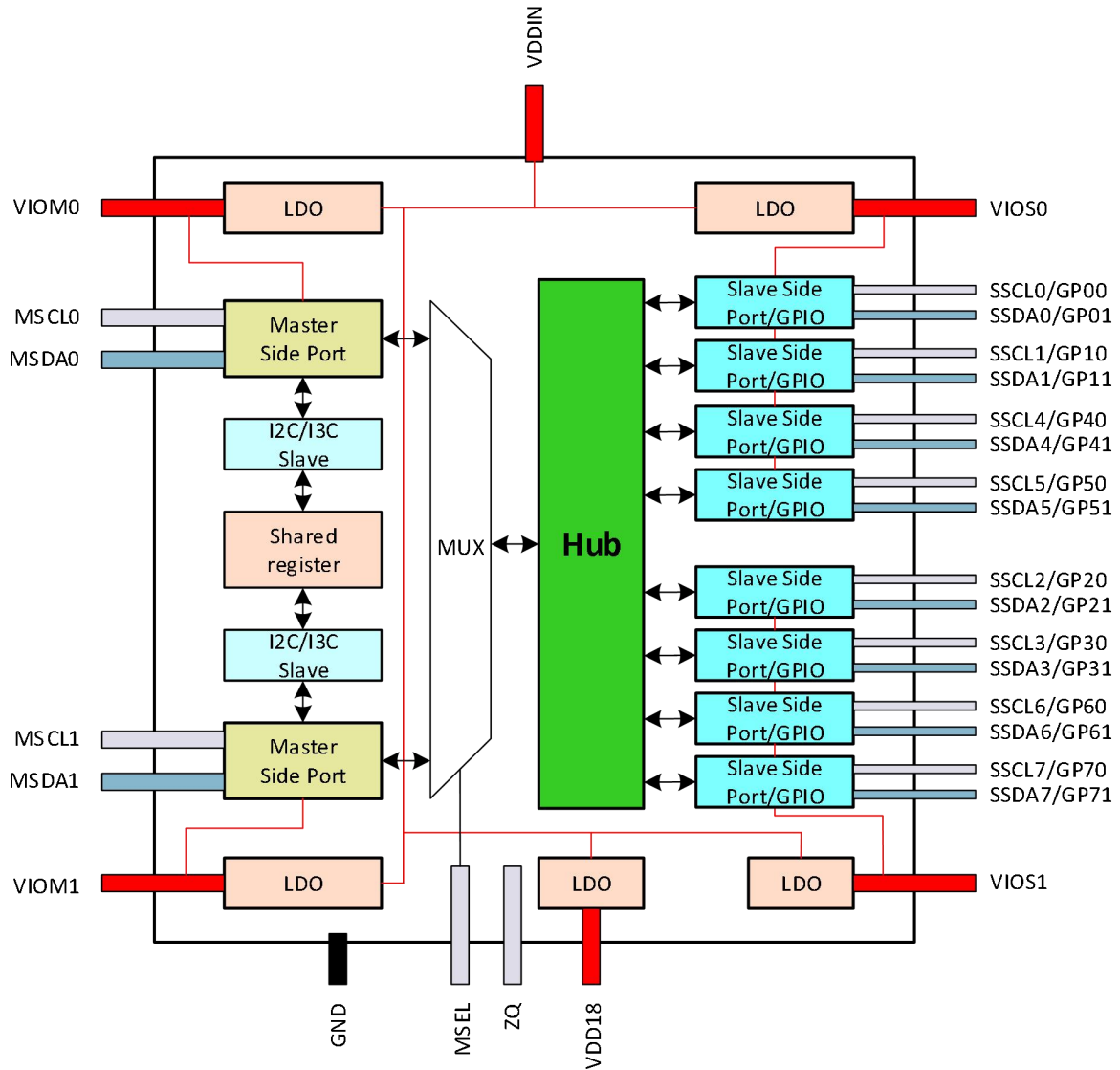
Pin Name	Pin No.	Pin Type	Description
VIOM0	1	Power	On-Chip LDO output or Push/Pull mode IO supply input for Port M0 IO. 1.0V, 1.1V, 1.2V, or 1.8V nominal. Connects to a 1uF Cap.
MSDA0	2	Bidirectional	Master Side Port 0 SDA signal.
MSCL0	3	Input	Master Side Port 0 SCL signal.
MSEL	4	Input	Master Select Input. Logic level compatible with 1.0V, and is 3.3V IO logic. This pin can be floated. If this pin is not actively driven, it will be internally biased to around 0.5V.
MSDA1	5	Bidirectional	Master Side Port 1 SDA signal.
MSCL1	6	Input	Master Side Port 1 SCL signal.
VIOM1	7	Power	On-Chip LDO output or Push/Pull mode IO supply input for Port M1 IO. 1.0V, 1.1V, 1.2V, or 1.8V nominal. Connects to a 1uF Cap.
SSDA2/GP21	8	Bidirectional	Slave Side Port 2 SDA signal, or GPIO P2[1]. GPIO function only applies to devices that support GPIO.
SSCL2/GP20	9	Bidirectional	Slave Side Port 2 SCL signal, or GPIO P2[0]. GPIO function only applies to devices that support GPIO.
SSDA6/GP61	10	Bidirectional	Slave Side Port 6 SDA signal, or GPIO P6[1]. GPIO function only applies to devices that support GPIO. This pin exists in eight Slave Side Port Devices.
SSCL6/GP60	11	Bidirectional	Slave Side Port 6 SCL signal, or GPIO P6[0]. GPIO function only applies to devices that support GPIO. This pin exists in eight Slave Side Port Devices.
VIOS1	12	Power	On-Chip LDO output or Push/Pull mode IO supply input for Port S2/S3/S6/S7, 1.0V, 1.1V, 1.2V, or 1.8V nominal. Connects to a 1uF Cap.
SSDA3/GP31	13	Bidirectional	Slave Side Port 3 SDA signal, or GPIO P3[1]. GPIO function only applies to devices that support GPIO.
SSCL3/GP30	14	Bidirectional	Slave Side Port 3 SCL signal, or GPIO P3[0]. GPIO function only applies to devices that support GPIO.
SSDA7/GP71	15	Bidirectional	Slave Side Port 7 SDA signal, or GPIO P7[1]. GPIO function only applies to devices that support GPIO. This pin exists in eight Slave Side Port Devices.
SSCL7/GP70	16	Bidirectional	Slave Side Port 7 SCL signal, or GPIO P7[0]. GPIO function only applies to devices that support GPIO. This pin exists in eight Slave Side Port Devices.
ZQ	17	I/O	Device Calibration, Connect a 1% 240Ω resistor to Ground.
VDDIN	18	Power	Regulator Power Supply, 3.3V nominal. Connects to a 20uF Cap.
VDD18	19	Power	On-Chip LDO output, and Chip supply, 1.8V nominal. Connects to a 1uF Cap.
SSDA5/GP51	20	Bidirectional	Slave Side Port 5 SDA signal, or GPIO P5[1]. GPIO function only applies to devices that support GPIO. This pin exists in eight Slave Side Port Devices.
SSCL5/GP50	21	Bidirectional	Slave Side Port 5 SCL signal, or GPIO P5[0]. GPIO function only applies to devices that support GPIO. This pin exists in eight Slave Side Port Devices.
SSCL1/GP10	22	Bidirectional	Slave Side Port 1 SCL signal, or GPIO P1[0]. GPIO function only applies to devices that support GPIO.
SSDA1/GP11	23	Bidirectional	Slave Side Port 1 SDA signal, or GPIO P1[1]. GPIO function only applies to devices that support GPIO.

Pin Name	Pin No.	Pin Type	Description
VIOS0	24	Power	On-Chip LDO output or Push/Pull mode IO supply input for Port S0/S1/S4/S5, 1.0V, 1.1V, 1.2V, or 1.8V nominal. Connects to a 1uF Cap.
SSCL4/GP40	25	Bidirectional	Slave Side Port 4 SCL signal, or GPIO P4[0]. GPIO function only applies to devices that support GPIO. This pin applies to eight Slave Side Port Devices.
SSDA4/GP41	26	Bidirectional	Slave Side Port 4 SDA signal, or GPIO P4[1]. GPIO function only applies to devices that support GPIO. This pin exists in eight Slave Side Port Devices.
SSCL0/GP00	27	Bidirectional	Slave Side Port 0 SCL signal, or GPIO P0[0]. GPIO function only applies to devices that support GPIO.
SSDA0/GP01	28	Bidirectional	Slave Side Port 0 SDA signal, or GPIO P0[1]. GPIO function only applies to devices that support GPIO.
VSS	29 (EPAD)	Ground	Ground.
RFU	-	-	Reserved for Future Usage. Those pins need to be left no-correction in the system.

## 6 Function Block



**Figure3.PJ71M49BQW Block Diagram**



**Figure4.PJ71M89BQW Block Diagram**

## 7 Ordering Information

**Table2.Order Information**

Order PN	Marking ID	Package	Definition	MSL	Description
PJ71M47BQW	71M47B YMDNN	QFN4x4-28	71M47B:Product code YMDNN : Y : Year code M : Month code D : Day code NN: Serial Number	Level-3	Halogen free RoHS compliant in T/R,5,000 pcs/Reel
PJ71M48BQW	71M48B YMDNN	QFN4x4-28	71M48B:Product code YMDNN : Y : Year code M : Month code D : Day code NN: Serial Number	Level-3	Halogen free RoHS compliant in T/R,5,000 pcs/Reel
PJ71M49BQW	71M49B YMDNN	QFN4x4-28	71M49B:Product code YMDNN : Y : Year code M : Month code D : Day code NN: Serial Number	Level-3	Halogen free RoHS compliant in T/R,5,000 pcs/Reel
PJ71M87BQW	71M87B YMDNN	QFN4x4-28	71M87B:Product code YMDNN : Y : Year code M : Month code D : Day code NN: Serial Number	Level-3	Halogen free RoHS compliant in T/R,5,000 pcs/Reel
PJ71M88BQW	71M88B YMDNN	QFN4x4-28	71M88B:Product code YMDNN : Y : Year code M : Month code D : Day code NN: Serial Number	Level-3	Halogen free RoHS compliant in T/R,5,000 pcs/Reel
PJ71M89BQW	71M89B YMDNN	QFN4x4-28	71M89B:Product code YMDNN : Y : Year code M : Month code D : Day code NN: Serial Number	Level-3	Halogen free RoHS compliant in T/R,5,000 pcs/Reel

Notes:

(1) MetaWells can meet RoHS 2.0/REACH requirement. So most package types MetaWells offers only states halogen free, instead of lead free.

**Table3.Power Up Feature Defaults for Different Part Number**

Function	Feature	Feature's Power-Up Default Setting					
		PJ71M87BQW	PJ71M47BQW	PJ71M88BQW	PJ71M48BQW	PJ71M89BQW	PJ71M49BQW
Controller Side Ports	IO Operating Mode	Open Drain Only (I2C/SMBUS)	Open Drain Only (I2C/SMBUS)	Open Drain/Push-Pull compatible (I3C)	Open Drain/Push-Pull compatible (I3C)	Open Drain Only (I2C/SMBUS)	Open Drain Only (I2C/SMBUS)
	VIOM Voltage (Interface Voltage)	1.0, up to 3.3V	1.0, up to 3.3V	1.8V	1.8V	1.1V	1.8V
	VIOM Internal LDO	Disabled	Disabled	Disabled	Disabled	Enabled	Enabled
Target Side Ports	Connection to Hub Network	No	No	Yes	Yes	No	No
	IO Operating Mode	Open Drain Only (I2C/SMBUS)	Open Drain Only (I2C/SMBUS)	Open Drain/Push-Pull compatible (I3C)	Open Drain/Push-Pull compatible (I3C)	Open Drain Only (I2C/SMBUS)	Open Drain Only (I2C/SMBUS)
	VIOS Voltage (Interface Voltage)	1.0, up to 3.3V	1.0, up to 3.3V	1.8V	1.8V	1.0V	1.1V
Hub Network	Operating mode	I2C	I2C	I3C	I3C	I2C	I2C

## 8 Specification

### 8.1 Absolute Maximum Ratings

Stressing the device above the rating listed in the Absolute Maximum Rating 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.

**Table4.Absolute Maximum Ratings**

Symbol	Parameter	Min	Max	Unit
T <sub>STG</sub>	Storage temperature	-65	150	°C
V <sub>DDIN</sub>	Chip Supply Voltage	-0.5	3.8	V
V <sub>IOx</sub>	IO Supply Voltage	-0.5	2.1	V
V <sub>pin</sub>	MSEL	-0.5	3.8	V
V <sub>pin</sub>	MSCLx, MSDAx, SSCLx, SSDAx; OD Only Mode	-0.5	3.8	V
V <sub>pin</sub>	MSCLx, MSDAx, SSCLx, SSDAx; I3C compatible Mode	-0.5	2.1	V

### 8.2 Supply and Operating Conditions

**Table5.Supply and Operating Conditions**

Symbol	Parameter	Min	Typ	Max	Unit
V <sub>DDIN</sub>	Input Supply Voltage	1.7	3.3	3.6	V
V <sub>DD18</sub>	Logic Supply Voltage, Generated by LDO	1.7	1.8	1.9	V
V <sub>IO_1.0</sub>	IO Supply Voltage, LDO Output set to 1.0V	0.95	1.0	1.05	V
V <sub>IO_1.1</sub>	IO Supply Voltage, LDO Output set to 1.1V	1.05	1.1	1.15	V
V <sub>IO_1.2</sub>	IO Supply Voltage, LDO Output set to 1.2V	1.15	1.2	1.25	V
V <sub>IO_1.8</sub>	IO Supply Voltage, LDO Output set to 1.8V	1.7	1.8	1.9	V
T <sub>A</sub>	Ambient Operation Temperature	-40	-	85	°C
T <sub>J</sub>	Junction Operating Temperature	-40	-	125	°C

### 8.3 Electrical Characteristics

If not explicitly specified, the test condition will be nominal voltage, room temperature.

**Table6.Electrical Characteristics**

Symbol	Parameter	Test Condition	Min	Max	Unit
IDD	VDDIN Current	VDDIN=3.3V, Controller/Target Ports in idle state	-	6	mA
SW_R <sub>ON</sub>	Analog Switch Path Resistance	DC level around 0.5V	-	10	Ω
C <sub>IO</sub>	IO pin Capacitance	100Mhz, DC = 0.5V	-	6	pF
C <sub>SW</sub>	Analog Switch Path total Capacitance, including IO pin capacitance	100Mhz, DC = 0.5V, Master Side Port and Slave Side Port connected by Analog Switch	-	25	pF
Z <sub>o_50</sub>	Driver Impedance at 50-ohm setting	DC voltage at 0.5*V <sub>IO</sub> Temperature Range: 0~85°C, nominal V <sub>IO</sub>	40	60	Ω
Z <sub>o_40</sub>	Driver Impedance at 40Ω setting	DC voltage 0.5*V <sub>IO</sub> Temperature Range: 0~85°C, nominal V <sub>IO</sub>	32	48	Ω
Z <sub>o_30</sub>	Driver Impedance at 30Ω setting	DC voltage 0.5*V <sub>IO</sub> Temperature Range: 0~85°C, nominal V <sub>IO</sub>	24	36	Ω
Z <sub>o_20</sub>	Driver Impedance at 20Ω setting	DC voltage 0.5*V <sub>IO</sub> Temperature Range: 0~85°C, nominal V <sub>IO</sub>	16	24	Ω
R <sub>pu_2k_33</sub>	2kΩ Pull-up Resistor value to V <sub>DDIN</sub> 3.3V	REG#25, SSPzORTS_SDA_Pullup resistor setting = 2kΩ; REG#23, OD_Only bit is set	1.65 <sup>[1]</sup>	2.30	KΩ
R <sub>pu_2k_18</sub>	2kΩ Pull-up Resistor value to V <sub>IO</sub> 1.8V	REG#25, SSPORTS_SDA_Pullup resistor setting = 2kΩ; REG#23, OD_Only bit is cleared	1.65	2.35	KΩ
R <sub>pu_2k_10</sub>	2kΩ Pull-up Resistor value to V <sub>IO</sub> 1.0V	REG#25, SSPORTS_SDA_Pullup resistor setting = 2kΩ; REG#23, OD_Only bit is cleared	1.60	2.20	KΩ
R <sub>pu_1k_33</sub>	1kΩ Pull-up Resistor value to V <sub>DDIN</sub> 3.3V	REG#25, SSPORTS_SDA_Pullup resistor setting = 1kΩ; REG#23, OD_Only bit is set	0.93	1.35	KΩ
R <sub>pu_1k_18</sub>	1kΩ Pull-up Resistor value to V <sub>IO</sub> 1.8V	REG#25, SSPORTS_SDA_Pullup resistor setting = 1kΩ; REG#23, OD_Only bit is cleared	0.88	1.35	KΩ
R <sub>pu_1k_10</sub>	1kΩ Pull-up Resistor value to V <sub>IO</sub> 1.0V	REG#25, SSPORTS_SDA_Pullup resistor setting = 1kΩ; REG#23, OD_Only bit is cleared	0.84	1.20	KΩ
R <sub>pu_500_33</sub>	500Ω Pull-up Resistor value to V <sub>DDIN</sub> 3.3V	REG#25, SSPORTS_SDA_Pullup resistor setting = 500Ω; REG#23, OD_Only bit is set	0.45	0.66	KΩ
R <sub>pu_500_18</sub>	500Ω Pull-up Resistor value to V <sub>IO</sub> 1.8V	REG#25, SSPORTS_SDA_Pullup resistor setting = 500Ω; REG#23, OD_Only bit is cleared	0.44	0.66	KΩ
R <sub>pu_500_10</sub>	500Ω Pull-up Resistor value to V <sub>IO</sub> 1.0V	REG#25, SSPORTS_SDA_Pullup resistor setting = 500Ω; REG#23, OD_Only bit is cleared	0.41	0.58	KΩ
R <sub>pu_250_33</sub>	250Ω Pull-up Resistor value to V <sub>DDIN</sub> 3.3V	REG#25, SSPORTS_SDA_Pullup resistor setting = 250Ω; REG#23, OD_Only bit is set	0.23	0.33	KΩ
R <sub>pu_250_18</sub>	250Ω Pull-up Resistor value to V <sub>IO</sub> 1.8V	REG#25, SSPORTS_SDA_Pullup resistor setting = 250Ω; REG#23, OD_Only bit is cleared	0.22	0.33	KΩ
R <sub>pu_250_10</sub>	250Ω Pull-up Resistor value to V <sub>IO</sub> 1.0V	REG#25, SSPORTS_SDA_Pullup resistor setting = 250Ω; REG#23, OD_Only bit is cleared	0.22	0.33	KΩ
V <sub>ILM_OD</sub>	Input low voltage Master Side Ports in OD_Only Mode	DC swipe, OD Only Operation	-0.3	0.5	V
V <sub>IHM_OD</sub>	Input high voltage Master Side Ports in OD_Only Mode	DC swipe, OD Only Operation	0.7	3.6	V
V <sub>ILM</sub>	Input low voltage Master Side Ports in OD/PP compatible Mode	DC swipe	-0.3	0.35V <sub>IO</sub>	V

Symbol	Parameter	Test Condition	Min	Max	Unit
V <sub>IHM</sub>	Input high voltage Master Side Ports in OD/PP compatible Mode	DC swipe	0.65*V <sub>IO</sub>	V <sub>IO</sub> <sup>[2]</sup> +0.3	V
V <sub>ILS_OD</sub>	Input low voltage Slave Side Ports in OD_Only Mode	DC swipe, OD Only Operation	-0.3	0.6	V
V <sub>IHS_OD</sub>	Input high voltage Slave Side Ports in OD_Only Mode	DC swipe, OD Only Operation	0.85	V <sub>DDIIN</sub> +0.3	V
V <sub>ILS</sub>	Input low voltage Slave Side Ports in OD/PP compatible Mode	DC swipe, Push/Pull Operation, GPIO, Bus Agent Mode	-0.3	0.35*V <sub>IOS</sub>	V
V <sub>IHS</sub>	Input high voltage Slave Side Ports in OD/PP compatible Mode	DC swipe, Push/Pull Operation, GPIO, Bus Agent Mode	0.65*V <sub>IOS</sub>	V <sub>IOS</sub> <sup>[3]</sup> +0.3	V
R <sub>DRV_EXT</sub>	Slave Side Port external slave device's pull down driver impedance	Measure at V <sub>ILS</sub> Level.	-	100	Ω
C <sub>LOAD</sub>	Slave Side Port maximum Capacitive Load	Lumped capacitance, including the capacitance of the PCB trace and device pin capacitance		200 <sup>[4]</sup>	pF
V <sub>IL_MSEL</sub>	Input low voltage of MSEL pin	DC swipe	-0.3	0.3	V
V <sub>IH_MSEL</sub>	Input high voltage of MSEL pin	DC swipe	0.75	3.6	V
I <sub>LMSEL</sub> <sup>[5]</sup>	MSEL leakage current tolerance	Leakage current limit of keeping the MSEL as HiZ state	-	3	uA
R <sub>DRV_MSEL</sub>	MSEL Pin driver resistance for static logic setting <sup>[6]</sup>	The resistance of IO driver (or tie high/tie-low resistor) drives MSEL pin to static logic high or logic low.	-	1	KΩ
V <sub>OL_ODL</sub>	Output low voltage I2C/I3C Ports	DC swipe, with 1K internal pull-up, V <sub>IO</sub> <1.5V	-	0.25	V
V <sub>OL_ODH</sub>	Output low voltage I2C/I3C Ports	DC swipe, with 1K internal pull-up, V <sub>IO</sub> >=1.5V	-	0.35	V
V <sub>OL_PP</sub>	Output low voltage I2C/I3C Ports	DC swipe, I <sub>load</sub> = 4mA	-	Z <sub>o</sub> <sup>[7]</sup> *I <sub>load</sub> * 1.25	V
V <sub>OH_PP</sub>	Output high voltage I2C/I3C Ports	DC swipe, 4mA load, V <sub>IO</sub> =1.0V, 1.1V, 1.2V, 1.8V; Zo calibrated	V <sub>IO</sub> - Z <sub>o</sub> <sup>[8]</sup> *I <sub>load</sub> * 1.3	-	V
V <sub>IO_PG_Clr_Lo_10</sub> <sup>[9]</sup>	V <sub>IO</sub> Power Good Status Clear Low Threshold for 1.0V setting	DC swipe, check Power Good Status Register	0.85	0.9	V
V <sub>IO_PG_Clr_Hi_10</sub> <sup>[10]</sup>	V <sub>IO</sub> Power Good Status Clear High Threshold for 1.0V setting	DC swipe, check Power Good Status Register	1.1	1.17	V
V <sub>IO_PG_Clr_Lo_11</sub>	V <sub>IO</sub> Power Good Status Clear Low Threshold for 1.1V setting	DC swipe, check Power Good Status Register	0.92	0.99	V
V <sub>IO_PG_Clr_Hi_11</sub>	V <sub>IO</sub> Power Good Status Clear High Threshold for 1.1V setting	DC swipe, check Power Good Status Register	1.21	1.29	V
V <sub>IO_PG_Clr_Lo_12</sub>	V <sub>IO</sub> Power Good Status Clear Low Threshold for 1.2V setting	DC swipe, check Power Good Status Register	1.01	1.08	V
V <sub>IO_PG_Clr_Hi_12</sub>	V <sub>IO</sub> Power Good Status Clear High Threshold for 1.2V setting	DC swipe, check Power Good Status Register	1.32	1.39	V
V <sub>IO_PG_Clr_Lo_18</sub>	V <sub>IO</sub> Power Good Status Clear Low Threshold for 1.8V setting	DC swipe, check Power Good Status Register	1.51	1.62	V
V <sub>IO_PG_Clr_Hi_18</sub>	V <sub>IO</sub> Power Good Status Clear High Threshold for 1.8V setting	DC swipe, check Power Good Status Register	1.98	2.09	V
HYS_PG_10 <sup>[11]</sup>	V <sub>IO</sub> Power Good Set/Clear Threshold Hysteresis for 1.0V V <sub>IO</sub>	DC swipe, check Power Good Status Register	0.033	0.035	V
HYS_PG_11	V <sub>IO</sub> Power Good Set/Clear Threshold Hysteresis for 1.1V V <sub>IO</sub>	DC swipe, check Power Good Status Register	0.033	0.035	V

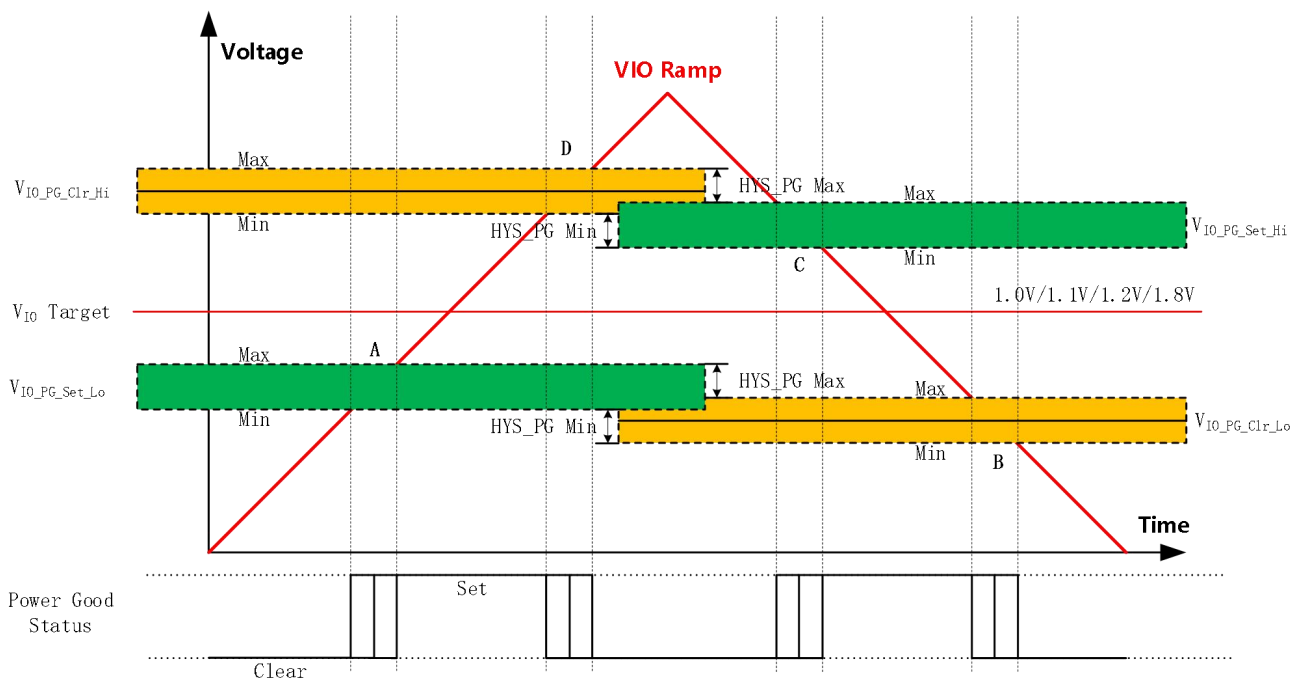
Symbol	Parameter	Test Condition	Min	Max	Unit
HYS_PG_12	V <sub>IO</sub> Power Good Set/Clear Threshold Hysteresis for 1.2V V <sub>IO</sub>	DC swipe, check Power Good Status Register	0.033	0.035	V
HYS_PG_18	V <sub>IO</sub> Power Good Set/Clear Threshold Hysteresis for 1.8V V <sub>IO</sub>	DC swipe, check Power Good Status Register	0.048	0.052	V
V <sub>IO_PG_Set_Lo</sub>	V <sub>IO</sub> (1.0/1.1/1.2/1.8V) Power Good Status Set Low Threshold	DC swipe, check Power Good Status Register	V <sub>IO_PG_Clr_Lo_10/11/12/18</sub> + HYS_PG_10/11/12/18		V
V <sub>IO_PG_Set_Hi</sub>	V <sub>IO</sub> (1.0/1.1/1.2/1.8V) Power Good Status Set High Threshold	DC swipe, check Power Good Status Register	V <sub>IO_PG_Clr_Hi_10/11/12/18</sub> - HYS_PG_10/11/12/18		V

Note:

- (1). Pull-up Resistor value is measured at 0.3V DC; Junction Temperature range -40 ~ 125°C.
- (2). V<sub>IO</sub> supply of the input port.
- (3). V<sub>IO</sub> supply of the input port.
- (4). If the capacitive load is greater than this number, please consult with MetaWells for best device settings.
- (5). Board level leakage plus MSEL pin driver HiZ state leakage can not exceed this limit, to have MSEL pin consider this pin in HiZ state.
- (6). This is the maximum impedance allowed to drive a static logic. For dynamic MSEL switching, timing parameter tMSEL\_TR needs also to be met. User need to check driver impedance and total capacitance at the MSEL pin.
- (7). Z<sub>o</sub> can be set by programming REG#20, and can be re-calibrated in the application
- (8). Z<sub>o</sub> can be set by programming REG#20, and can be re-calibrated in the application
- (9). Refer to "V<sub>IO\_PG\_Clr\_Lo</sub>" in the diagram in 3.3.1Power Good Threshold Test Setup
- (10). Refer to "V<sub>IO\_PG\_Clr\_Hi</sub>" in the diagram in 3.3.1Power Good Threshold Test Setup
- (11). Refer to "HYS\_PG" in the diagram in 3.3.1Power Good Threshold Test Setup

### 8.3.1 Power Good Threshold Test Setup

The Power Good Status of V<sub>IO0</sub>/V<sub>IO1</sub>/V<sub>IO50</sub>/V<sub>IO51</sub> gets set and cleared at different threshold range when V<sub>IO</sub> ramp approaching to the V<sub>IO</sub> target and departing from the V<sub>IO</sub> target.



## 8.4 Timing Parameters – Master Side Port On-Chip Slave Interface

**Table7.Timing Parameters – Master Side Port On-Chip Slave Interface**

Symbol		Parameter	I2C and I3C Open Drain		I3C Push-Pull		Unit
			Min	Max	Min	Max	
PJ71MXX	MIPI						
f <sub>SCL</sub>	f <sub>SCL</sub>	Clock frequency	0.01	1	0.01	12.5	MHz
t <sub>HIGH</sub>	t <sub>HIGH</sub>	Clock pulse width high time	260	-	24	-	ns
t <sub>LOW</sub>	t <sub>LOW</sub>	Clock pulse width low time	500	-	24	-	ns
t <sub>TIMEOUT</sub>	-	Detect clock low timeout time	0.8*Timeout <sup>[1]</sup>	1.2*Timeout	0.8*Timeout	1.2*Timeout	ms
t <sub>R</sub>	t <sub>r,CL</sub> , t <sub>r,DA</sub> ,t <sub>CR</sub>	SCL/SDA rise time	-	120	-	150e6/f <sub>SCL</sub> <sup>[2]</sup>	ns
t <sub>F</sub>	t <sub>f,CL</sub> , t <sub>f,DA</sub> , t <sub>CF</sub>	SCL/SDA fall time	-	120	-	150e6/f <sub>SCL</sub> <sup>[3]</sup>	ns
t <sub>SU:DAT</sub>	t <sub>SU:DAT</sub>	MSDA Data in setup time	50	-	4	-	ns
t <sub>HD:DAT</sub>	t <sub>HD:DAT</sub>	MSDA Data in Hold Time	0.5	350	3	-	ns
t <sub>SU:STA</sub>	t <sub>SU:STA</sub>	Repeated Start condition setup time	260	-	12	-	ns
t <sub>HD:STA</sub>	t <sub>HD:STA</sub>	Start condition hold time	260	-	30	-	ns
t <sub>SU:STO</sub>	t <sub>SU:STO</sub>	Stop condition setup time	260	-	12	-	ns
t <sub>BUF</sub>	t <sub>BUF</sub>	Time between Stop Condition and next Start Condition	500	-	500	-	ns
t <sub>INIT</sub>	-	Time from power on to first command	10	-	10	-	ms
t <sub>PD</sub>	-	Push/Pull Re-driving propagation delay between Master Side Port and Slave Side Port <sup>[4]</sup>	-	-	-	4	ns
t <sub>AVAIL</sub>	t <sub>AVAIL</sub>	Bus Available time (no edges seen on MSDA and MSCL)	-	-	1	-	us
t <sub>IDLE</sub>	t <sub>IDLE</sub>	Bus Idle time (no edges seen on MSDA and MSCL)	200	-	-	-	us
t <sub>IBI_ISSUE</sub>		Time to issue IBI after an event is detected when the Bus is available	-	-	-	15	us
t <sub>DOUT</sub>	t <sub>SCO</sub>	MSCL Falling Clock in to MSDA Valid Data Out Time	N/A	N/A	0.5	12	ns
t <sub>DOFFS</sub>	t <sub>SCO</sub>	MSCL Rising Clock in to MSDA Out- put Off	N/A	N/A	0.5	12	ns
t <sub>DOFFM</sub>		MSCL Rising Clock in to Master MSDA Output Off <sup>[5]</sup>	N/A	N/A	0.5	12	ns
t <sub>CL_r_DAT_f</sub>		MSCL Rising Clock in to Master Driving MSDA Signal Low	N/A	N/A	20	-	ns
t <sub>CCC_Delay</sub>		Any CCC followed by RSTDA CCC delay	N/A	N/A	2.5	-	ms

Note:

- (1). Timeout Setting is defined in REG#122.
- (2). Capped at 60ns
- (3). Capped at 60ns
- (4). Output port with 5pF total load; Input signal slew rate (30%-70%VIO\_inport) <=2ns, delay is 50% to 50% from Input to Output
- (5). Requirement of the master.

## 8.5 Timing Parameters – Slave Side Port Bus Agent

**Table8.Timing Parameters – Slave Side Port Bus Agent**

Symbol	Parameter	Min	Max	Unit
f <sub>SCL</sub>	Clock frequency	0.01	1	MHz
t <sub>HIGH</sub>	Clock pulse width high time	260	-	ns
t <sub>LOW</sub>	Clock pulse width low time	500	-	ns
t <sub>TIMEOUT</sub>	Detect clock low timeout	25	35	ms
t <sub>R</sub>	SCL/SDA rise time	-	120	ns
t <sub>F</sub>	SCL/SDA fall time	-	120	ns
t <sub>SU:DAT</sub>	SDA Data in setup time	50	-	ns
t <sub>HD:DAT</sub>	SDA Data In Hold Time	0.5	350	ns
t <sub>SU:STA</sub>	Start condition setup time	260	-	ns
t <sub>HD:STA</sub>	Start condition hold time	260	-	ns
t <sub>SU:STO</sub>	Stop condition setup time	260	-	ns
t <sub>BUF</sub>	Time between Stop Condition and next Start Condition	500	-	ns

## 8.6 Timing Parameters – Device

**Table9.Timing Parameters – Device**

Symbol	Parameter	Min	Max	Unit
t <sub>MSEL_DET</sub>	MSEL pin reset pulse detection window time	3	4	ms
t <sub>MSEL_TR</sub>	MSEL pin logic transition (slew) time, between 0.3V-0.75V MSEL pin transition has to be faster than t <sub>MSEL_TR</sub> for pin reset detection	-	50	ns
t <sub>RESET</sub>	Device Reset Time	2	-	ms
t <sub>IOCal</sub>	IO Drive Strength Calibration Time	-	2	ms

## 8.7 Timing Parameters – Hub Network

**Table10.Timing Parameters – Hub Network**

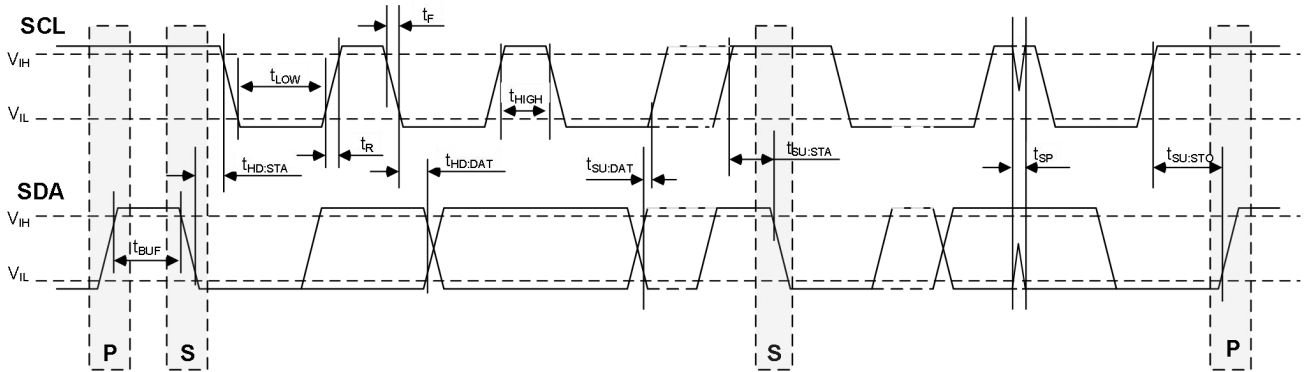
Symbol	Parameter	Min	Max	Unit
t <sub>PD</sub>	Push/Pull Re-driving propagation delay between Master Side Port and Slave Side Port <sup>[1]</sup>	-	-4	ns
t <sub>OR</sub>	Slave Side Port Push/Pull mode Re-driver output Rising time, 30% VIO to 70% VIO; Master Side Input Slew Rate (30% VIOM - 70% VIOM) =2ns		1.5	ns
t <sub>OF</sub>	Slave Side Port Push/Pull mode Re-driver output Rising time, 30% VIO to 70% VIO; Master Side Input Slew Rate (30% VIOM - 70% VIOM) =2ns		1.5	ns

Note:

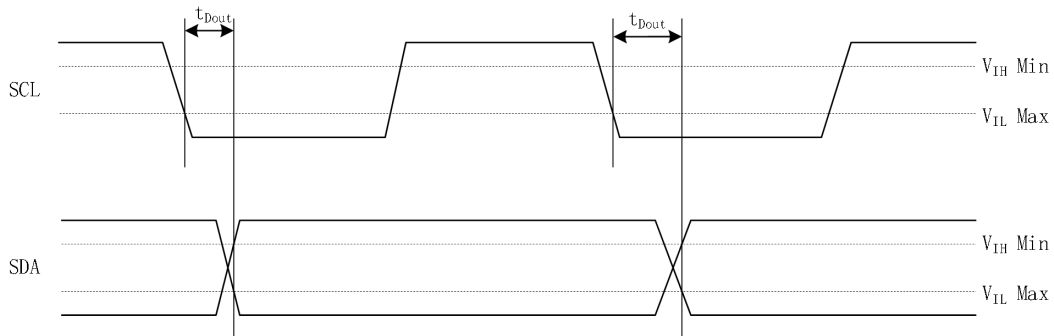
- (1). Output port directly drive the mid-point of two series 100ohm resistors connected between VIO\_outport and GND; Input signal slew rate (30%-70%VIO\_inport) <=2ns, delay is 50% to 50% from Input to Output

## 8.8 Timing Waveforms and Timing

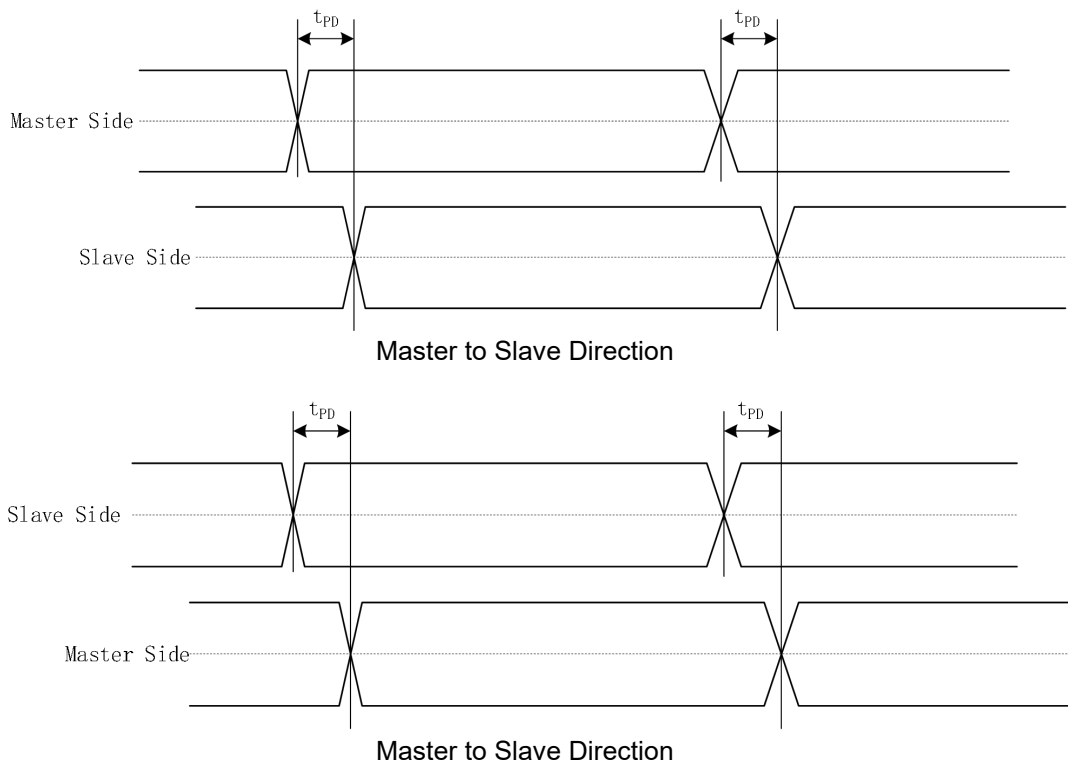
### 8.8.1 I3C/I2C Bus Waveform



### 8.8.2 I3C Read Timing

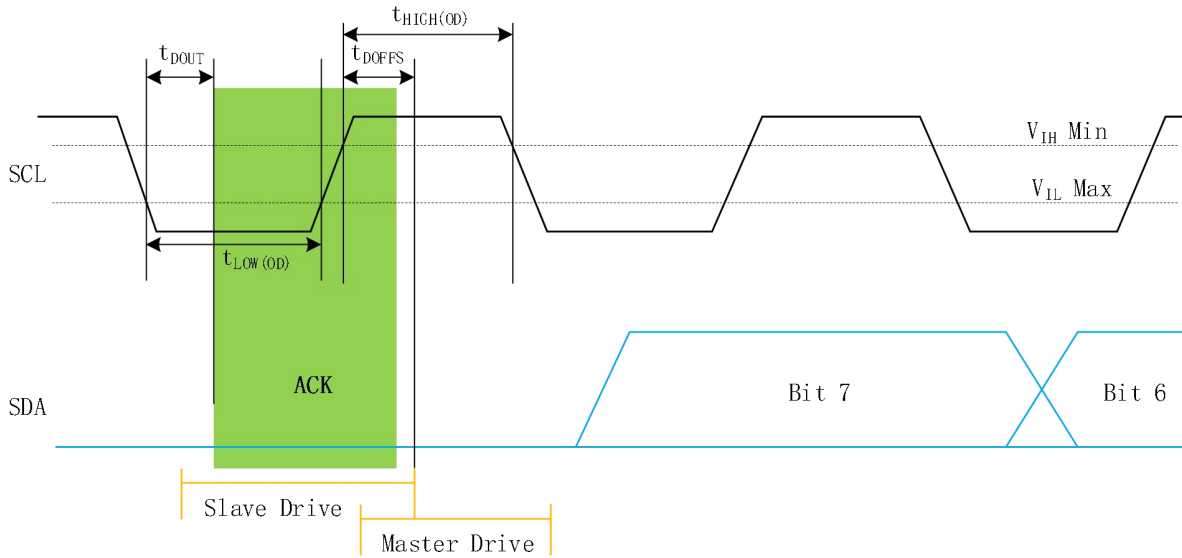


### 8.8.3 Hub Push/Pull Re-driving Mode Propagation Delay



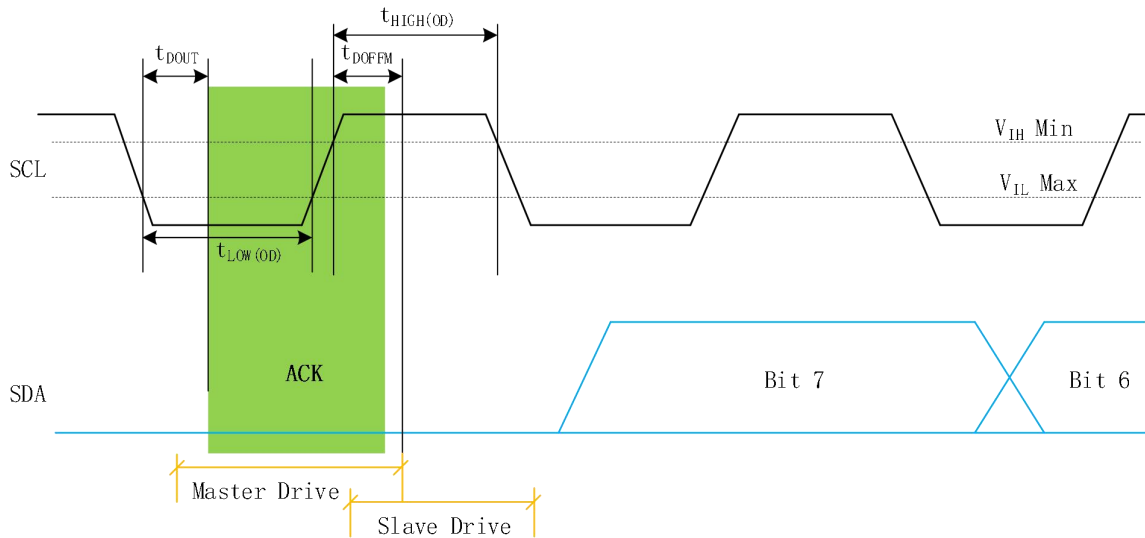
### 8.8.4 Hand-off Timing Diagram

#### 8.8.4.1 Slave to Master



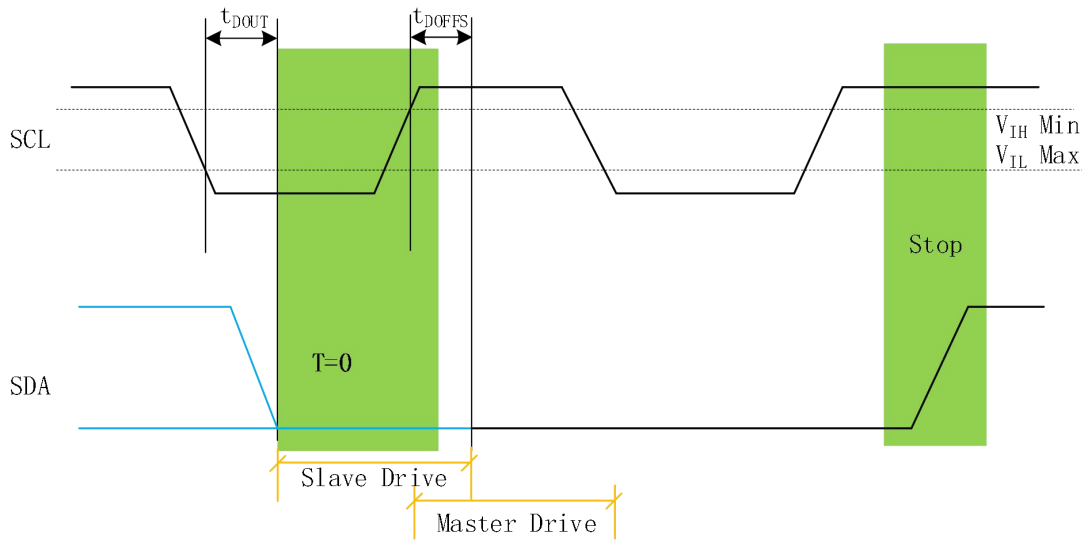
Slave to Master handoff (Write)

#### 8.8.4.2 Master to Slave



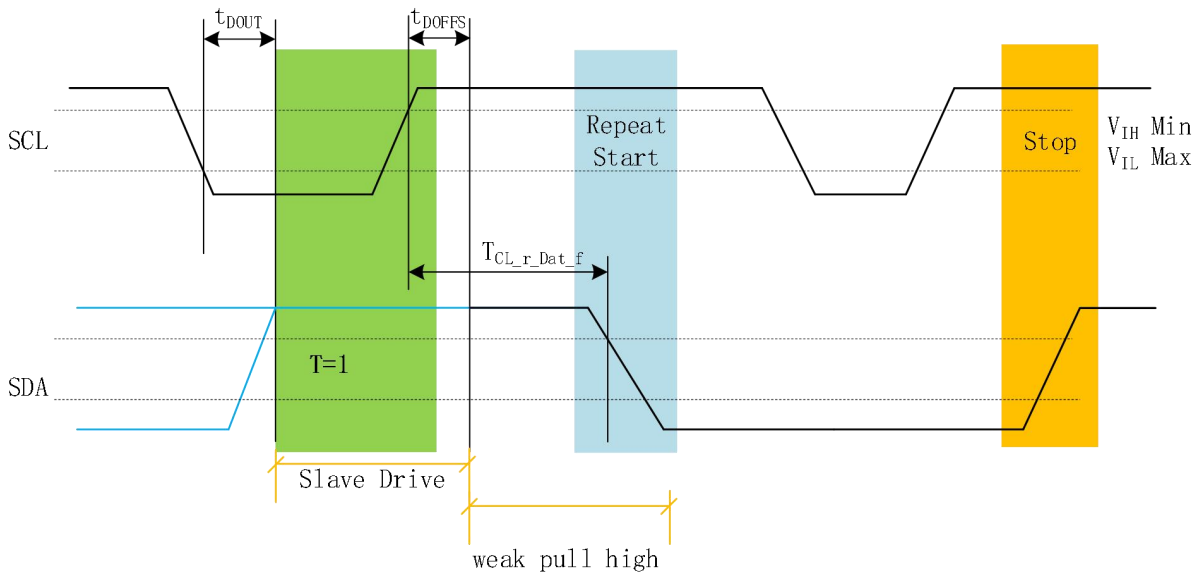
Master to Slave handoff (IBI)

**8.8.4.3 T-bit T = 0 Read Termination**



T=0 Slave to Master handoff (Tbit)

**8.8.4.4 T-bit T=1 Master Repeat-Start Read Termination**



T=1 Slave Master Terminates with Sr (Tbit)

## 9 Function Description

The PJ71MXX is comprised of the following independently operating functions:

- Two Master-interfacing Ports (or Master Side Ports, M0, M1)
- On-Chip I2C/I3C Slave Interface attached to the master-interfacing port (the Slave)
- A Master Side Port multiplexer (The MUX)
- A Hub Network that connects the selected Master Side Port and all Slave Side Ports (The Hub Network)
- Up to eight Slave-interfacing Ports (S0 - S7)
- SMBus Agent attached to the Slave-interfacing Ports (or Slave Side Ports)
- On-Chip registers that are accessed via the on-chip slave interface (The Registers)

### 9.1 Definitions

**I3C compatible I2C devices** – I2C devices that are accessed with I2C Open Drain protocol only, do not stretch SCL, and tolerate I3C high-speed transactions / CCCs on the same bus. These devices co-exist with the I3C slave devices in the bus referenced as “Mixed Fast Bus” as defined in I3C specification.

- **Slow I2C devices** – I2C devices that are accessed with I2C Open Drain protocol only, do not stretch SCL, do not tolerate I3C high-speed transactions/ CCCs on the same bus. If these devices exist with I3C slave devices on the same bus, the bus will be referenced as “Mixed Slow/Limited Bus” as defined in the I3C specification.
- **Clock Stretching I2C Device** – Refers to I2C Slave Device that may stretch the SCL during operation.
- **SMBus Device** – Refers to SMBus Master / Slave Device that may stretch the SCL during operation.
- **Master Side Port** – PJ71MXX 2-wire interface Port that interfaces with an I2C/I3C Master. It is also referenced as “Upstream Port” or “Controller Side Port”
- **Slave Side Port** – PJ71MXX 2-wire interface Port that interfaces downstream with I2C, I3C, or SMBus devices. It is also referenced as “Device Side Port”
- **SMBus Agent** – A function block in PJ71MXX that handles the SMBus Master or Slave Function, attached to a Slave Side Port
- **Transparent Bridging** – A hub operating mode that bridges master side and slave side ports in a transparent manner – i.e. device only introduces delay and level shifting. The network operates as if the PJ71MXX does not exist, or is transparent.
- **Agenting** – The non-transparent bridging mode of the PJ71MXX device. In this mode, a bus Agent receives transaction descriptors and transaction data, and then handles the device transactions under the slave side port.
- **Selected Master** – The I3C master selected to control the I3C port network linked by transparent bridging mode. Only one I3C master is selected to control the I3C transparent network.
- **Disconnect and disconnected Slave Side Port** – Slave Side Port is connected or disconnected from the Hub Network. Disconnected Slave Side Port is unable to communicate with other ports via the transparent bridging network. The Bus Agent can/may work with the port. When a Bus Agent is working, the port is disconnected from the transparent Hub network.
- **OD Only Mode** – A mode in the IO circuit. In this mode, the IO tolerates 3.3V levels and supports Open Drain mode.
- **OD/PP Compatible Mode** – (I3C compatible mode.) A mode in the IO circuit. In this mode, the IO handles Open Drain or Push/Pull transactions and switches between those two modes as per I3C/I2C protocol. The electrical level is defined by the VIO voltage – VIO\_M for master interfacing ports and VIO\_S for slave interfacing ports. VIO\_M and VIO\_S are the push/pull mode output supply for pull high drivers.

- **Network Partition** – In PJ71MXX, any Slave Side Port can be disconnected from the Hub Network. This supports an I3C/I2C network that is partitioned for another purpose. Examples are the ability to:
  - Expand the address space to support additional devices.
  - Isolate a low-speed I2C device from I3C devices.

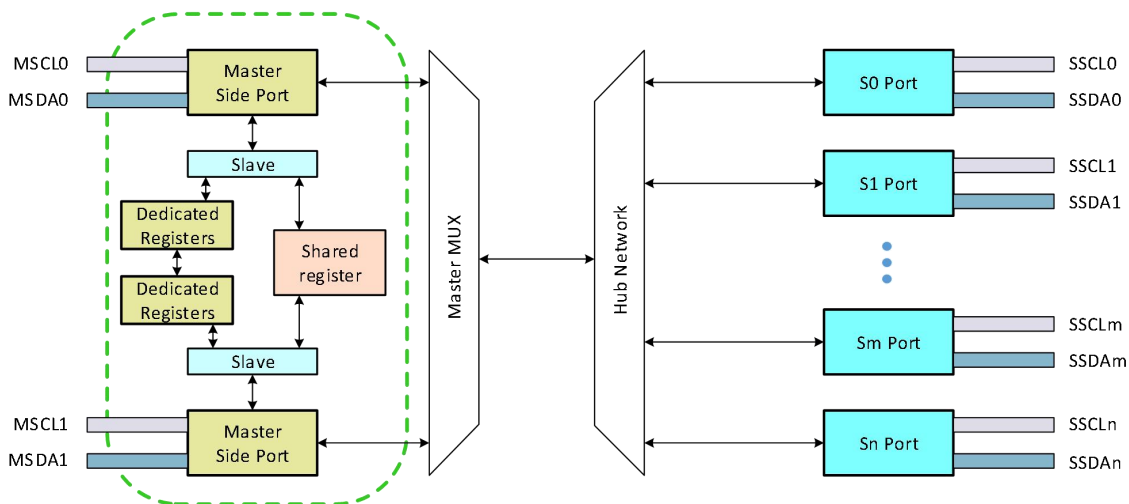
## 9.2 Master Side Ports

### 9.2.1 Overview

The I3C Hub Device features two master interfacing ports: Master Side Port 0 (**M0**) and Master Side Port 1 (**M1**). Each Master Side Port is associated with a dedicated I2C/I3C On-Chip Slave Interface. Each Slave interface has direct access to a 256-byte register space.

Master Side Ports are associated with dedicated (per port) registers and shared, device-level registers that map to the I2C/I3C Slave register space.

Two Master Side Ports connect to a Master Side Port Multiplexer, which allows a selected Master Side Port to connect to the Slave Side Port Hub Network. The Hub Network connects to up to eight Slave Interfacing Ports.



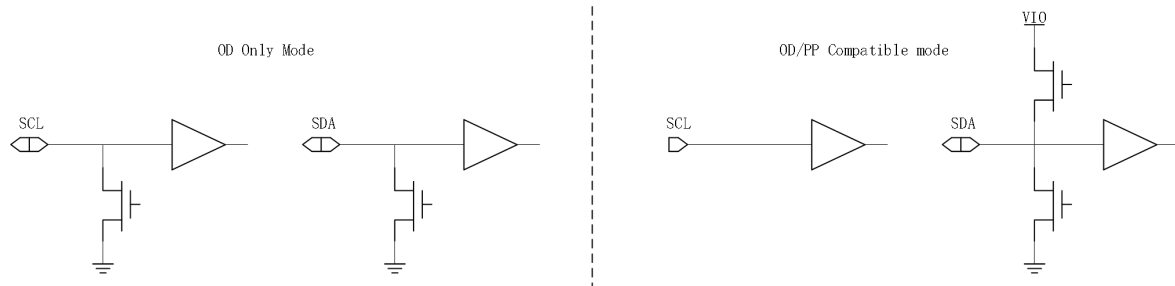
**Figure 5. Master Side Port**

### 9.2.2 IO Operating Mode and Signaling Level

Master Interfacing Ports operate in the following modes:

- **Open-Drain Only mode.** In this mode, only the I2C protocol is supported. The voltage level of the MSCLx/MSDAx signaling is up to 3.3V.
  - Both SCL and SDA work in Open-Drain Mode
- **Open-Drain (OD) /Push-Pull (PP) compatible Mode.** In this mode, the Protocol is I2C or I3C, the IO operation and signaling is determined by the I2C/I3C protocol. It is either Open Drain as in SMBus operation, or dynamically switching between OD and PP mode as in I3C Operation. The signaling voltage level is defined by the VIOM pins. The VIOMx pins are directly supplied by the system/platform, or generated by on-chip regulators. The VIOMx on-chip regulators are configured as 1.0, 1.1, 1.2 or 1.8V, by programming the VIOM0\_LDO\_Voltage (VIOM) registers.
  - The SCL is input only. The SDA is input/output compatible and Open-Drain / Push-Pull compatible for signaling VIO voltage.

Master Interfacing Ports do not have on-chip pull-up resistors.



**Figure 6. Master IO Signaling Mode**

### 9.2.2.1 Operation Mode Transition

The MPort\_OD\_Only bit defines the IO Operating mode. Users program the bit to “0” to transition to OD/PP compatible mode when the master device is I3C master. If REG#17[3], “MPort\_OD\_Manual” bit is not set, IO Operating mode is updated automatically when I3C operation is detected when the IO operation mode is still configured in OD\_Only mode, and the REG#17[4] “MPort\_OD\_Only” bit will be cleared at the same time.

- When an PJ71MXX’s on-chip slave received a 7Eh address in any transaction on its associated master interfacing port, it assumes the transaction is an I3C transaction. The IO operating mode switches to OD/PP compatible mode.

Example scenarios of IO operating mode update:

1. The PJ71MXX’s M0 port powers up with 3.3V tolerant I2C operation. The VIOM0 rail is supplied with 1.0V by the system. After power-up, the master attached to the port operates at a 1.0V signaling and starts to send CCC to the PJ71MXX. Upon receipt of the 7E address in CCC, PJ71MXX responds to the CCC and changes the operating voltage to 1.0V on the fly. The push/pull operation in the CCC operates at VIOM voltage of 1.0V.
2. The PJ71MXX’s M1 port powers up with 3.3V tolerant I2C operation. M1 port’s IO supply VIOM1 is supplied by an on-chip regulator. After power-up, the master attached to this port operates at 1.8V signaling. The master initializes the PJ71MXX device by setting the VIOM1 LDO voltage to 1.8V, enabling the LDO, and clears the M0 port OD\_Only configuration register to set the operating voltage to 1.8V. Push/pull operation, in the following CCC operates at the configured VIOM voltage of 1.8V.

### 9.2.2.2 SMBUS Master Support

When Master Side Ports are in OD Only mode, they support SMBUS master and SCL clock stretching. That means that both the MSCL / SSCL and MSDA / SSDA are connected via the Hub Network in an Open-Drain manner.

- If the Hub Network connects to a Slave Side Port that is in Open Drain Only Mode, the SCL stretching can be propagated from the Slave Side Port to the Master Side Port. See I2C/SMBus Open-Drain Only Operation for further details.

### 9.2.3 On-Chip Slaves

Each of the two Master Interfacing Ports is attached with a dedicated on-chip I2C/I3C slave interface (the slave). The two on-chip slaves are always active and accessible by the associated master after power-up, regardless of which master is selected and is connected to the Hub Network.

#### 9.2.3.1 Slave Addressing

The two on-chip slaves power up with a default I2C Slave static address:

- 1110\_000

The default I2C static slave address is updated by programming the I2C Reconfigured Slave Address Register SLAVE\_ADDR. User should only change this register before an I3C dynamic address is assigned.

- If the user programs the Reconfigured Slave Address Register via the on-chip slave interface, the subsequent transactions need to use the updated slave address.

## Dynamic Address Assignment

After power-up, the on-chip slave stays in I2C/I3C compatible mode. The I3C Hub device is accessed via the I2C static address. I3C dynamic address can be obtained via the following ways:

- SETAASA CCC – Upon receipt, the Slave dynamic address is assigned with the same value as the static address.
- SETDASA CCC – Upon receipt, the Slave dynamic address is assigned with the specified value in the SETDASA CCC.
- ENTDAACCC procedure – Upon receipt, the Slave dynamic address is assigned by the Master Side Port.
  - To support the ENTDAACCC procedure, the PJ71MXX has a 48bit PID that is initialized in the production test.

## Dynamic Address Update

The dynamic address can be changed with SETNEWDA CCC – Upon receipt, the slave's dynamic address is assigned with a designated value.

The assigned dynamic address can be removed by RSTDAA CCC.

## I2C/I3C Framing and Transaction Format

The on-chip slave only receives and responds to I2C framing transactions with its slave static address.

When a dynamic address is assigned, the on-chip slave assumes all transactions that matches the dynamic address as I3C format transactions. The on-chip slave does not respond to the original static address, if a dynamic address is assigned.

The PJ71MXX does not respond to any direct CCC until a dynamic address is assigned. Parity Checking and Packet Error Checking (PEC) are only valid when a dynamic address is assigned, as referenced as "I3C Mode" in the JEDEC SPD Bus definition.

### 9.2.3.2 Bus Reset

The I3C Hub Device supports the Bus Reset feature in the JEDEC SPD Bus operation.

If enabled (see Bus\_Reset\_Enable), the I3C Hub Device supports the SCL-low timer-based bus reset procedure. When a SCL-low time-based time reset is received (see REG#122), the interface removes the dynamic address assigned to it and clear the PEC/Parity settings. This is primarily for the JEDEC SPD Bus context application.

### 9.2.3.3 On-Chip Slave Accessible Device Register Space

#### 9.2.3.4 Read/Write Frame Format

The PJ71MXX follows a standard I2C / I3C transaction format. The Read/Write transaction frame below defines the register access scheme.

Each on-chip I2C/I3C interface maintains a 256-byte register space. The 256 registers are indexed by an address pointer register. The first half, i.e. the 0-127 Address Space is directly mapped, and the second half, i.e. 128-255 Address Space is paged. Before accessing to paged registers, the page pointer register needs to be set to the correct page number.

The first byte of the write transaction sets the register space address pointer to that byte's value. The second and following bytes are written to the location that the register address pointer points within the 128-byte space. The address pointer automatically increments during the Read Transaction after each register is read or during Write Transaction after a byte of data is written. When the address pointer hits the 128-boundary, it automatically wraps back to zero (if it is 127), or 128 (if it is 255).

The address pointer is explicitly updated by each write transaction.

In the I3C Operation, the burst length register can be enabled. The burst length register sets the number of payload bytes to be transferred in Private Read or Write transactions. If the BL value =7'h00, it represents a burst length of 128. The "BL R/Wn" bit indicates the BL field meaning for the current Write Transaction (R/Wn=0), or the following

Read Transaction (R/Wn=1). For an I3C Read transaction, the PJ71MXX device sets the T bit to zero after # of BL bytes has been readout. For an I3C Write transaction, the PJ71MXX ignores the following bytes after # of BL bytes has been received. The master can write or read with the number of bytes that less than the burst length. In the Write transaction, the PJ71MXX device writes each received byte immediately and does not buffer it.

In the JEDEC SPD bus operation, a PEC function (see Packet Error Checking Feature) can be enabled. PEC enabled framing has a burst length and a CRC byte defined for the payload bytes in the Read or Write transactions. The “BL R/Wn” bit indicates the BL field meaning for the current Write Transaction (R/Wn=0), or the following Read Transaction (R/Wn=1). In the case of “BL R/Wn” =1, the CRC byte of the current write transaction immediately follows the BL byte.

PEC enabled write-transaction data may be buffered, and only written to the device if PEC checking passes for the whole packet. The PJ71MXX device keeps a 4-byte cache buffer for write transactions in PEC enabled mode. The write transaction data is stored in the buffer, checked for PEC and then “Written Back” to the target register fields. The exception is the page mapped transaction buffers to the bus agent, and the scratch board buffers. Those paged buffers are always in “write through” mode, i.e. data is passing through the cache buffer before PEC checking is done. The Burst Length of transfers can be up to 128 bytes. For the directly mapped registers, the maximum burst length supported for “Write Back” protection is 4. The write transaction that has a burst length greater than 4 is written in a write through mode. The PEC is checked throughout the whole burst. User should avoid using write through mode for registers that trigger a hardware action or data change.

#### 9.2.3.4.1 I2C and I3C without Burst Length Framing

**Table11. Write Transaction Framing without Burst Length**

Start	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A/N/T	Stop	Note
S or Sr	SlvAddr 6	SlvAddr 5	SlvAddr 4	SlvAddr 3	SlvAddr 2	SlvAddr 1	SlvAddr 0	W=0	A		
	Address Pointer Register Value [7:0]								A/T		
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	A/T		Address Point +0
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	A/T		Address Point +1
	.	.	.	.	.	.	.	.	.		...
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	A/T	P or Sr [1]	Address Point +n

(1). Sr = Repeat Start, P = Stop

**Table12. Read Transaction Framing without Burst Length**

Start	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A/N/T	Stop	Note
S or Sr	SlvAddr 6	SlvAddr 5	SlvAddr 4	SlvAddr 3	SlvAddr 2	SlvAddr 1	SlvAddr 0	W=0	A		
	Address Pointer Register Value [7:0]								A/T		
Sr	SlvAddr 6	SlvAddr 5	SlvAddr 4	SlvAddr 3	SlvAddr 2	SlvAddr 1	SlvAddr 0	R=1	A		
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	A/T		Address Point +0
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	A/T		Address Point +1
	.	.	.	.	.	.	.	.	.		...
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	A/T	P or Sr [1]	Address Point +n

(1). Sr = Repeat Start, P = Stop

#### 9.2.3.4.2 with Burst Length Framing

In I3C Mode, if enabled (see BL\_Enable) a burst length register is programmed in a write transaction after the Address Pointer Register to allow the length of the burst for the specified Read and Write transactions.

**Table13. Write Transaction Framing with Burst Length**

Start	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A/N/T	Stop	Note
S or Sr	SlvAddr 6	SlvAddr 5	SlvAddr 4	SlvAddr 3	SlvAddr 2	SlvAddr 1	SlvAddr 0	W=0	A		
	Address Pointer Register Value [7:0]								T=Parity		
	BL R/Wn= 0	Burst Length							T=Parity		
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	T=Parity		Address Point +0
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	T=Parity		Address Point +1
	.	.	.	.	.	.	.	.	.		...
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	T=Parity	P or Sr	Address Point +BL-1

Read Transaction:

**Table14. Read Transaction Framing with Burst Length**

Start	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A/N/T	Stop	Note
S or Sr	SlvAddr 6	SlvAddr 5	SlvAddr 4	SlvAddr 3	SlvAddr 2	SlvAddr 1	SlvAddr 0	W=0	A		
	Address Pointer Register Value [7:0]								T=Parity		
	BL R/Wn= 1	Burst Length							T=Parity		
Sr	SlvAddr 6	SlvAddr 5	SlvAddr 4	SlvAddr 3	SlvAddr 2	SlvAddr 1	SlvAddr 0	R=1	A		
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	T=1		Address Point +0
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	T=1		Address Point +1
	.	.	.	.	.	.	.	.	.		...
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	T=0	P or Sr	Address Point +BL-1

#### 9.2.3.4.3 I3C with PEC Framing

I3C with PEC framing is the JEDEC SPD5 Bus defined feature (see Packet Error Checking Feature). Burst Length can be any number from 1 to 4 in the I3C with PEC framing if PEC Write Back operation is desired.

When PEC is enabled using PEC\_Fun\_Enable, the transaction framing is as follows:

**Table15. Write Transaction framing with PEC**

Start	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A/N/T	Stop	Note
S or Sr	SlvAddr 6	SlvAddr 5	SlvAddr 4	SlvAddr 3	SlvAddr 2	SlvAddr 1	SlvAddr 0	W=0	A		
	Address Pointer Register Value [7:0]								T=Parity		
	BL R/Wn= 0	Burst Length (BL)							T=Parity		
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	T=Parity		Address Point +0
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	T=Parity		Address Point +1
	.	.	.	.	.	.	.	.	.		...
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	T=Parity		Address Point +BL-1
	PEC Byte								T=Parity	P or Sr	

**Table16. Read Transaction framing with PEC**

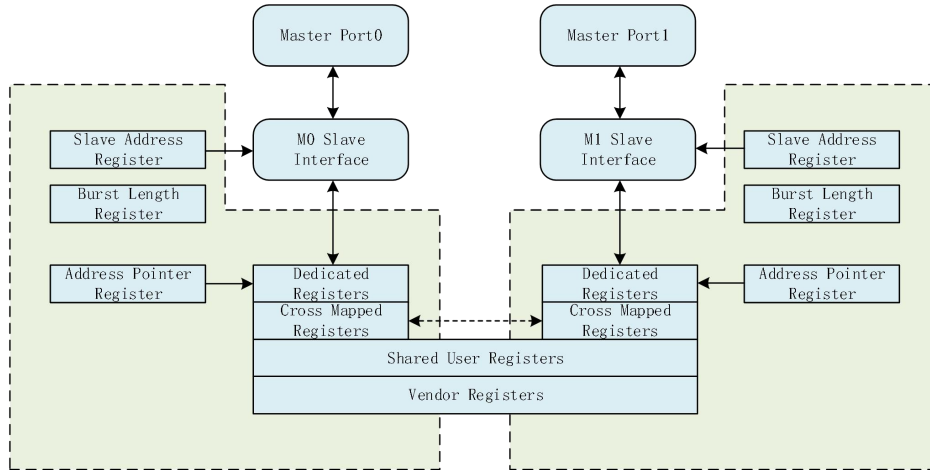
Start	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A/N/T	Stop	Note
S or Sr	SlvAddr 6	SlvAddr 5	SlvAddr 4	SlvAddr 3	SlvAddr 2	SlvAddr 1	SlvAddr 0	W=0	A		
	Address Pointer Register Value [7:0]								T=Parity		
	BL R/Wn= 1	Burst Length							T=Parity		
	PEC Byte								T=Parity		
Sr	SlvAddr 6	SlvAddr 5	SlvAddr 4	SlvAddr 3	SlvAddr 2	SlvAddr 1	SlvAddr 0	R=1	A		
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	T=1		Address Point +0
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	T=1		Address Point +1
	.	.	.	.	.	.	.	.	.		...
	DataBit 7	DataBit 6	DataBit 5	DataBit 4	DataBit 3	DataBit 2	DataBit 1	DataBit 0	T=1		Address Point +BL-1
	PEC Byte								T=0	P or Sr	

### 9.2.3.5 Register Space and Addressing

- On-chip registers map to the two on-chip slave interface's addresses in the following manners:
- There are registers in each on-chip slave interface's space that are **dedicated** to that interface (i.e. they are per-port, each of slave interfaces is a unique copy). For these registers, the register-name is the same in the two slave interfaces. However, each register is the private copy that is attached to that master's interfacing port.
- There are registers in each on-chip slave interface's space that are shared (i.e. they are per-device, one copy for both interfaces).

- For some registers, the write operation is only be allowed by one designated Master Side Port.
- There are per-port registers are cross-referenced. Registers are mapped to different register addresses on different on-chip slave interfaces. These registers are used for master-master communication.

The block diagram below shows two Master Side Ports, two master interfaces, two register-spaces, mapped registers, as well as register sharing / cross-referencing.



**Figure7.On-Chip Slave Address Space and Mapping**

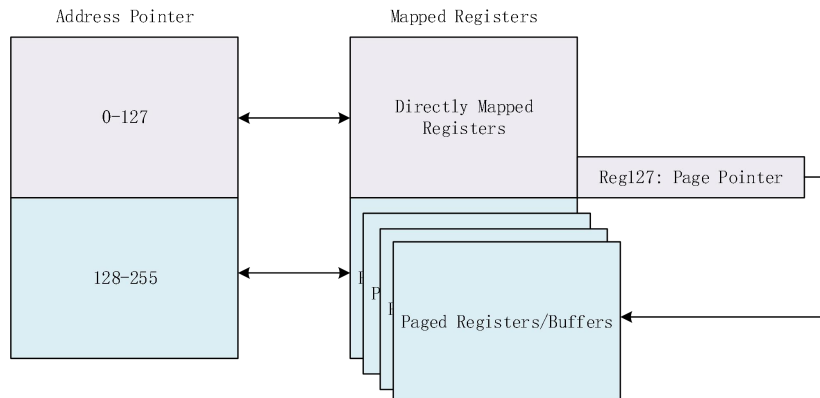
For a detailed map of registers, see Register Map.

**9.2.3.6 Paged Buffers and Registers**

Register address ranges from 0-127 are directly mapped. Directly mapped registers are not subject to the page pointer register control.

Register address ranges from 128 to 255 are paged. Reg#127 is the page pointer.

Users set the page pointer in order to access to the paged registers.



**Figure8.On-Chip Slave Address and Page Pointers**

**9.2.3.7 I3C IBI Support**

The two on-chip slaves support the I3C Basic 1.0 IBI feature. The I3C Slave generates IBI when the following criteria are met:

- A dynamic address is assigned
- IBI is enabled (with ENEC CCC)
- An enabled IBI event occurs

The Hub Network supports IBI arbitration from any connected Slave Side Port. The IBI address header in the Hub Network will be arbitrated with the selected master interfacing port slave, as well as any other devices that are hooked onto the master interfacing port.

#### 9.2.3.7.1 IBI Source

There are five interrupt sources: PEC Error, Parity Error, Slave Port Event, Other Master Message Request, and Bus Agent event. They are mapped to interrupt numbers 1,2,3,4,5 respectively when requested by GETSTATUS CCC. Please note that the PEC Error and Parity Error apply to a specific Master Interfacing Port only and have port-specific status.

#### 9.2.3.7.2 IBI event flag set and clear

The events, when occurred and enabled, are captured in associated **event flag** registers. Once an event is captured, an event flag register remains set, and is only cleared by writing a “1” to that bit.

- A “flag status” register aggregates the event flag registers for easier IBI source indexing and identification. The flag status register is a reference to the combination of a group of event flag registers, and is automatically cleared if all event flags are cleared.
- The event flag registers are cleared by a writing “1” to the specific event flag bit location. Usually, the flag is cleared when the event handling in the host device is finished.
- The event flags, and event flag status bits are gated by IBI Enabled registers for IBI participation. If one or more event flags is set and a IBI enable is set at the same time, the IBI pending status bit is set.
- The event flags are accessible by the master device even when IBI is disabled.
- When all flags are cleared, the IBI pending status is cleared as well. Flags are set/cleared regardless of whether IBI function is enabled and regardless the interface protocol is (I2C or I3C).

#### 9.2.3.7.3 Event Capture and IBI Generation Registers

When the interface operates in I3C mode, i.e. after the dynamic address gets assigned, and the IBI function enabled by ENEC CCC, the IBI\_Enabled status is set. When IBI is enabled, the IBI generation logic will check if an IBI is **pending**.

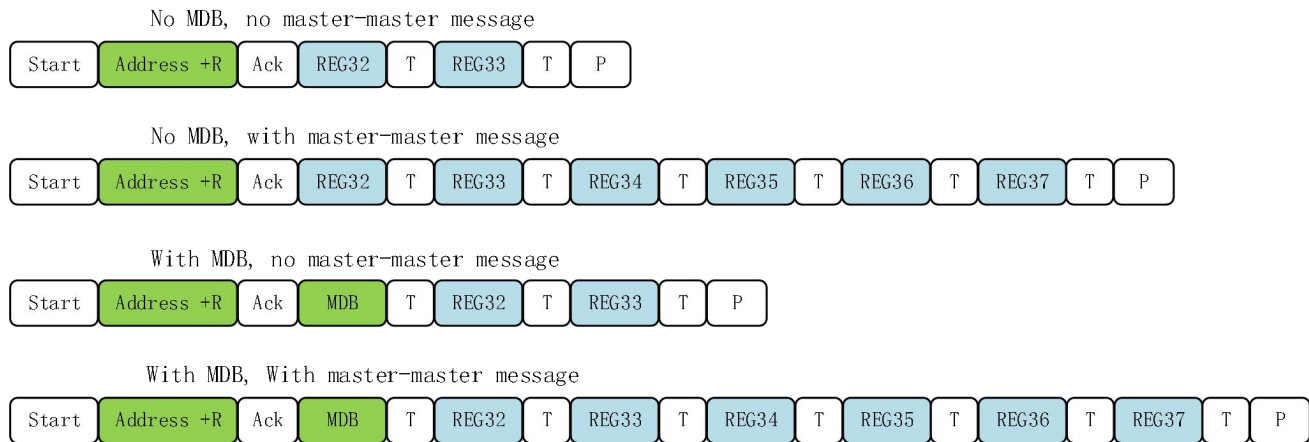
- “IBI pending” status is **set** by any newly set event flags, and **cleared** by a successful IBI transaction.
- If there is an IBI pending, the IBI logic participates in address arbitration after a new Start is detected. If there is no Start is detected while the I3C Bus Available(tAVAL) condition is detected, the IBI circuit will trigger a new Start condition and participate in the address arbitration.
- If the on-chip slave wins the IBI address arbitration, the IBI logic considers the IBI was successful. The IBI Pending Status will be cleared.
- Subsequent IBI generation only occur when the IBI pending status is set again, i.e. a new 0 to 1 positive edge transition is detected on an event flag bit. That means the IBI is regenerated when a new event type is captured in the event flag. Duplicated events, if already captured, will not trigger a new IBI generation.
- If the device loses IBI address arbitration, the IBI logic tries to re-participate in the address arbitration on the next new Start condition; or if Bus Idle is detected, re-generates the Start condition and participate in the address arbitration process. This action repeats until a successful IBI was generated and the internal “IBI Pending” flag is cleared.

#### 9.2.3.7.4 IBI Payload

PJ71MXX device on-chip slave sends the IBI data bytes of 2 or 6 bytes – REG#32 to REG#37 after IBI arbitration wins.

- The first 2 bytes are the interrupt sources and are sent out first.
- If the Message Request Flag is set in REG#32 then the four message bytes will be sent out if the master did not interrupt the payload bytes in their T-bits.

- If the customized MDB byte is enabled (see IBI\_MDB\_Enable) a programmable MDB byte will be inserted after the address byte, before the payload bytes.
- Below are the four possible IBI frame formats, if the master elects to allow the PJ71MXX to transmit all data bytes.



**Figure9.IBI Transaction Framing**

### 9.2.3.8 Hot Join Support

The PJ71MXX Device Master Interfacing Port supports Hot Join. If Hot Join Enable bit (see HJ\_Enabled) is set, a dynamic address is not assigned, a Hot Join request (see Other\_Master\_HJ\_Req) from other Master Side Port is received, and after a Bus Idle condition(tIDLE) is detected.

The I3C On-Chip Slave on the Master Side Port will:

- Launch Hot Join procedure and arbitrate with address “0000\_010” as defined in I3C Basic specification, similar to the IBI procedure
- Use “W” instead of “R” in the R/W bit

The Hot Join bit is by default “Enabled”. It can be disabled if a DISHJ bit is received in DISEC CCC (JEDEC and I3C Basic 1.0).

If the master NAKed the Hot Join Request, it is retried after another tIDLE after a bus idle condition, unless the Hot Join Enable bit is cleared.

### 9.2.3.9 Bus Reset

Refer to Bus Reset.

### 9.2.3.10 Bus Context and CCC Support

An PJ71MXX Hub device supports a superset of CCCs, which includes the mandatory CCCs as defined in I3C Basic 1.0 and JEDEC SPD/Module Management Bus Specifications.

The supported CCCs of the on-chip I3C slave are defined in Bus Context.

An PJ71MXX Hub device NAKs unsupported CCC after its assigned dynamic address, and then wait for STOP or repeated start.

### 9.2.3.11 Error Handling

The PJ71MXX device supports I3C Basic 1.0 errors S0 to S5 detection and Recovery.

The PJ71MXX device does not support HDR. When an Error occurs (S0/S1), it does not wait for HDR Exit since it may not arrive. It reports S0 only for 7’h7E+R. Below is the MIPI Error detection and Recovery flow. S0/S1 handling in the PJ71MXX device is different from MIPI.

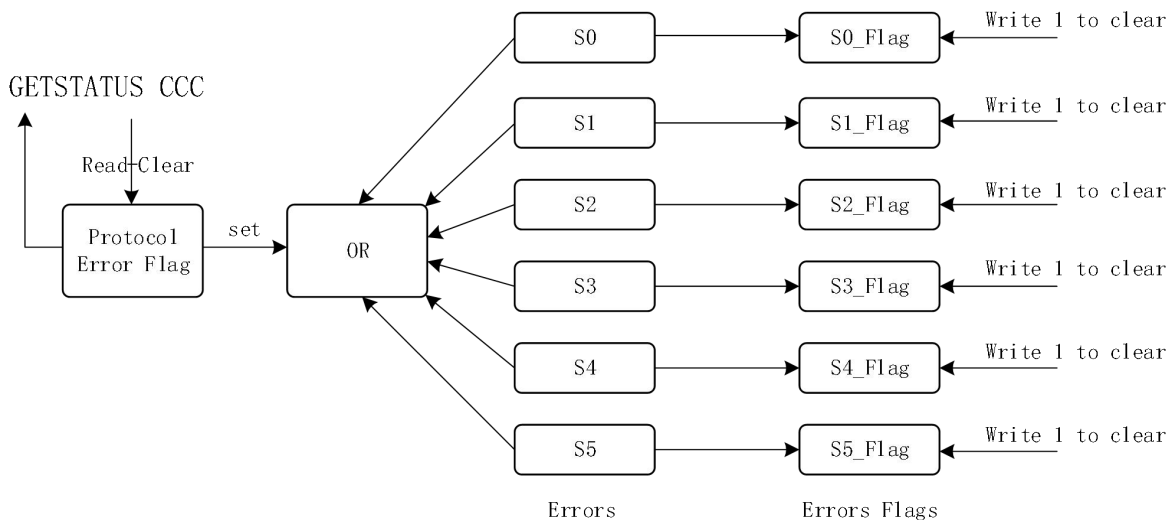
Error Type	Description	Error Detection Method	Error Recovery Method
S0	Invalid Broadcast Address/W (=7'h7E/W) or Dynamic Address/RW after DA assignment	Detect any of the following: 7'h7E / R	Ignore the following transaction until Repeat Start or Stop
S1	CCC Code	Parity Check, using T-bit	Ignore the transaction until Repeat Start or Stop
S2	Write Data	Parity Check, using T-bit	Enable STOP or Repeated START detector and neglect other patterns.
S3	Assigned Address during Dynamic Address Arbitration	Parity Check, using PAR Bit	Generate NACK (after PAR), then wait for another Repeated START and 7E/R to re transmit the Provisional ID.
S4	7'h7E/R missing after Sr during Dynamic Address Arbitration	Detect 7'h7E/R missing after Sr during Dynamic Address Arbitration	Generate NACK (after 7'h7E/R), then enable STOP or Repeated START Detector and ignore all other patterns
S5	Transaction after detecting CCC	Unsupported direct CCC received, or illegally formatted CCC received. Detect illegally formatted CCC: <ol style="list-style-type: none"> <li>Direct Write CCC with a Read in the R/W bit in the address byte: ENEC, DISEC, RSTDAA, DEVCTRL, SETNEWDA</li> <li>Direct Read CCC with a Write in the R/W bit in the address byte: GETBCR, GETDCR, GETPID</li> </ol>	Generate NACK (after Slave Address), then enable STOP or Repeated START Detector and ignore all other patterns

In order to be JEDEC SPD/Module Management Bus compliant, the PJ71MXX detects PEC errors.

Both Parity and PEC errors are logged in registers and reported to the master via IBI and GETSTATUS CCC.

Protocol Errors are logged into Protocol Error Log Registers (see REG#123).

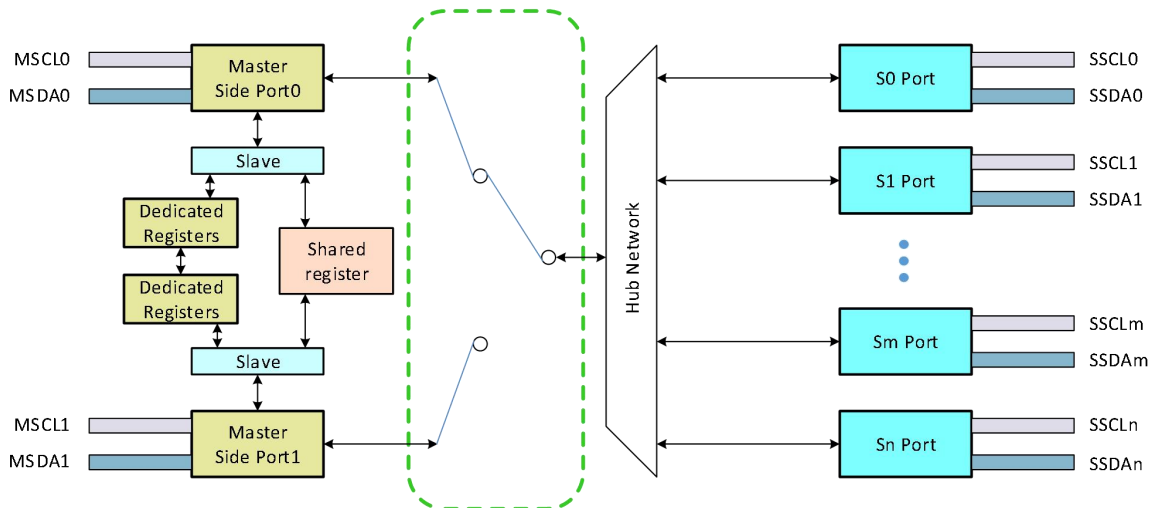
Protocol Errors are captured in a read-and-clear protocol error register. This register is reported to GETSTATUS CCC (JEDEC and I3C Basic 1.0).



**Figure10. On-Chip I3C Slave Error Reporting and Logging**

### 9.3 Master Multiplexer

The Master Multiplexer selects one of the two Master Side Ports to the Hub network.



**Figure 11. Master Multiplexer**

### 9.3.1 Multiplexer Selection

The master multiplexer selection is determined by the following order:

- If the Master Selection Pin (MSEL) was actively driven to logic high or logic low, the master interfacing ports are selected by the MSEL pin.
- If the MSEL pin is not driven (high-Z, or HiZ), the master interfacing ports are selected by Master Selection logic, which is determined by the ReqHubCtrl (see Master Multiplexer Selection Control (M0/M1 Dedicated - Per Port) bits in the master control register).

#### 9.3.1.1 Pin Selection

MSEL pin is internally biased to HiZ, but can be driven high or low to select the Master Side Port that connects to the Slave Side Port Hub Network.

MESL Pin	Master Side Port Selected
Logic Low (0)	Master-Port 0
Logic High (1)	Master-Port 1
HiZ	Register Selection

#### 9.3.1.2 Register Selection

When configured by the Multiplexer Control register (see Master Multiplexer Selection Control (M0/M1 Dedicated - Per Port) the M1ReqHubCtrl and M0ReqHubCtrl fields will be used to select the Master Side Port that connecting to the Slave Side Port Hub Network.

M1ReqHubCtrl, M0ReqHubCtrl	Master Side Port Selected
01	Master-Port 0
10	Master-Port 1
11,00	The port has configured with higher priority (see Master_Port_Mux_Priority)

The currently selected master is read from the multiplexer master-selection status (see Master Multiplexer Selection Status (M0/M1 Dedicated - Per Port) register).

### 9.3.2 Multiplexer Port Switching

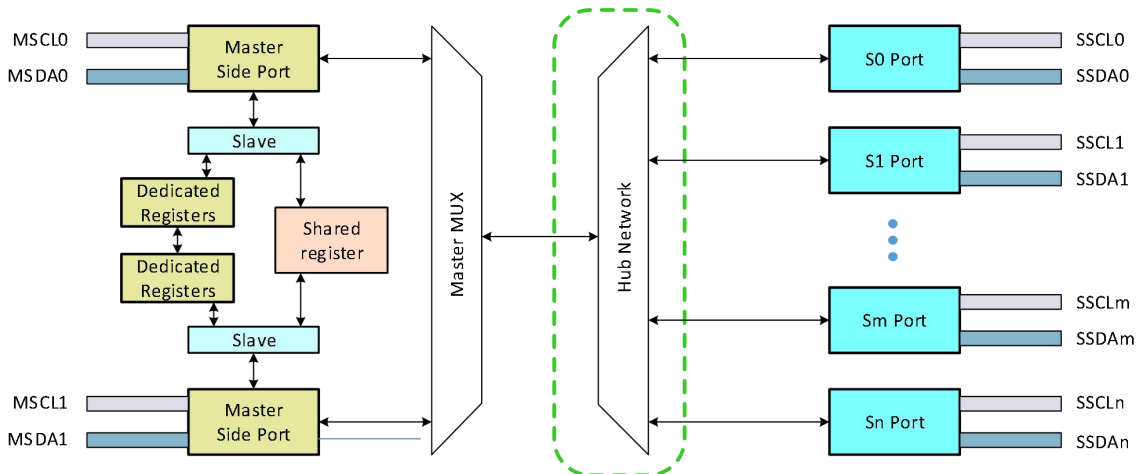
In MSEL pin selection mode, the user should switch the port when both master buses are in the idle state.

If the port selection is done by the in-band command to the master selection register, the user should terminate the I2C/I3C transaction with a STOP. The multiplexer does the switching when both ports are idle.

## 9.4 Hub Network

The PJ71MXX Hub device features a 1:N (N equals up to eight) I2C/I3C transparent bridging network. The Hub Network connects the selected Master Side Port to Enabled Slave Side Ports.

An intelligent Hub Controller checks the transaction frames that pass through the Slave Side Port Hub Network (the Hub) and dynamically adjusts the Hub Network operation mode.



**Figure12.Hub Network**

### 9.4.1 Hub Control Logic

The Hub control logic maintains the Hub network operation in a protocol-aware manner and dynamically adjusts the Hub network operating mode between the Open Drain operation and the Push/Pull operation.

A transaction starts with an I2C/I3C START condition and stops with an I2C/I3C STOP condition. Each Repeated Start condition begins a new transaction or sub-transaction without giving away control of the bus.

The Hub Control logic determines at any moment if the Hub Network should be in Open Drain mode or Push/Pull mode. In Push/Pull mode, the hub logic further determines which Slave Side Port in the Hub Network is the transaction target port as well as the direction of the Push/Pull data transmission (from master to slave or from slave to master).

#### 9.4.1.1 Target Port Determination

The target port is the Slave Side Port where the transaction target device is hooked up

A target port is determined in the following ways:

- Broadcast CCCs are considered as a write-operation to all slave ports. All Slave Side Ports are assumed as “target” and put “forwarding” mode.
- Direct CCC, the target port, is determined by the Slave Side Ports that acknowledged the ACK bit.
- For a private Read/Write transaction targeting a dynamic address, the target port is determined by the Slave Side Ports that acknowledged the ACK bit.
- In IBI mode, the Slave Side Port that wins the arbitration is assumed as the target port.

#### 9.4.1.2 Generic I3C Basic 1.0 Bus Operation

- In a Generic I3C Basic application, the Hub Network deals with mixed I2C/I3C devices in the same network. The framing of the I2C device and I3C device is different so the Hub Network Control logic needs to understand if each transaction target is an I2C device (with static address) or an I3C device.
- Refer to Hub Network for further details.

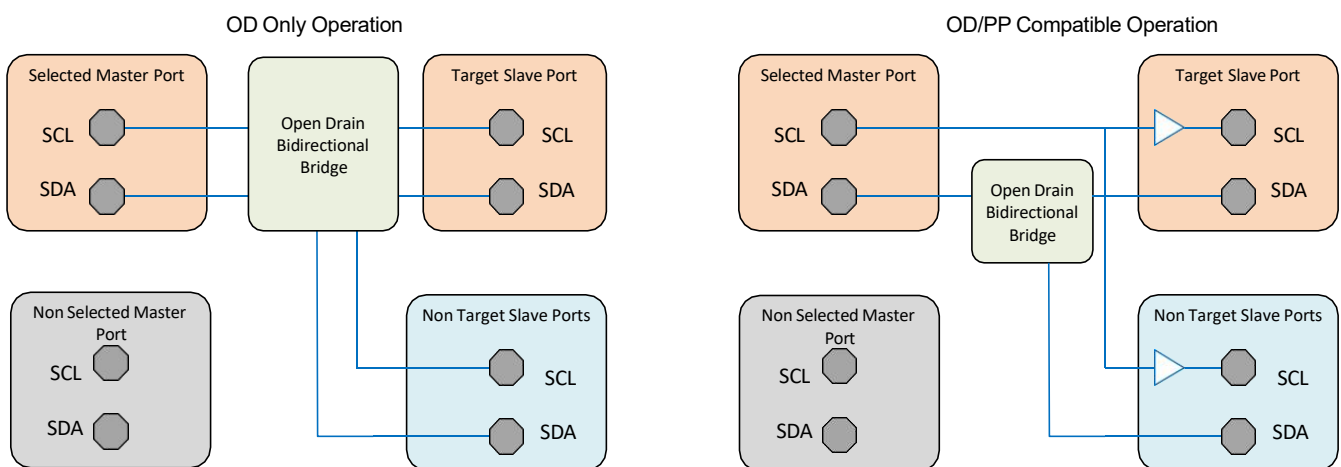
### 9.4.1.3 JEDEC SPD Bus Operation

JEDEC SPD I3C Bus has additional CCCs defined on top of the MIPI I3C specification. The Hub Controller analyzes all legal frame formats in the JEDEC Bus Context, even for the transactions are not supported by the PJ71MXX On-Chip Slave Interface. For instance, the Hub Controller can handle SETHID CCC. However, the PJ71MXX device's on-chip I3C Slave Interface ignores this CCC.

Refer to Hub Network for further details.

### 9.4.2 Open Drain Mode

If both the Master Side Port and Slave Side Port are in the Open-Drain Only mode, the SCL Signal is bridged bidirectionally. Otherwise, In I2C/I3C compatible open-drain operation mode, the SCL signal is re-driven from Master Side Port to Slave Side Port, i.e. it is unidirectional. The SDA port is bridged bi-directionally during open drain operation.



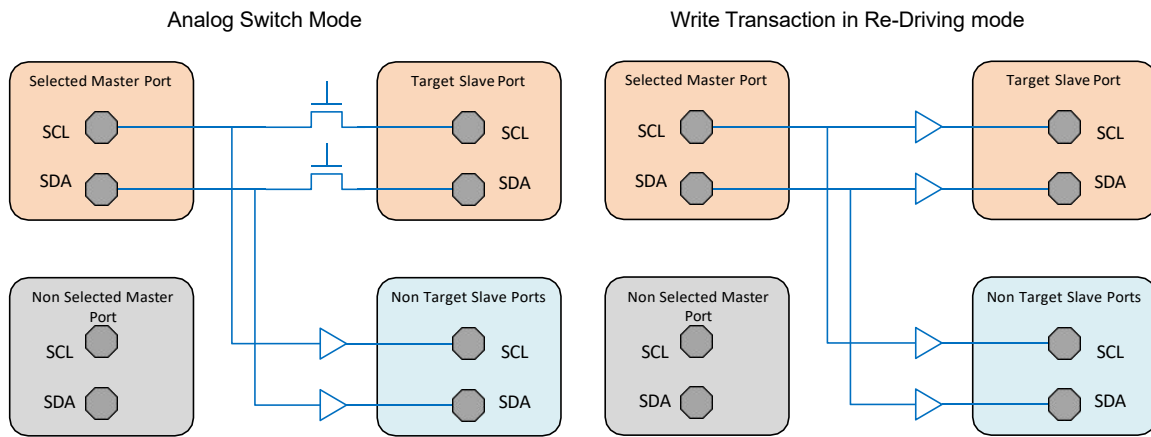
**Figure 13. Hub Network Open-Drain Mode IO Operation**

### 9.4.3 Push/Pull Mode

In Push/Pull mode, the “**target Slave Side Port**” is connected to the Master Side Port in the following ways:

- Via a high-performance analog switch for any traffic between Master Side Port and Slave Side Port;
  - To enable this operation, the IO Operation Voltage of the master and slave (VIOM and VIOS) should be identical.
- Via a level shifting re-driver. The direction of the buffer is determined by the transaction frame structure as defined in the I3C specification.
  - This operation allows level shifting between masters and slaves.
- Via a high-performance analog switch for SDA in a read transaction, and does the re-driving for write transaction or Push/Pull traffic from Master Side Port to Slave Side Port.

The **non-target Slave Side Ports** that are connected to the network re-drive the received master bus signal.



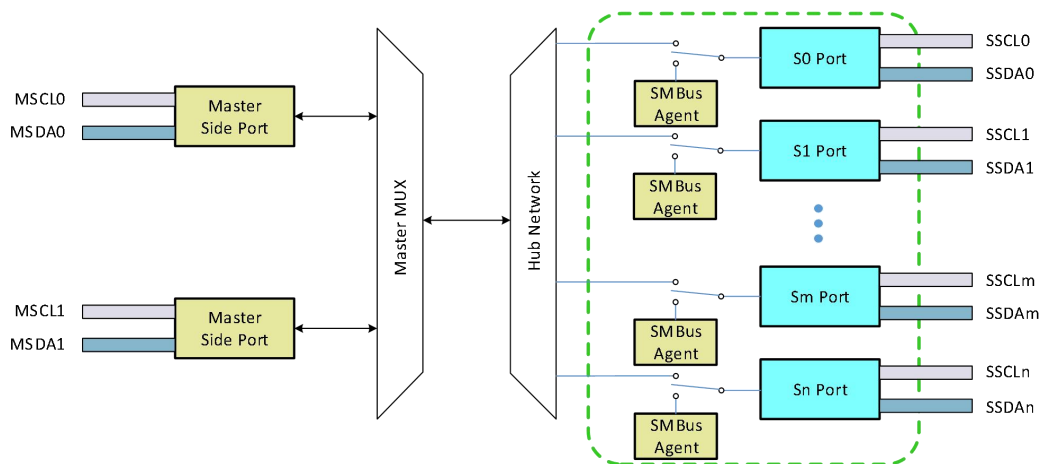
**Figure14.Hub Network Push/Pull Mode IO Operation**

### 9.4.4 Hub Network Transparency

The Hub Network connects the selected master interfacing port and a set of (configurable) slave interfacing ports in a transparent manner. The selected master sees all the connected slaves through the Hub Network as if they are directly connected on the same bus.

### 9.5 Slave Side Ports

The Slave Side Ports connect to the Hub Network and each port is associated with a bus agent.



**Figure15.Slave Side Port Overview**

#### 9.5.1 IO Operating Mode and Signaling Level

Each Slave Side Port consists of an SCL IO and an SDA IO.

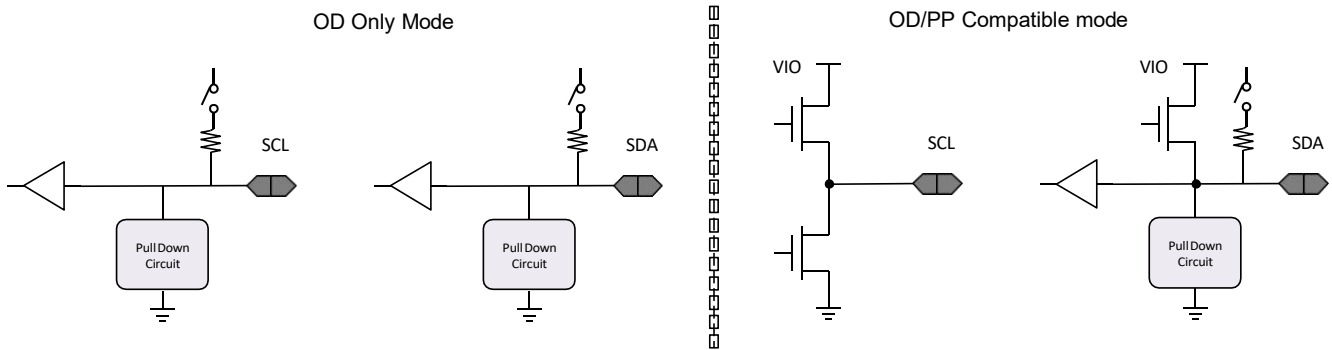
The port needs to be enabled to allow all features to be used. Ports that are not enabled will be in a High Impedance state.

Slave Interfacing Ports operates in the following two modes:

- Open Drain Only mode (see REG#23). In this mode, only SMBus or I2C protocol is supported. The voltage level of the signaling is up to 3.3V. The input logic buffer threshold level is set to 0.85V in this mode to comply with SMBUS Operation.
- Open-Drain/Push-Pull compatible Mode. In this mode, the Protocol can be I2C or I3C, and the IO OD/PP operation is determined by the protocol. The voltage level in this mode is defined by the VIOS0/1 pins. The VIOS0/1 pins are directly supplied on the board by the system, or optionally generated by on-chip LDO with voltage settings of 1.0, 1.1, 1.2, or 1.8V.
  - Users programs the Slave Interfacing Port operating voltage (see REG#22) if an on-chip regulator is used.

- The input logic detector threshold level will scale according to the VIOS0/1 voltage.

Slave Interfacing Ports can have an on-chip pull-up resistor to the VIOS rail or VDDIN rail. The on-chip pull-up resistor is connected to VDDIN rail if the Slave Interface Ports were in OD Only mode.



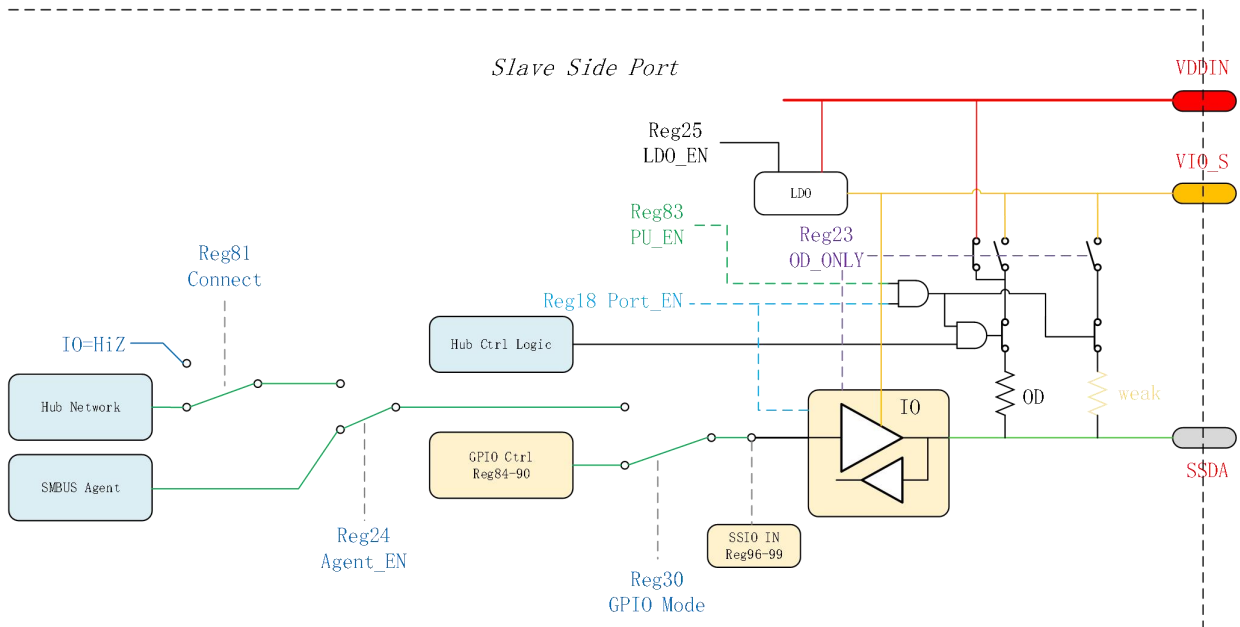
**Figure16.Slave Side IO Operation and Signaling Mode**

### 9.5.2 Slave Side Port Features and Configuration

Each Slave Side Port can be configured with:

- Port Enable/Disable Control (See REG#18)
- Pull-up Resistor Value and Enable (see REG#25, REG#83)
- GPIO Mode (see REG#30)
- SMBus Agent (see REG#24)
- Hub Network Connection (see REG#81)

These features are configurable via Slave Side Port Configuration and Control Registers. Below is the system level diagram of the Slave Side Port features, associated control register and priority.



**Figure17.Slave Side Port Operation and Control**

### 9.5.3 Port Enable

Each Slave Side Port is individually enabled or disabled via the Slave Side Port Enable Registers (see REG#18). A disabled port is put into a high-impedance state. Slave Side Port features are not accessible if the port is not enabled.

If enabled by REG#19[3], the Slave Side Port can be treated as disabled if the associated VIO level does not fall into the +/-12% of the nominal voltage as set by REG#22.

### 9.5.4 Pull-up Resistor

Each Slave Side Port's pull-up resistor is individually enabled or disabled via the Slave Side Port Pull-up Enable Registers (see REG#83). The pull-up value is per VIOS domain programmable (see REG#25).

A disabled port has pull-up resistor disabled.

in OD\_Only (see REG#23) configuration, both SCL and SDA channel has the pull-up connected to VIN rail, if pull-up is enabled for that port.

If OD\_Only bit for the port is not set, the port works in OD/PP compatible mode. In this mode, SCL channel will work in Push/Pull (PP) mode and pull-up will be disabled. The pull-up of SDA resistor will be connected to the VIOS rail.

### 9.5.5 GPIO Mode

The Slave Side Ports can be configured to work in GPIO mode. The configuration (see REG#30) is per-port. Once configured as GPIO, the Port is disconnected from the Hub Network and the SMBus Agent is unable to control the port. The IO of Slave Interfacing Port is enabled and used as GPIO.

The OD\_Only (see REG#23) Setting and Pull-up Setting (See REG#83, REG#25) of the Slave Side Port are in effect in GPIO mode. In OD only mode, the GPIO output mode relies on a pull-up resistor (internal or external) for logic high output.

### 9.5.6 Agenting

In Agenting, the Port is disconnected from the Hub Network. A Bus Agent attached to the Slave Interfacing Port can be enabled to handle all the transactions with downstream devices.

### 9.5.7 Hub Network Connection

Slave Side Port can connect to or disconnect from the Hub Network if the port is enabled, not in GPIO mode and not in Agenting mode.

Each Slave Side Port can be independently connected, or disconnected from the Hub Network.

The Port-Connection Register (see REG#82) define if the Slave Interfacing Port will connect to the Hub Network.

- If a slave interfacing port is connected to the Hub Network, it works in the transparent bridging mode with other ports. The Hub Network controls the IO operation.
- If a slave interfacing port is disconnected from the Hub Network, it stays in standby mode as in a network isolation application. The IO in disconnected Port does not drive the bus, but allows the pull-up resistor, if enabled (see REG#83), to pull the bus high.

#### 9.5.7.1 Hub Network Connected Operation

##### 9.5.7.1.1 IO Operates in I2C OD Only Mode

The Slave Side Port in OD Only mode can only be connected to the Hub Network if selected master side port is in Open-Drain only mode. Both SCL and SDA are in bidirectional Open-Drain operation and are able to tolerate up to 3.3V signaling.

If the master ports are I3C masters, an "OD Only" Slave Side Port is unable to operate at the Push/Pull speed of I3C transactions and hence the port should be disconnected from the Hub Network.

##### 9.5.7.1.2 IO Operates in I2C/I3C OD/PP Compatible Mode

In this operation, the SCL channel is push-pull output only.

- SCL path works in Push/Pull Operation, even in I2C mode or I3C address arbitration header. The IO operates in voltage level defined by the VIOS pin.

- When a Slave Side Port is a target port and “Analog Switch enable for target port” (see HubNetwork\_PP\_Analog\_Switch\_En) is enabled, the output driver will be disabled, an analog switch channel between the “selected Master Side Port input SCL” and this “target port output SCL” is established during Push/Pull operation, until a repeat start or stop is detected.

SDA channel is open-drain and push/pull dynamically switching as per I3C protocol.

- In Open Drain mode, this channel connects to the selected Master Side Port’s SDA channel with a bi-directional open-drain circuit. An on-chip pull-up resistor is engaged if enabled.
- In Push-Pull mode, the IO operates in IO voltage as defined by the VIOS pin. The direction of the buffer is determined by the I3C protocol. If the Slave Side Port is a target port and “Analog Switch enable for target port” (see HubNetwork\_PP\_Analog\_Switch\_En), an analog switch channel between the “selected Master Side Port output SDA” and this “target port input SDA” is established during Push/Pull operation, until a repeat start or stop is detected.

### **9.5.7.2 Hub Network Disconnected Operation**

#### **9.5.7.2.1 Transparent Standby**

In this operation, the Port’s disconnection from the Hub Network is temporary. It connects to the Hub Network later and disconnects the port by putting it into standby mode. This is a typical setup for the Network Partition operation of the Hub Network. In a Network Partition application, a set of ports are disconnected from the Hub Network to allow the master to access a portion of the I3C hierarchy at different times.

The disconnected Slave Side Ports may receive IBI from the downstream devices in the standby hierarchy. Since the Port is disconnected from the Hub Network, the IBI does not participate the arbitration through the Hub Network and pending at the port. The PJ71MXX device can facilitate the master to monitor and service the disconnected ports.

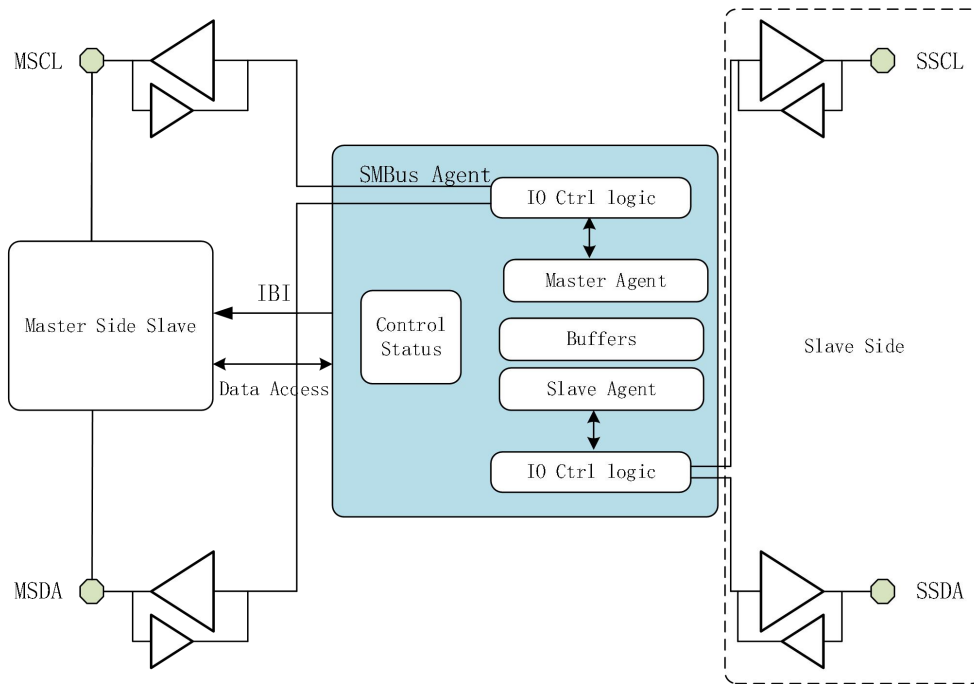
- PJ71MXX device has status registers (see SSSDA\_IO\_IN) that report the SDA level and can be read by the master with Polling.
- PJ71MXX device has IO input detection logic that is configured (see REG#88) to detect a fixed level or any input transition. The detection result (see REG#98), if enabled, (see REG#89) generates an IBI from the on-chip slave interface to notify pending port events.

## **9.6 Bus Agent**

### **9.6.1 Bus Agent Overview**

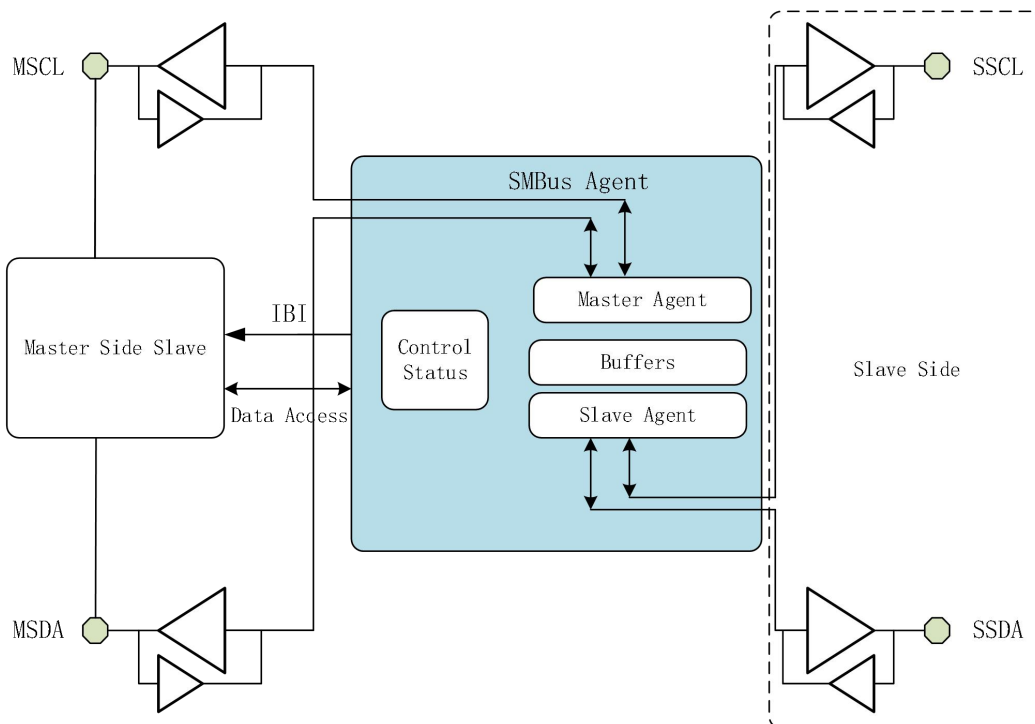
Each Slave Interfacing Port is configurable to connect to a SMBus Agent. The SMBus Agent supports Slave Interfacing Port side SCL stretching and is capable of interfacing with I2C Slaves, SMBus Slaves and SMBus devices working in messaging mode that initiates write transactions like MCTP message.

The SMBus Agent of each port is comprised of an SMBus master agent and an SMBus slave agent. The master and slave agents operate in parallel and independently.



**Figure 18. Slave Side Port Bus Agent Attachment**

To have an agent actively handle a slave side port transaction, the agent's associated port need to be set to Agenting mode (see REG#24) and disable the GPIO mode (see REG#30). The agent should be the only entity that interacts with downstream devices behind the Slave Side Port.



**Figure 19. System View when Bus Agent is Enabled**

### 9.6.2 SMBus Agent Data Buffer

Each SMBus Agent has a dedicated 256 bytes data buffer. This buffer is partitioned into three sections, and is accessed via the paged register space (see Paged Register Map).

**Table17. Bus Agent Data Buffer Mapping**

Byte	Content	Comments
0-87	Master Agent Buffer, 88 bytes. This block is partitioned to three sections: <ul style="list-style-type: none"> <li>Transaction descriptor: Byte 0-3</li> <li>Write or Read data buffer: Byte 4-87.               <ul style="list-style-type: none"> <li>For write transaction, this is the transaction payload; For write followed by Read transaction, this is the write transaction's payload plus the data that is received from the Slave in the followed Read Transaction</li> </ul> </li> </ul>	Maps to Page#: 16 + 4*Port_ID+0 byte 0 maps to offset[0] in the page Example: for Slave Side Port 2, the Master Agent buffer 88 bytes maps to page # 24, byte offset 0 to byte offset 87 in the page.
88-95	Slave Agent local Slave address list, eight bytes. Up to eight addresses can be configured as local slave addresses. The Local Slave address maps to bit [7:1] of any of the address list byte. If the received transaction's target slave address matches any local address list entries, it will not launch IBI and do a silent discard.	Maps to Page#: 16 + 4*Port_ID+1 byte 88 maps to offset[0] in the page

**Table18. Bus Agent Data Buffer Mapping**

Byte	Content	Comments
96-175	Slave Agent Data Buffer 0, 80 bytes	Maps to Page#: 16 + 4*Port_ID+2 byte 96 maps to offset[0] in the page
176-255	Slave Agent Data Buffer 1, 80 bytes	Maps to Page#: 16 + 4*Port_ID+3 byte 176 maps to offset[0] in the page

### 9.6.3 SMBus Master Agent

The PJ71MXX Master Agent can interpret the transaction descriptor from the master agent buffer and initiate an SMBus master transaction at the Slave Side Port. The Master Agent supports SCL stretching.

#### 9.6.3.1 Master Transaction Descriptor

**Table19. Bus Agent Master Transaction Mapping**

Byte	Content	Comments
0	Bit[7:1] is the Target Slave Address, Bit[0] is the R/W bit	Map to first byte of the SMBus transaction
1	Bit[0]: Transaction Type 0 = Single Transaction; Transaction (Write or Read) 1 = If Byte0[0] is Write, a Read is followed by a Repeat Start Bit[2:1]: Transaction Speed when not stretched. 00 = 100KHz 01 = 200KHz 10 = 400KHz 11 = 1000KHz Bit[3] = Enable SDA stuck at low detection during the transaction Bit[4] = Enable using SCL_Low timeout to clear the SDA stuck at low. Bit[5] = Enable I3C framing: Handoff at Ack bit, and Parity generation at the T bit. Bit[6] = Enable Bus Reset Pattern Generation Bit[7] = Enable Transaction No-Stop Configuration 0: Transaction ends with Stop symbol, then stay at idle (SCL/SDA=High) 1: The transaction end with SDA go high before SCL go high, No Stop Symbol, then stay at idle condition (SCL/SDA=high)	
2	For Write Transaction - WriteBytes: # of Bytes for Write For Bus Reset Pattern Generation - bit[5:0] = # of ms of SCL_Low. 0=64ms, 1=1ms, 2=2ms, ... 63=63ms. Bit[7:6] = Reserved.	WriteBytes Valid value: 0-83. WriteBytes + ReadBytes <=83
3	ReadBytes: # of Bytes for Read	Valid value: 0-83. WriteBytes + ReadBytes <=83

Byte	Content	Comments
4- (WriteBytes+3)	Payload data for Write Transaction	(does not exist if WriteBytes = 0)
(WriteBytes+4)- (WriteByte+3+ ReadBytes) Maximum#: 87	Payload data for Write Transaction, or Received data for Write followed by Read transaction	Byte #87 may be used to return transaction execution context information when transaction finishes

### 9.6.3.2 Master Transaction Execution

Once the active master configured the SMBus descriptor and started the transaction (see REG#80), the Agent launches the transaction at the attached slave side port.

The execution result are reflected in the Master Agent Status Register (see REG#100). When the transaction finishes, the status and flag registers are updated. For any Master Side Port, if the IBI of its attached I3C Slave is enabled and associated Agent related IBI generation is enabled (see REG#27) an IBI will be generated to the bus at that Master Side Port.

The Master Agent Status Register REG#100 is comprised of IBI flags and a transaction execution return code. The four bit return code is encoded as in the following table.

**Table20.Master Agent 4-bit Return Codes**

Return Code	Meaning
0000	Transaction finish successfully
0001	Device Absence: Write Address received NAK, transaction finish
0010	Device busy: Write Address received ACK, data received NAK - transaction finish. The # of acknowledged data bytes will be saved in transaction buffer Byte#87.
0011	Device Read not Ready: For Write followed by Read transaction - Write transaction received ACK, but after Repeat Start, Read address received NAK
0100	Transaction Sync issue Recovered: SDA stuck at low timeout detected, and recovered by 9 SCL pulses
0101	Transaction Sync issue Bus Clear: SDA stuck low timeout detected, and it is not recovered by 9 SCL pulse. It is recovered by SCL low 35ms
0110	Bus Fault: SDA stuck low timeout detected. After toggle 9 SCL and STOP generated, the SDA stuck at low. After keeping SCL low for 25ms, the SDA/SCL are not in idle state after 35ms. System attention and action is needed
0111	Arbitration loss.
1000	SCL timeout: Master Bus Agent detected SCL time out during the transaction and aborted.
1001	SCL timeout: Slave Bus Agent detected SCL time out during the transaction and aborted; Master Agent i not active.
1001~1111	Reserved

The host resides at the master side port takes further actions, like re-try, according to the return code value, if the transaction does not finish successfully.

## 9.6.4 SMBus Slave Agent

The PJ71MXX Slave Agent receives a write transaction from the Slave Side Port. The Slave Agent does not interpret the transaction, but saves the whole transaction, including the slave address, to a packet buffer.

The Slave Agent works in parallel with the Master Agent. That means that the slave agent can capture write-transactions initiated by the master agent, if enabled in REG#108. The local address list could be configured to have the Slave Agent to discard the transactions that targeting at local slaves.

When the transaction finishes, the Agent event flag registers are updated. These registers are routed to the on-chip slaves' IBI logic of the two masters, if enabled. If a Hot Join message is received, the Slave Agent will treat it as a transaction finish after the ACK bit, regardless it is ACKed or NACKed.

### 9.6.4.1 Dual Buffer Operation

The Slave Agent has access to two transaction buffers: buffer 0 and buffer 1. It can receive the second slave transaction immediately after the first slave transaction is received and before the master retrieves the buffered data. The data will be saved alternating to the two buffers, i.e. it will starts from buffer 0, then buffer 1, then buffer 0, then buffer 1 and iterates.

If both buffers are full, the slave agent ACKs the subsequent transaction's address byte but NAKs the data bytes of the transaction.

After a master serviced the IBI and read the buffered transaction data, it clears the associated buffer data ready status (see Slave\_Agent\_Buf0/1\_Received\_Flag in REG#100), so that the buffer can be used for the next transaction.

### 9.6.4.2 Hot Join Message Reception

The Slave Agent can generate the SCL clock to detect an I3C Hot Join message. See definition in REG#108 for details. This feature helps the discovery of I3C capable devices at the Slave Side Port.

### 9.6.4.3 Local Traffic Silent Discard

Each Slave Agent observes up to eight "local slave" addresses. Traffic targeted to these addresses are considered as "local traffic" and are silently discarded.

### 9.6.4.4 Packet Buffer Transaction Byte Mapping

**Table21.Bus Slave Agent – Received Data Buffer Mapping**

Packet Buffer Byte Index	Received Transaction Data Mapping
0	PL: Packet Length, total byte count received in the SMBUS transaction, from Start to Stop, including the slave address byte
1	First byte received (the address byte)
2	Second byte received (protocol specific)
3	Third byte received (protocol specific)
...	(Sequentially Received packet data)
PL	Last byte received before the STOP symbol
PL+1	Undefined data
...	Undefined data
79	Undefined data

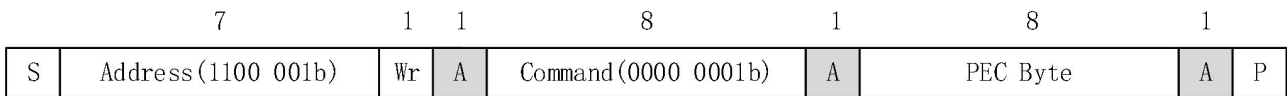
### 9.6.5 SMBus Agent Related Registers

- REG#23, Slave Side Port Interface Operation Mode Setting
- REG#81, Slave Side Port to Master Side Port Hub Connection Enable
- REG#33, Slave Port SMBus Agent IBI Status
- REG#80, Slave Side Port Master Agent Transaction Start Control
- REG#100-107, SMBus Agent Status Register

### 9.6.6 Example SMBus Transaction Descriptor Mapping

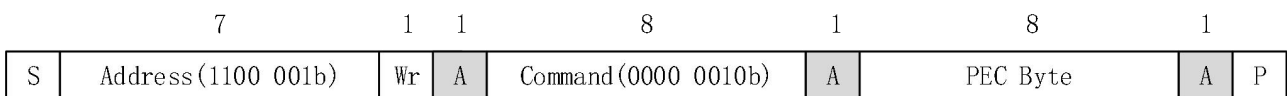
#### 9.6.6.1 SMBus “Prepare to ARP” Command with PEC

Byte	Content	Comments
0	1100_0010	Address
1	0000_0110	Single Write / 1MHz
2	0000_0010	2 bytes to write (command + PEC)
3	0000_0000	Not applicable for this transaction
4	0000_0001	“Prepare to ARP” Command code
5	????_????	PEC Byte
6-87	xxxx_xxxx	Not used



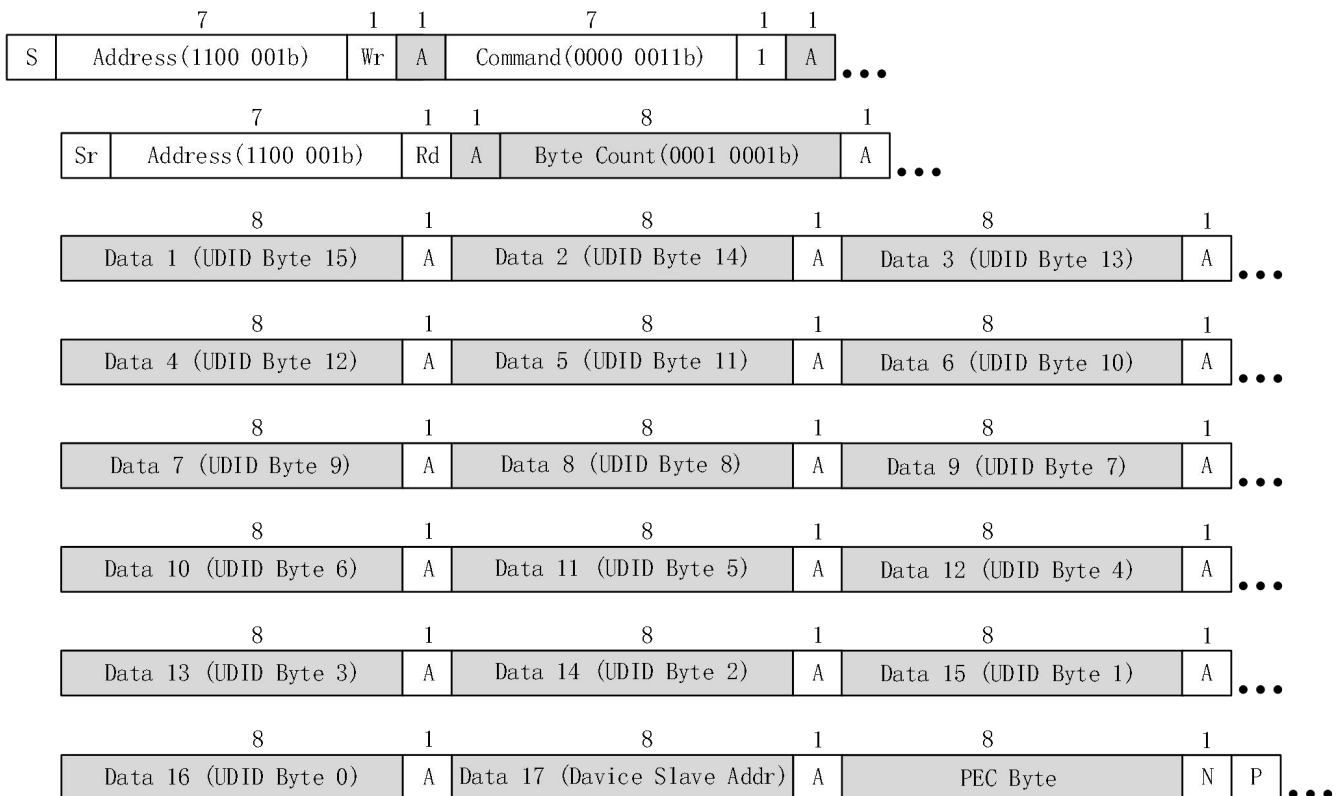
#### 9.6.6.2 SMBus “Reset Device” Command with PEC

Byte	Content	Comments
0	1100_0010	Address
1	0000_0110	Single Write / 1MHz
2	0000_0010	2 bytes to write (command + PEC)
3	0000_0000	Not applicable for this transaction
4	0000_0010	“Reset Device” Command code
5	????_????	PEC Byte
6-87	xxxx_xxxx	Not used



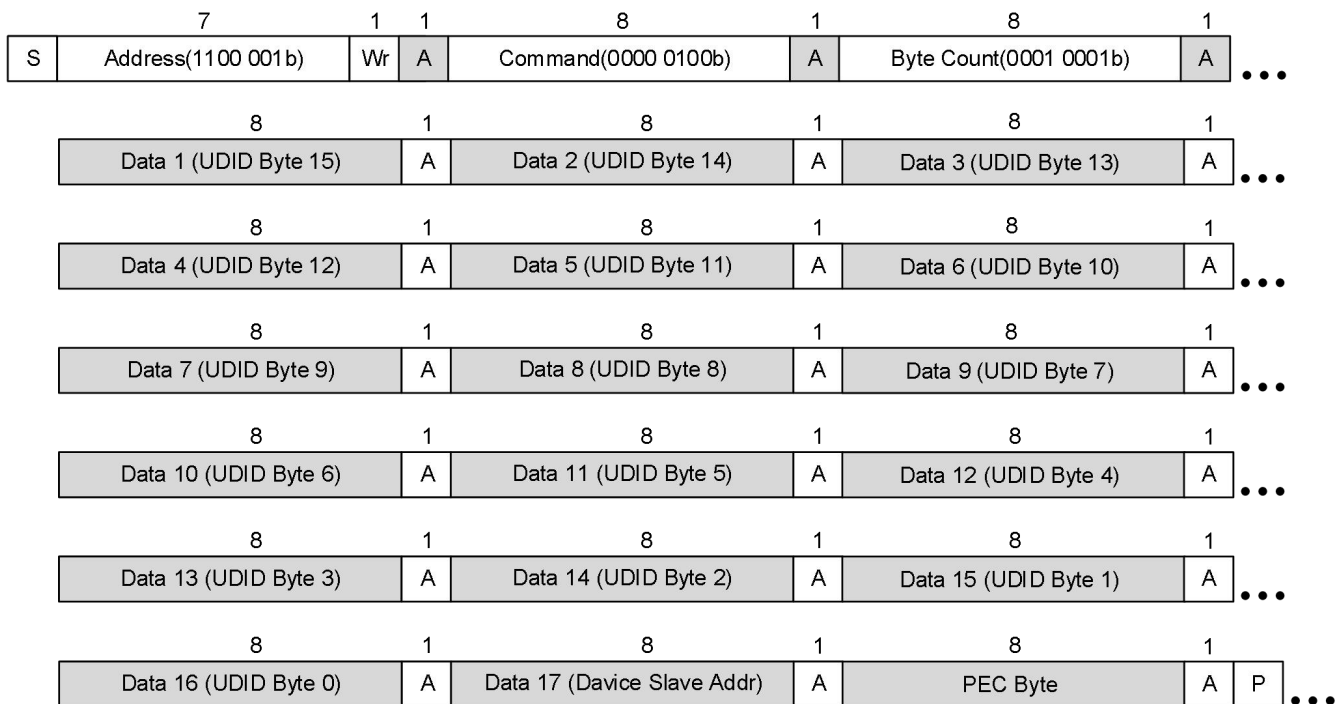
**9.6.6.3 SMBus “GET UDID” General Command with PEC**

Byte	Content	Comments
0	1100_0010	Address
1	0000_0111	Combo-Write followed by Read / 1MHz
2	0000_0001	1 byte to write (command)
3	0001_0011	19 bytes to read
4	0000_0011	“Get UDID” Command code
5	0001_0001	Received Byte Count
6	????_????	Data 1 (UDID Byte 15)
7	????_????	Data 2 (UDID Byte 14)
8	????_????	Data n (UDID Byte 16-n)
9-21	????_????	Data 16 (UDID Byte 0)
22	????_????	Data 17 (Device Slave Address)
23	xxxx_xxxx	PEC Byte
24-87	xxxx_xxxx	Not used



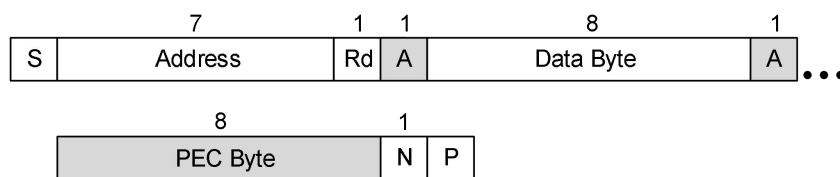
**9.6.6.4 SMBus “Assign Address” Command with PEC**

Byte	Content	Comments
0	1100_0010	Address
1	0000_0110	Single Write / 1Mhz
2	0001_0100	20 bytes to write (command + PEC)
3	0000_0000	Not applicable for this transaction
4	0000_0100	“Assign Address” Command code
5	0001_0001	“Byte Count” of the Command
6	????_????	Data 1 (UDID Byte 15)
7	????_????	Data 2 (UDID Byte 14)
8-20	????_????	Data n (UDID Byte 16-n)
21	????_????	Data 16 (UDID Byte 0)
22	????_????	Data 17 (Assigned Address)
23	????_????	PEC Byte
24-87	xxxx_xxxx	Not used



### 9.6.6.5 SMBus “Receive Byte” Command with PEC

Byte	Content	Comments
0	1100_0011	Address
1	0000_0110	Single Read / 1Mhz
2	0000_0000	0 bytes to write
3	0000_0010	2 byte to read
4	????_????	Data Read from the Bus
5	????_????	PEC Byte
6-87	xxxx_xxxx	Not used



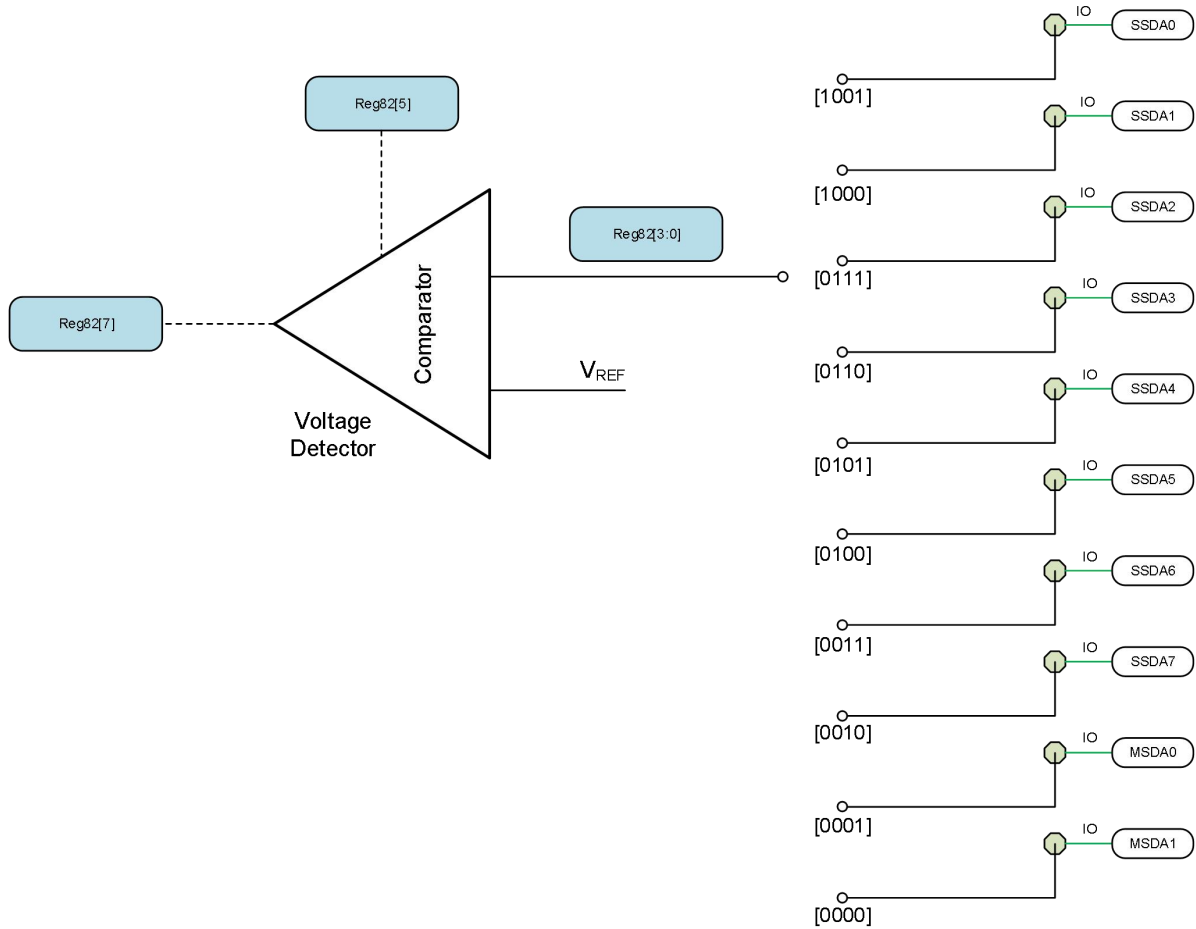
### 9.6.6.6 Generic I2C Write Followed by Read command

Byte	Content	Comments
0	1010_0100	SlaveAddr=101_0010, Write
1	0000_0111	Write followed by Read, SCL = 1Mhz
2	0000_0010	2 bytes to write
3	0000_0100	4 byte to read
4	0100_0000	Byte 0 of Write Transaction (e.g. an address byte)
5	0000_0110	Byte 1 of Write Transaction (e.g. 2nd address byte)
6	Read Byte0	Received Byte 0 of the Read Transaction (Received Read bytes starting at buffer location of "WriteBytes + 4": 2+4 =6)
7	Read Byte1	Received Byte 1 of the Read Transaction
8	Read Byte2	Received Byte 2 of the Read Transaction
9	Read Byte3	Received Byte 3 of the Read Transaction
10-86	Undefined	
87	Potential Context # (e.g. 0)	Context byte - If NAK received in the Write Transaction, this byte contains the number of bytes get Acknowledged

### 9.7 IO Voltage Detector

S	8 1010 0100	1 A	8 0100 0000	1 A	8 0000 0110	1 A			
Sr	8 Read Byte 0	1 A	8 Read Byte 1	1 A	8 Read Byte 2	1 A	8 Read Byte 3	1 N	1 P

The PJ71MXX has an on-chip comparator which selectively detects the MSDA and SSDA pins' IO voltage. The Threshold voltage, Pin selection and detection result are mapped to REG#82. Vref can be configured to 1.5V or 2.1V.



**Figure20.IO Voltage Detector**

## 10 Registers

### 10.1 Register Space and Attribute

All user visible registers are volatile and have attributes as defined in the following table.

**Table22.Register Attributes**

Attribute	Abbreviation	Description
Read Only	R	This bit is read via the I2C/I3C interface. Writes have no effect.
Write	W	This bit is written via the I2C/I3C interface.
Write by Other Port	OW	This bit can only be written by the other port
Write Protected	P	This bit is protected, and cannot be written to unless the protection code has been written into the protection code registers.
Write 1 Clear	W1C	This bit can be cleared with a "1" written via the I2C/I3C interface.
Write 1 Set	W1S	This bit can be set with a "1" written via the I2C/I3C interface.
Write 1 Act	W1A	Writing "1" to this bit triggers an action in the device.
Blank	B	This bit is reserved for future use. Writing to the bit will have no effect. However, the master should always write "0" to maintain compatibility. Reading from bit returns "0"
Reserved	V	This bit is reserved for future expansion or vendor internal use. The bit may return any value when read. The master should avoid writing to reserved registers. If writing to this bit cannot be avoided, the software must preserve the value read unless otherwise indicated.

### 10.2 Register Map

The port associated registers are "Dedicated" to that port.

The device-level registers, shared buffers, or messaging registers, are shared by two on-chip slaves. These registers map to register space of both on-chip slaves and are labeled as "Shared" in the following table.

Reserved registers may be defined in newer version of Hub device. User should avoid writing to reserved registers to avoid compatibility issue. Reading from reserved registers returns undefined value.

#### 10.2.1 On-Chip Slave Register Space Map

**Table23.Register Map**

Register Address	Master 0/1 Sharing	Description
<b>Device Information Registers</b>		
0-1	Shared	Device Information
2-7	Dedicated	M0/M1 PID. M0 port could write to M1 port's PID register.
8-9	Shared	BCD, DCR
10	Shared	Device Capability
11	Shared	Device Revision
<b>Device Configuration Registers</b>		
16	Dedicated	Device Configuration Protection Mode

(1). Volatile registers lose their configuration value after power cycle

**Table 23. Register Map (Cont.)**

Register Address	Master 0/1 Sharing	Description
17	Dedicated	Master Side Port configuration
18	Shared	Slave Side Port Enable
19	Shared	Hub Device Configuration
20	Shared	IO Drive Strength Configuration
21	Dedicated	Hub Network Operation Mode Configuration
22	Shared	IO Interface Voltage LDO Setting
23	Shared	Slave Side Port IO OD Only Signaling Mode Setting
24	Shared	Slave Side Port SMBus Agent Enable
25	Shared	LDO Enable and Slave Side Port Pull Up Resistor Value Configuration
26	Dedicated	Master Side Interfacing Port IBI Generation Enable Configuration
27	Dedicated	Slave Side Port SMBus Agent IBI Enable Configuration
28	Dedicated	User Customized IBI MDB Value
29	Shared	SETBUSCON JEDEC Context Identification ID
30	Shared	Slave Side Port GPIO Mode Enable
31		Reserved
<b>Device Status and IBI Related Registers</b>		
32	Dedicated	Device and Port IBI Status
33	Dedicated	Slave Side Port SMBus Agent IBI Status
34	Dedicated	Reference/Link to other master's IBI message buffer Byte 0
35	Dedicated	Reference/Link to other master's IBI message buffer Byte 1
36	Dedicated	Reference/Link to other master's IBI message buffer Byte 2
37	Dedicated	Reference/Link to other master's IBI message buffer Byte 3
38-39		Reserved
40	Dedicated	This master's IBI message buffer Byte 0
41	Dedicated	This master's IBI message buffer Byte 1
42	Dedicated	This master's IBI message buffer Byte 2
43	Dedicated	This master's IBI message buffer Byte 3
44	Dedicated	This master's IBI message Request Register
45-47		Reserved
<b>Generic Scratch Board Registers</b>		
48	Shared	Generic Message Scratch Board register Byte 0
49	Shared	Generic Message Scratch Board register Byte 1
50	Shared	Generic Message Scratch Board register Byte 2
51	Shared	Generic Message Scratch Board register Byte 3
52-55		
<b>Master Interfacing Port Multiplexer Control/Status Registers</b>		
56	Dedicated	Master Multiplexer Selection Control
57	Dedicated	Master Multiplexer Selection Status
58-63		Reserved
<b>Dynamic Address Flags</b>		
64	Shared	Slave Address Dynamic Address Flag - Byte 0
65	Shared	Slave Address Dynamic Address Flag - Byte 1
66	Shared	Slave Address Dynamic Address Flag - Byte 2
67	Shared	Slave Address Dynamic Address Flag - Byte 3

**Table 23. Register Map (Cont.)**

Register Address	Master 0/1 Sharing	Description
<b>Dynamic Address Flags</b>		
68	Shared	Slave Address Dynamic Address Flag - Byte 4
69	Shared	Slave Address Dynamic Address Flag - Byte 5
70	Shared	Slave Address Dynamic Address Flag - Byte 6
71	Shared	Slave Address Dynamic Address Flag - Byte 7
72	Shared	Slave Address Dynamic Address Flag - Byte 8
73	Shared	Slave Address Dynamic Address Flag - Byte 9
74	Shared	Slave Address Dynamic Address Flag - Byte 10
75	Shared	Slave Address Dynamic Address Flag - Byte 11
76	Shared	Slave Address Dynamic Address Flag - Byte 12
77	Shared	Slave Address Dynamic Address Flag - Byte 13
78	Shared	Slave Address Dynamic Address Flag - Byte 14
79	Shared	Slave Address Dynamic Address Flag - Byte 15
<b>Slave Side Port Control Registers</b>		
80	Shared	Slave Side Port Master Agent Transaction Start Control
81	Shared	Slave Side Port Hub Network Connection Enable
82	Shared	Master/Slave Side Port SDA Voltage Detection
83	Shared	Slave Side Port Pull Up resistor Enable
84	Shared	Slave Side Port GPIO Mode SSCL Enable
85	Shared	Slave Side Port GPIO Mode SSDA Enable
86	Shared	Slave Side Port GPIO SSCL Output Level
87	Shared	Slave Side Port GPIO SSDA Output Level
88	Shared	Slave Side Port Input Detection Mode Configuration
89	Dedicated	Slave Side Port SSCL IO Input Detected Event IBI Enable
90	Dedicated	Slave Side Port SSDA IO Input Detected Event IBI Enable
91-95		Reserved
<b>Slave Side Port Status Registers</b>		
96	Shared	Slave Side Port IO SSCL Input Status
97	Shared	Slave Side Port IO SSDA Input Status
98	Shared	Slave Side Port SSCL IO Detection Flags
99	Shared	Slave Side Port SSDA IO Detection Flags
<b>SMBus Agent Status Registers</b>		
100-107	Shared	SMBus Agent Status Register
108	Shared	Slave Side Port Bus Agent Configuration
109-120	Reserved	Reserved
<b>Special Registers</b>		
121	Shared	LDO Power Good and MSEL input Status
122	Dedicated	Bus Reset SCL Timeout Setting
123	Dedicated	I3C Protocol Error Flags
124	Dedicated	Device Command
125	Dedicated	On-Chip Slave Reference Status
126	Dedicated	Slave Address setting
127	Dedicated	Page Pointer
128-255		Vendor Registers and paged buffers

## 10.2.2 Two On-Chip Slaves' Register Space and Sharing/Selection Map

In the following table, “follow master selection control” means that if the REG#19[1] bit is set, only the selected master can access to the specified device-level registers; the unselected master can only access the port's associated registers.

**Table24.Two On-Chip Slaves' Register Space and Sharing Map**

M0 Address	M0 Register	Sharing, REG19[1] Master Selection Ctrl	M1 Register	M1 Address
<b>Device Info Registers</b>				
0-1	Shared			0-1
2-7	Dedicated		Dedicated	2-7
8-11	Shared			8-11
12-15	Reserved			12-15
<b>Device Configuration Registers</b>				
16	Dedicated		Dedicated	16
17	Dedicated		Dedicated	17
18	Shared, follow master selection control			18
19	Shared			19
20	Shared, follow master selection control			20
21	Dedicated		Dedicated	21
22	Shared, follow master selection control			22
23	Shared, follow master selection control			23
24	Shared, follow master selection control			24
25	Shared, follow master selection control			25
26	Dedicated		Dedicated	26
27	Dedicated		Dedicated	27
28	Dedicated		Dedicated	28
29	Shared, follow master selection control			29
30	Shared, follow master selection control			30
31				31
<b>Device Status and IBI Related Registers</b>				
32	Dedicated		Dedicated	32
33	Dedicated		Dedicated	33
34	A (Read Only)		E (Read Only)	34
35	B (Read Only)		F (Read Only)	35
36	C (Read Only)		G (Read Only)	36
37	D (Read Only)		H (Read Only)	37
38-39				
40	E (R/W)		A (R/W)	40
41	F (R/W)		B (R/W)	41
42	G (R/W)		C (R/W)	42

**Table 24. Two On-Chip Slaves' Register Space and Sharing Map (Cont.)**

M0 Address	M0 Register	Sharing, REG19[1] Master Selection Ctrl	M1 Register	M1 Address
<b>Device Status and IBI Related Registers</b>				
43	H (R/W)		D (R/W)	43
44	Dedicated		Dedicated	44
45-47				46-47
<b>Generic Scratch Board Registers</b>				
48	Shared			48
49	Shared			49
50	Shared			50
51	Shared			51
52-55				52-55
<b>Master Interfacing Port Multiplexer Control/Status Registers</b>				
56	Dedicated		Dedicated	55
57	Dedicated		Dedicated	57
58-63				58-63
<b>Dynamic Address Flags</b>				
64	Shared, follow master selection control			64
65	Shared, follow master selection control			65
66	Shared, follow master selection control			66
67	Shared, follow master selection control			67
68	Shared, follow master selection control			68
69	Shared, follow master selection control			69
70	Shared, follow master selection control			70
71	Shared, follow master selection control			71
72	Shared, follow master selection control			72
73	Shared, follow master selection control			73
74	Shared, follow master selection control			74
75	Shared, follow master selection control			75
76	Shared, follow master selection control			76
77	Shared, follow master selection control			77
78	Shared, follow master selection control			78
79	Shared, follow master selection control			79
80	Shared, follow master selection control			80
81	Shared, follow master selection control			81
82	Shared, follow master selection control			82
83	Shared, follow master selection control			83
84	Shared, follow master selection control			84
85	Shared, follow master selection control			85
86	Shared, follow master selection control			86

**Table 24. Two On-Chip Slaves' Register Space and Sharing Map (Cont.)**

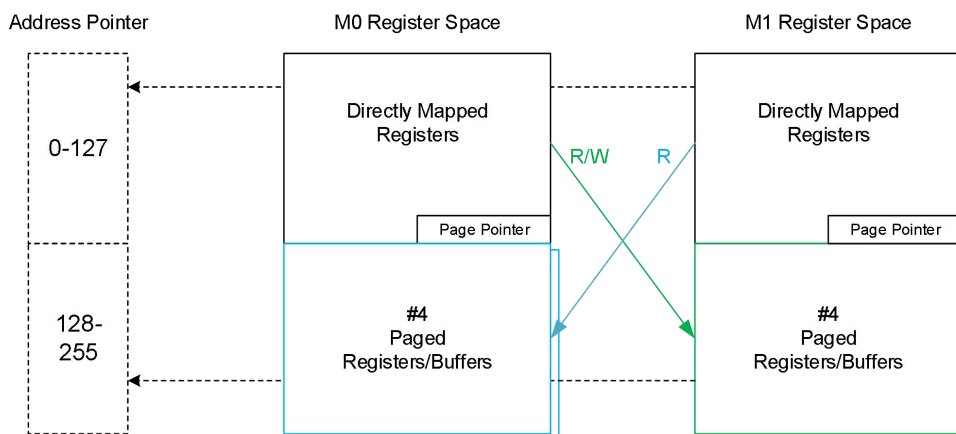
M0 Address	M0 Register	Sharing, REG19[1] Master Selection Ctrl	M1 Register	M1 Address
<b>Slave Side Port Control Registers</b>				
87	Shared, follow master selection control			87
88	Shared, follow master selection control			88
89	Dedicated		Dedicated	89
90	Dedicated		Dedicated	90
<b>Slave Side Port Status Registers</b>				
96	Shared			96
97	Shared			97
98	Shared, follow master selection control			98
99	Shared, follow master selection control			99
<b>SMBus Agent Status Registers</b>				
100-108	Shared, follow master selection control			100-108
<b>Special Register</b>				
121	Shared			121
122	Dedicated		Dedicated	122
123	Dedicated		Dedicated	123
124	Dedicated		Dedicated	124
125	Dedicated		Dedicated	125
126	Dedicated		Dedicated	126
127	Dedicated		Dedicated	127
128-255	Paged registers – Refer to the Paged buffer Map below			128-255

### 10.2.3 Paged Register Map

Note: Although each page has an 128 byte register space, only defined space is accessible. For instance, the Bus Agent buffers take a portion of the address space, starting from address 0 in the page.

**Table25.Paged Register Map**

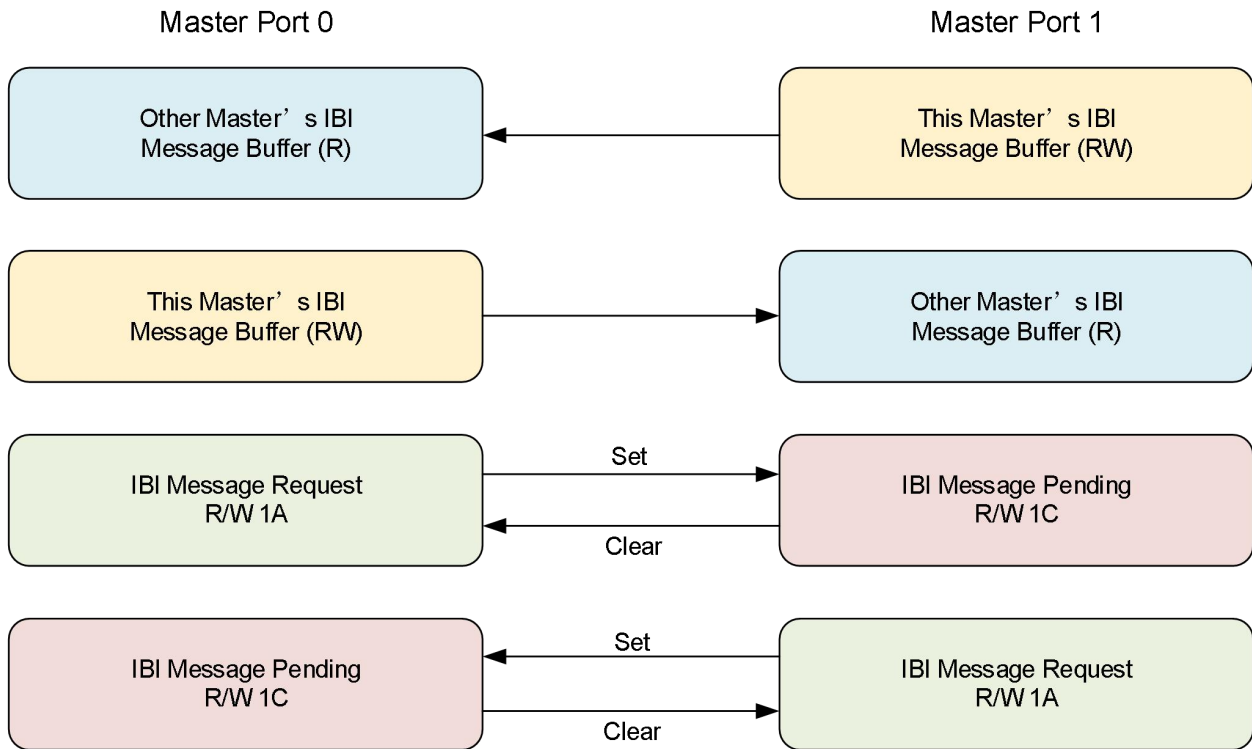
Page	Master 0/1 Sharing	Description
0	Shared	Master Side Port Scratch Board buffer 0, 128 bytes, R/W
1	Shared	Master Side Port Scratch Board buffer 1, 128 bytes, R/W
2	Shared	Reserved for Future Use
3	Shared	Reserved for Future Use
4	Referencing/ Dedicated	The other master's Reg0-127 (directly mapped registers) Access. Directly Mapped Address "0" of other port maps to Address "128" on the Page4 of this port. For M1, Referenced M0 registers 0-127 are read only. For M0, Referenced M1 registers 0-127 are readable, and writes to those registers if the M0 REG16 configuration password register was written with "0x6A".
5-15		Reserved
16	Shared	Slave Side Port 0 Transmitting Transaction buffer 0, 88 bytes, R/W If the REG#19[1] bit is set, only the selected master can write to the buffer.
17	Shared	Slave Side Port 0 Received Transaction local Slave address list, eight bytes, R/W. If the REG#19[1] bit is set, only the selected master can write to the buffer.
18	Shared	Slave Side Port 0 Received Transaction buffer 0, 80 bytes, Read Only
19	Shared	Slave Side Port 0 Received Transaction buffer 1, 80 bytes, Read Only
20-47	Shared	Slave Side Port 1-7 Transmit/Receive buffers. Same format as 16-19 as listed above. 20-23 = Slave Side Port 1. 24-27 = Slave Side Port 2. 28-31 = Slave Side Port 3. 32-35 = Slave Side Port 4. 36-39 = Slave Side Port 5. 40-43 = Slave Side Port 6. 44-47 = Slave Side Port 7. If the REG#19[1] bit is set, only the selected master can write to the R/W buffers.
48-127	Shared	Reserved
128-255	Shared	Vendor Registers



**Figure21.Paged Buffer Mapping**

### 10.2.4 Master Side Port Message Registers

Mapping of Register Values between two masters' messaging registers:



**Figure22.Master-Master Messaging Buffer Mapping and Interaction**

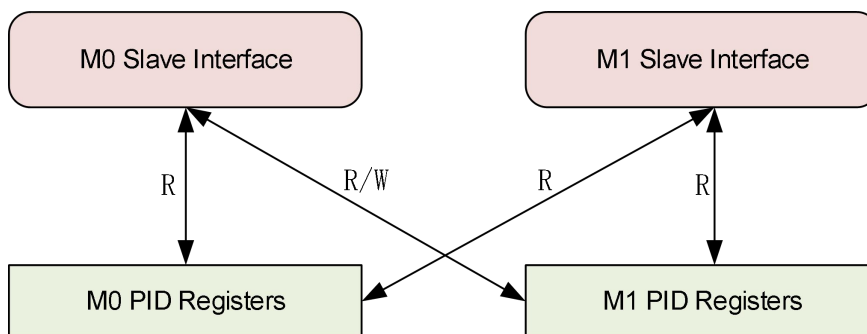
### 10.2.5 Device Level Registers

The Device Level Control Registers and Vendor Registers are shared by the on-chip slave interfaces. They are the same registers that can be accessed from either interface.

The scratch-board registers are shared registers, and designed for master-master communication usage. The detailed usage model of these registers is up to the system-level firmware designer.

### 10.2.6 PID Registers

The M1 PID registers are write-able only by M0 port via the paged register space window. Writing to the PID registers is password-protected.



**Figure23.PID Register Access**

### 10.3 Register Definition Details

Note: The “default” value defined in this section is the value that the register holds after power up, if not otherwise explicitly defined in 5.4 Part Number Specific Configurations section.

Reserved register bit fields should be written with 0 in a write transaction. Data reading from reserved bit fields are not defined and should be ignored.

#### 10.3.1 Device Information Registers

Address:	0	Register:	Device Info Byte 0 (M0/M1 Shared - Per Device)
Bits	Attribute	Default	Description
7:0	R	See Table 26	DEV_INFO0 TBD –This register returns the first byte of the device identification code.

Address:	1	Register:	Device Info Byte 1 (M0/M1 Shared - Per Device)
Bits	Attribute	Default	Description
7:0	R	0x12	DEV_INFO1 TBD - The Register returns the second byte of the device identification code.

Address:	2	Register:	PID (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:0	M0:RE M1: ROWP	Random	PID_Byte0, PID[7:0], Value TBD. The Register in Master Side Port 1 can be Written by Master Side Port 0 Slave via paged access.

Address:	3	Register:	PID (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:0	M0:RE M1: ROWP	Random	PID_Byte1, PID[15:8], Value TBD. The Register in Master Side Port 1 can be Written by Master Side Port 0 Slave via paged access.

Address:	4	Register:	PID (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:0	M0:RE M1: ROWP	Random	PID_Byte2, PID[23:16], Value TBD. The Register in Master Side Port 1 can be written by the Master Side Port 0 Slave via paged access.

Address:	5	Register:	PID (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:0	M0:RE M1: ROWP	Random	PID_Byte3, PID[31:24], Value TBD. The Register in Master Side Port 1 can be written by the Master Side Port 0 Slave via paged access.

Address:	6	Register:	PID (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:0	M0:RE M1: ROWP	0xcd	PID_Byte4, PID[39:32], Value TBD. The Register in Master Side Port 1 can be written by the Master Side Port 0 Slave via paged access.

Address:	7	Register:	PID (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:0	M0:RE M1: ROWP	0x04	PID_Byte5, PID[47:40], Value TBD. The Register in Master Side Port 1 can be written by the Master Side Port 0 Slave via paged access.

Address:	8	Register:	BCR (M0/M1 Shared - Per Device)
Bits	Attribute	Default	Description
7:0	RWP	0x16	BCR (see PJ71MXX BCR) I3C BCR

Address:	9	Register:	DCR (M0/M1 Shared – Per Device)
Bits	Attribute	Default	Description
7:0	RWP	0Xc2	DCR (see PJ71MXX BCR) I3C DCR

Address:	10	Register:	Device Capability (M0/M1 Shared - Per Device)
Bits	Attribute	Default	Description
7:4	R	0x0	Reserved
3	R	See table 26	Eight_Port The register reflects four or eight downstream port of the device. 0: Four Slave Side Ports supported 1: Eight Slave Side Ports supported
2	R	0x1	Analog_Switch_Capability The register reflects the Analog Switch capability of the device. 0: Feature not supported 1: Feature supported
1	R	0x1	BusAgent_Capability The register reflects the SMBus Agent capability of the device. 0: Feature not supported 1: Feature supported
0	R	0x1	GPIO_Capability The register reflects the GPIO capability of the device. 0: Feature not supported 1: Feature supported

Address:	11	Register:	Device Revision (M0/M1 Shared - Per Device)
Bits	Attribute	Default	Description
7:0	R	0x0X	Device_Revision The register reflects the revision of the device. 0x00 for PJ71M47/8B and PJ71M87/8B 0x01 for PJ71M49/89B

### 10.3.2 Device Configuration Registers

Configuration Registers programming needs the protection code register properly set.

Address:	16	Register:	Unlock Device Configuration Protection Code (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:0	RW	0x00	Protection_Code The protection code unlocks the Configuration Registers (range from address 17 to address 31), the PID register of the other port (register address from 2 to 13), and other registers with an attribute "P" (address range below 128). This register must be written to 0x69 to allow the protected registers get programmed. Reading from the configuration registers is not subject to the control of the password register.

Address:	17	Register:	Master Side Port Configuration (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7	RWP	0x0	Reserved
6	RWP	0x0	<b>BL_Enable</b> Effective when PEC is not enabled. If set, this bit defines a Burst Length byte in the I3C write transaction after the Address Point byte. 0 = Disable 1 = Enabled This bit field is cleared by SETBUSCON CCC with JEDEC Context.
5	RWP	0x0	<b>IBI_MDB_Enable</b> 0 = No extra IBI MDB inserted after the address byte. 1 = A MDB byte will be inserted after the address byte. This bit field is set by SETBUSCON CCC with JEDEC Context.
4	RWP	See table 26	<b>MPort_OD_Only</b> Master Interfacing Port OD Only Mode. This bit controls the Master Interfacing Port operation mode. 1 = OD only mode, only I2C Open Drain operation is supported. 0 = OD/PP compatible mode. Both I2C and I3C operation supported. If MPort_OD_Manual (Bit[3]) = 0, this bit is cleared automatically when a CCC or 0x7E address is received.
3	RWP	0x0	<b>MPort_OD_Manual</b> Master Interfacing Port OD Only Register (bit[4] of this register) control. This bit controls the Master Interfacing Port OD_Only bit Control mode. This bit can only be updated once after power up or reset. 1 = OD_Only bit is changed manually by explicitly register access. 0 = OD_Only bit is cleared automatically when a CCC or 0x7E address is received.
2	RWP	0x0	<b>Bus_Reset_Enable</b> This bit can also get updated with SETBUSCON CCC, regardless of the protection code register setting. 0 = Feature Disabled 1 = Feature Enabled This bit field is set by SETBUSCON CCC with JEDEC Context. This bit field is cleared by SETBUSCON CCC with any non-JEDEC Context.
1	RWP	0x0	<b>I3C_Parity_Check_Disable</b> 0 = Enable 1 = Disable In the JEDEC Context, This bit is set/clear when DEVCTRL CCC is received in JEDEC Context, as per Byte0[6] value in CCC data payload byte; and Reset by a BUS Reset Event. This bit field is cleared by SETBUSCON CCC with any non-JEDEC Context.
0	RWP	0x0	<b>PEC_Fun_Enable</b> PEC Function Enable 0 = Disable 1 = Enable In the JEDEC Context, This bit is set/clear when DEVCTRL CCC is received in JEDEC Context, as per Byte0[7] value in CCC data payload byte; and reset by a BUS Reset Event. This bit field is cleared by SETBUSCON CCC with any non-JEDEC Context.

Address:	18	Register:	Slave Side Port Enable (M0/M1 shared - Per Device)
Bits	Attribute	Default	Description
7:0	RWP	See table 26	Slave_Ports_Enable Bit[7] 0 = Slave Side Port [7] Disabled. 1 = Slave Side Port [7] Enabled Bit[6] 0 = Slave Side Port [6] Disabled. 1 = Slave Side Port [6] Enabled ... Bit[n] 0 = Slave Side Port [n] Disabled. 1 = Slave Side Port [n] Enabled ... Bit[0] 0 = Slave Side Port [0] Disabled. 1 = Slave Side Port [0] Enabled Disabled Port stays HiZ, and is 3.3V tolerate. These bit fields are set by SETBUSCON with JEDEC Context. If REG#19[3] is set, the Port Enable bits are effective when corresponding Port's VIO supply (See Pin Descriptions) is in Power Good Status (See REG#121).

Address:	19	Register:	Hub Device Configuration (M0/M1 shared - Per Device)
Bits	Attribute	Default	Description
7	RWP	0x0	SETBUSCON_EN This bit enables SETBUSCON CCC's setting of configuration registers. 0: SETBUSCON CCC will not change the value of the configuration registers. 1: SETBUSCON CCC will update the value of the configuration registers if stated in the register description. This bit is only write-able by port M0.
6	RWP	0x0	M0_I2C_Addr_Strap 0: Not enabled 1: Feature enabled When this feature is enabled, and when "M1_On_Chip_Slave_Disable" bit is set, the M0 I2C Address' LSB 2bits (either the default as defined in 4.2.3.1, or defined by REG#126) can be pin-strapped from MSDA1(as address bit1) and MSCL1 (as address bit0). If "M1_SDA_As_Alert_n" is set, the MSDA1 will not be used in strapping and bit1 of the address is not updated. The pin strapping is real-timing and the Hub device does not latch the pin strapping values. This allows the flexibility of the master to expand the addressable Hub devices by dynamically controlling the strapping pin. In the pin-strapping mode, the Hub Network will select M0 regardless the MSEL Pin state and Register Selection settings with REG#56. This bit is only write-able by port M0.
5	RWP	0x0	M1_SDA_As_Alert_n If both "M1_On_Chip_Slave_Disable" bit and this bit is set, MSDA1 pin will be used as Alert_n pin of On-Chip Slave under Master Side Port 0. the Hub Network will select M0 regardless the MSEL Pin state and Register Selection settings with REG#56. In this mode, if the "ibi_pending" status is set, Alert_n pin will be pull low in an open- drain fashion. if the "ibi_pending" status is not set, the Alert_n pin will be left floating. This bit is only write-able by port M0.
4	RWP	0x0	M1_On_Chip_Slave_Disable When set, this bit disable the on-chip slave associated with Master Side Port 1. The Master Side Port 1 is still selected if MSEL pin driven high, but the on-chip slave is not visible and will not respond to bus activities. This bit is only write-able by port M0.

Address:	19	Register:	Hub Device Configuration (M0/M1 shared - Per Device)
Bits	Attribute	Default	Description
3	RWP	0x1	<p>SSPORT_As_Disabled_If_VIOS_Not_Good            Enable the feature of treat the Slave Side Port as Disabled if the associated VIOS supply is not good, even the port is configured as enabled.            0: Feature Disabled            1: Feature Enabled            This bit is only write-able by port M0.</p>
2	RW	0x1	<p>MSEL_RESET_EN            Enable using MSEL pulses to reset the device. If enabled, the I3C Hub device will reset the whole device when 4 MSEL level transitions that received within tMSEL_DET time. The period of the clock pulses should be lower than 1MHz.            0: Feature Disabled.            1: Feature Enabled.            If the bit is set, user should not toggle the MSEL pin too frequent and observe the Reset pulse interval of tMSEL_DET            This bit is only write-able by port M0.</p>
1	RWP	0x0	<p>Device_Reg_Always_Selected            0 = Device Level Resources (like Slave Side Port and associated GPIO/Bus Agents) are shared and accessed by the two masters. Hub Network connected Ports are controlled by selected master port.            1 = Device Level Resources will be only controllable/Writeable by selected master.            This bit is supposed to be a configuration bit and is not changed on the fly during operation. This bit is only writeable by port M0.            The Registers that controlled by this bit are listed in Two On-Chip Slaves' Register Space and Sharing/Selection Map.            This bit field is cleared by M0 SETBUSCON CCC with JEDEC Context.            This bit is only write-able by port M0.</p>
0	RWP	0x0	<p>Master_Port_Mux_Priority            Master Side Port Multiplexer Priority:            0 = M0 Priority            1 = M1 Priority            This bit is supposed to be a configuration bit and is not changed on the fly during operation. <b>This bit is only writeable by port M0.</b>            This bit field is cleared by M0 SETBUSCON CCC with JEDEC Context.            This bit is only write-able by port M0.</p>

Address:	20	Register:	IO Drive Strength Configuration (M0/M1 shared - Per Device)
Bits	Attribute	Default	Description
7:6	RWP	0x0	M1_IO_ROM MSDA[1] Driver Impedance Selection 00 = ~20 ohm 01 = ~30 ohm 10 = ~40 ohm 11 = ~50 ohm This bit field is set to "01" by SETBUSCON CCC with JEDEC Context.
5:4	RWP	0x0	M0_IO_ROM MSDA[0] Driver Impedance Selection 00 = ~20 ohm 01 = ~30 ohm 10 = ~40 ohm 11 = ~50 ohm This bit field is set to "01" by SETBUSCON CCC with JEDEC Context.
3:2	RWP	See table 26	SSPorts_ROM_VIOS1 SLCL[2/3/6/7], SLDA[2/3/6/7] Driver Impedance Selection 00 = ~20 ohm 01 = ~30 ohm 10 = ~40 ohm 11 = ~50 ohm This bit field is set to "01" by SETBUSCON CCC with JEDEC Context.
1:0	RWP	See table 26	SSPorts_ROM_VIOS0 SLCL[0/1/4/5], SLDA[0/1/4/5] Driver Impedance Selection 00 = ~20 ohm 01 = ~30 ohm 10 = ~40 ohm 11 = ~50 ohm This bit field is set to "01" by SETBUSCON CCC with JEDEC Context.

Address:	21	Register:	Hub Network Operation Mode (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:6	RWP	0x0	Reserved
5	RWP	See table 26	<p>HubNetwork_Always_I3C            I3C Hub Network Traffic Protocol Setting            0 = Use bit[2] of this register to determine the Hub Network frame protocol            1 = Assume only I3C protocol and traffic will pass the Hub Network. This bit will be cleared by SETBUSCON with JEDEC Context.</p>
4	RWP	0x0	<p>HubNetwork_Fixed_Analog_Switch_Mode            Hub Network Operation Mode:            0 = I2C/I3C Hub. The default function of this device.            1 = Fixed as Analog Switch Only. In this mode, the enabled and connected Slave Side Ports (defined by REG#81) are connected to the selected Master Side Port through an analog switch, and the disabled Slave Side Port will be disconnected. This bit is cleared by SETBUSCON with JEDEC Context.            This bit has higher priority than the Slave Side Port GPIO Enable setting, i.e. once enabled, the Slave Side Port cannot be used as GPIO, connecting to SMBus Agent or connect to a Hub Network.            If configured, the connection through Fixed Analog Switch Mode is built when both selected Master Side Port and Connected Slave Side Port are in "Power Good" (See Reg#121) Status, and same voltage setting (see REG#22). The Analog Switch connection is disconnected if any side of the connected ports' VIO has a cleared Power Good status (i.e. Power go out of nominal range). The Hub device rebuilds the connection if the Power Good Status restore the set status.</p>
3	RWP	0x0	<p>HubNetwork_IBI_Arbit_Opt_En            Hub Network Generic I3C Mode IBI arbitration optimization Enable:            0 = No I3C address arbitration optimization            1 = I3C dynamic address assigned between 7'h03 to 7'h3F. If A6 is "1" in the open drain header detected, the remaining bits can be transferred in Push/pull. Refer to I3C Basic specification section regarding "I3C Address Arbitration Optimization", for further details            This bit is cleared by SETBUSCON with JEDEC Context.</p>
2	RWP	0x0	<p>HubNetwork_Frame_Protocol            I3C Hub Network Traffic Protocol Setting            0 = Same as selected Master Side Port. If Master Side Port has Dynamic Address Assigned, assume all traffic is I3C            1 = Use the Dynamic Address Assignment table. Check each device address transaction and determine the transaction protocol.            This bit will be cleared by SETBUSCON with JEDEC Context.</p>
1	RWP	0x0	<p>Reserved            This bit is reserved and should be written as 0.</p>
0	RWP	0x0	<p>HubNetwork_PP_Analog_Switch_En            I3C Hub Network Push/Pull Mode Dynamic Analog Switch Enable            0 = Disabled. Analog Switch Path not enabled, target port is in re-driving mode.            1 = Enabled. Analog Switch Path is enabled for connected target port, dynamically in a read transaction.            This bit is set by SETBUSCON CCC with JEDEC Context.            This bit is effective when both selected Master Side Port and target Slave Side Port are in "Power Good" (See Reg#121) Status, and same voltage setting (see REG#22).</p>

Address:	22	Register:	Interface Voltage LDO Setting (M0/M1 Shared - Per Device)
Bits	Attribute	Default	Description
7:6	RWP	See table 26	VIOS1_LDO_Voltage VIOS1 LDO Voltage, or nominal applied platform voltage (For Slave Side Port 2/3/6/7) 00 = 1.0V 01 = 1.1V 10 = 1.2V 11 = 1.8V This bit field is set to "00" by SETBUSCON with JEDEC Context.
5:4	RWP	See table 26	VIOS0_LDO_Voltage VIOS0 LDO Voltage, or nominal applied platform voltage (For Slave Side Port 0/1/4/5) 00 = 1.0V 01 = 1.1V 10 = 1.2V 11 = 1.8V This bit field is set to "00" by SETBUSCON with JEDEC Context.
3:2	RWP	See table 26	VIOM1_LDO_Voltage VIOM1 LDO Voltage, or nominal applied platform voltage (for Master Side Port 1) 00 = 1.0V 01 = 1.1V 10 = 1.2V 11 = 1.8V This bit field is set to "00" by SETBUSCON with JEDEC Context.
1:0	RWP	See table 26	VIOM0_LDO_Voltage VIOM0 LDO Voltage, or nominal applied platform voltage (for Master Side Port 0) 00 = 1.0V 01 = 1.1V 10 = 1.2V 11 = 1.8V This bit field is set to "00" by SETBUSCON with JEDEC Context.

Address:	23	Register:	Slave Side Port IO OD Only Signaling Mode Setting (M0/M1 shared - Per Device)
Bits	Attribute	Default	Description
7:0	RWP	0x00	SSPorts_OD_Only Slave Interfacing Port IO Signaling Mode Configuration. Bit[0] Maps to Slave Interfacing Port [0]. Bit[1] Maps to Slave Interfacing Port [1]. Bit[n] Maps to Slave Interfacing Port [n]. (n is 0-7) For Any port n: Bit[n] = 1: Port is Open-Drain Only (SMBus or I2C). IO is 3.3V tolerate. Bit[n] = 0: Port is Open-Drain / Push-Pull compatible. The Push/Pull voltage and pull up voltage will be defined by the VIOS0 or VIOS1 Voltage Configuration. These bit fields are cleared by SETBUSCON CCC with JEDEC Context.

Address:	24	Register:	Slave Side Port SMBus Agent Enable (M0/M1 shared - Per Device)
Bits	Attribute	Default	Description
7:0	RWP	0x00	SSPorts_Agent_Enable Slave Side Port SMBus Agent Enable. These bits are only valid when the Slave Interfacing Port is disconnected from the Hub Network. Bit[n] maps to Slave Side Port n Control, i.e. Bit[0] Controls Port 0. For any Bit[n]: 0 = The SMBus Agent is disabled. 1 = The SMBus Agent is enabled. This bit field is cleared by SETBUSCON CCC with JEDEC Context.

Address:	25	Register:	LDO Enable and Pull-up Resistor Value Register (M0/M1 Shared - Per Device)
Bits	Attribute	Default	Description
7:6	RWP	See table 26	SSPORTS_SDA_Pullup_Res0 For Slave Side Ports that powered by VIOS0, SSDA/SSCL pull up resistance configuration 00 = ~250 ohm 01 = ~500 ohm 10 = ~1K ohm 11 = ~2K ohm This bit field is set to "01" by SETBUSCON CCC with JEDEC Context.
5:4	RWP	See table 26	SSPORTS_SDA_Pullup_Res1 For Slave Side Ports that powered by VIOS1, SSDA/SSCL pull up resistance configuration 00 = ~250 ohm 01 = ~500 ohm 10 = ~1K ohm 11 = ~2K ohm This bit field is set to "01" by SETBUSCON CCC with JEDEC Context.
3	RWP	0x0	VIOS1_LDO_Enable 0 = Disable 1 = Enable
2	RWP	0x0	VIOS0_LDO_Enable 0 = Disable 1 = Enable
1	RWP	0x0	VIOM1_LDO_Enable 0 = Disable 1 = Enable
0	RWP	0x0	VIOM0_LDO_Enable 0 = Disable 1 = Enable

Address:	26	Register:	Master Side Port IBI Generation Enable (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:3	RWP	0x0	Reserved
2	RWP	0x0	Master_Msg_Req_IBI_EN Other Master Message Request IBI Generation Enable 0 = Don't generate IBI when the other master sets the message request flag. 1 = Generate IBI when the other master sets the message request flag.
1	RWP	0x0	Parity_Err_IBI_EN Parity Error IBI Generation Enable 0 = Disable 1 = Enable
0	RWP	0x0	PEC_Err_IBI_EN PEC Error IBI Generation Enable 0 = Disable 1 = Enable

Address:	27	Register:	Slave Side Port SMBus Agent IBI Enable (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:0	RWP	0x00	SSPorts_Agent_IBI_EN Slave Side Port SMBus Agent IBI Enable. Each Master Interfacing Port has a dedicated configuration of this register. Bit[n] maps to Slave Side Port n Control, i.e. Bit[0] Controls Port 0. For any Bit[n]: 0 = The SMBus Event Flags (see Slave_Agent_Buf_Overflow_Flag) do not trigger IBI generation. 1 = The SMBus Event Flags, if set, triggers an IBI generation. This bit field is cleared by SETBUSCON CCC with JEDEC Context.

Address:	28	Register:	Customized IBI MDB Value (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:0	RWP	0x00	IBI_MDB An user specified IBI Mandatory Data Byte with value set by this byte, if enabled (e.g. as in the JEDEC Context), is inserted to the IBI payload bytes before other bytes.

Address:	29	Register:	JEDEC Context ID (Shared by two Master Side Ports)
Bits	Attribute	Default	Description
7:0	RWP	0x80	JEDEC_BUSCON_Code SETBUSCON Context Byte Value for JEDEC. If the SETBUSCON CCC Context Byte matches this register, the Context will be set to JEDEC.

Address:	30	Register:	Slave Side Port GPIO Mode Enable (M0/M1 Shared - Per Device)
Bits	Attribute	Default	Description
7:0	RWP	0x00	SSPorts_GPIO_EN Slave Side Port GPIO Mode Enable. This bit has higher priority to the other configuration bits (Hub disconnect, Port Enable, SMBus Agent enable). Bit[n] maps to Slave Side Port n Control, i.e. Bit[0] Controls Port 0. For any Bit[n]: 0 = The GPIO mode of the port IO is disabled. 1 = The GPIO mode of the port IO is enabled. This bit field is cleared by SETBUSCON CCC with JEDEC Context.

### 10.3.3 Device Status and IBI Related Registers

#### 10.3.3.1 Device and IBI Status Registers

Address:	32	Register:	Device and Port Status (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7	R	0x0	IBI_Pending This bit is set when there are one or more new event flag were set, and cleared when an IBI was generated successfully to the master.
6	R	0x0	Reserved
5	R	0x0	IO_Cal_Ongoing This bit is set when the IO drive-strength calibration is in progress. It is cleared when calibration finishes. This is a device level status.
4	R	0x0	SMBUS_Agent_Event_Flag_Status SMBus Agent Flag status, = logic OR of all bits in REG#33 0 = No Flags set 1 = Flags set This bit is cleared when the associated Flag was cleared.
3	R	0x0	SSPort_IO_Flag_Status Aggregated SSPort IO event flag. This bit is a logic OR of all bits in REG#98 and REG#99, if the corresponding event is enabled by register REG#89 and REG#90. 0 = No Flags set 1 = Flags set
2	R/W1C	0x0	Master_Msg_Pending_Flag Other Master Message Request Pending Flag 0 = No Message 1 = Message Pending This bit is cleared when DEVCTRL CCC is received in JEDEC Context, and IBI Global clear bit is set in Byte1[3]
1	R/W1C	0x0	Parity_Error_Flag

			0 = No Error 1 = Error Detected This bit is cleared when DEVCTRL CCC is received in JEDEC Context and IBI Global clear bit is set in Byte1[3], or a Bus Reset is received
0	R/W1C	0x0	PEC_Error_Flag 0 = No Error 1 = Error Detected This bit is cleared when DEVCTRL CCC is received in JEDEC Context and IBI Global clear bit is set in Byte1[3], or a Bus Reset is received

Address:	33	Register:	Slave Side Port SMBus Agent IBI Status (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:0	R	0x0	SSPorts_Agent_Flag_Status For each bit[n]: Slave Side Port [n] SMBus Agent IBI flag status. This bit is a combinational logic aggregation of agent flags associated with Slave Side Port[n]. The bit is set when any of the flags in REG#100 in SSPort[n] is set and associated bit in REG#27 is set, is cleared when all flags are cleared or when that port's Bus Agent IBI Enable bit is cleared.

### 10.3.3.2 Master-Master Messaging: Message from Other Master Registers

Address:	34	Register:	Other Master Message Byte 0 (Reference to other master's message buffer)
Bits	Attribute	Default	Description
7:0	R	0x0	In_MSG_Byte0 Message byte 0 from other Master.

Address:	35	Register:	Other Master Message Byte 1 (Reference to other master's message buffer)
Bits	Attribute	Default	Description
7:0	R	0x0	In_MSG_Byte1 Message byte 1 from other Master.

Address:	36	Register:	Other Master Message Byte 2 (Reference to other master's message buffer)
Bits	Attribute	Default	Description
7:0	R	0x0	In_MSG_Byte2 Message byte 2 from other Master.

Address:	37	Register:	Other Master Message Byte 3 (Reference to other master's message buffer)
Bits	Attribute	Default	Description
7:0	R	0x0	In_MSG_Byte3 Message byte 3 from other Master.

### 10.3.3.3 Master-Master Messaging: Message to Other Master Registers

Address:	40	Register:	Master Message Byte 0 (Sent to other master's message)
Bits	Attribute	Default	Description
7:0	RW	0x0	Out_Msg_Byte0 Message byte 0 to other Master.

Address:	41	Register:	Master Message Byte 1 (Sent to other master's message)
Bits	Attribute	Default	Description
7:0	RW	0x0	Out_Msg_Byte1 Message byte 1 to other Master.

Address:	42	Register:	Master Message Byte 2 (Sent to other master's message)
Bits	Attribute	Default	Description
7:0	RW	0x0	Out_Msg_Byte2 Message byte 2 to other Master.

Address:	43	Register:	Master Message Byte 3 (Sent to other master's message)
Bits	Attribute	Default	Description
7:0	RW	0x0	Out_Msg_Byte3 Message byte 3 to other Master.

Address:	44	Register:	Master Message Request (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:2	B	0x0	Reserved.
1	RW1S	0x0	Other_Master_HJ_Req Other Master Hot Join Request. Writing 1 to this bit sets the other master's Hot Join procedure if it does not has a Dynamic Address. This bit is self-cleared after the other master's Hot Join procedure was sent, or when the HJ_Enabled is cleared, or the other master already has a Dynamic Address.
0	RW1S	0x0	Master_Msg_Req Master Message Request. Writing 1 to this bit sets the other master's Message Request Flag. This bit is aliasing to the other master's IBI messaging request status flag. When the other master clears the Master_Msg_Pending_Flag this bit will be cleared as well.

### 10.3.4 Master-Master Generic Scratch Board Registers

These registers are Read/Write Accessible by both masters. The two masters utilize these registers to develop an application messaging system.

Address:	48	Register:	Common Message Buffer Byte 0 (M0/M1 Shared- Per Device)
Bits	Attribute	Default	Description
7:0	RW	0x0	Scratch_Byte0 Shared Master-Master Communication Message byte 0.

Address:	49	Register:	Common Message Buffer Byte 1 (M0/M1 Shared- Per Device)
Bits	Attribute	Default	Description
7:0	RW	0x0	Scratch_Byte1 Shared Master-Master Communication Message byte 1.

Address:	50	Register:	Common Message Buffer Byte 2 (M0/M1 Shared- Per Device)
Bits	Attribute	Default	Description
7:0	RW	0x0	Scratch_Byte2 Shared Master-Master Communication Message byte 2.

Address:	51	Register:	Common Message Buffer Byte 3 (M0/M1 Shared- Per Device)
Bits	Attribute	Default	Description
7:0	RW	0x0	Scratch_Byte3 Shared Master-Master Communication Message byte 3.

### 10.3.5 Master Multiplexer Selection Control and Status Registers

Address:	56	Register:	Master Multiplexer Selection Control (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:1	B	0x0	Reserved
0	RW	0x0	MSPort_Mux_Req Requests Hub Control 0 = Relinquishes Hub Control 1 = Requests Hub Control

Address:	57	Register:	Master Multiplexer Selection Status (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:1	B	0x0	Reserved
0	R	0x0	MUX_MSPort_Selected Multiplexer Selection Control Status. Reports the status of the master selection Hub. 0 = This Master Side Port is not connected to the Hub Network 1 = This Master Side Port is connected to the Hub Network

### 10.3.6 Slave Dynamic Address Assignment Flag Registers

There are 128 bit fields, map to slave address 0-127. Each bit field indicates if the associated slave address is an assigned I3C dynamic address or not.

Address:	64 65 66 67 68 69 70 71	Register:	Slave Dynamic Address Assigned Flag Byte0 (M0/M1 Shared- Per Device) Slave Dynamic Address Assigned Flag Byte1 (M0/M1 Shared- Per Device) Slave Dynamic Address Assigned Flag Byte2 (M0/M1 Shared- Per Device) Slave Dynamic Address Assigned Flag Byte3 (M0/M1 Shared- Per Device) Slave Dynamic Address Assigned Flag Byte4 (M0/M1 Shared- Per Device) Slave Dynamic Address Assigned Flag Byte5 (M0/M1 Shared- Per Device) Slave Dynamic Address Assigned Flag Byte6 (M0/M1 Shared- Per Device) Slave Dynamic Address Assigned Flag Byte7 (M0/M1 Shared- Per Device)
Bits	Attribute	Default	Description
7:0	RW	0x0	DA_Status Slave Address at (Byte# * 8 + Bit#) dynamic address assignment status. 0 = Hub Network treat this address as a I2C Static Address 1 = Hub Network treat this address as a I3C Dynamic Address Byte# Register mapping: 0 for REG64, 1 for REG 65...15 for REG79. Slave Address corresponding byte # = SlaveAddr[6:3] (Address Byte[7:4]) Slave Address corresponding bit # = SlaveAddr[2:0] (Address Byte[3:1]) Example: Address 0x13 corresponds byte # is 2, bit# is 3, and maps to REG66[3]. If the Address is a dynamic address, REG66[3] should be set.

<b>Address:</b>	72 73 74 75 76 77 78 79	<b>Register:</b>	<b>Slave Dynamic Address Assigned Flag Byte8 (M0/M1 Shared- Per Device)</b> <b>Slave Dynamic Address Assigned Flag Byte9 (M0/M1 Shared- Per Device)</b> <b>Slave Dynamic Address Assigned Flag Byte10 (M0/M1 Shared- Per Device)</b> <b>Slave Dynamic Address Assigned Flag Byte11 (M0/M1 Shared- Per Device)</b> <b>Slave Dynamic Address Assigned Flag Byte12 (M0/M1 Shared- Per Device)</b> <b>Slave Dynamic Address Assigned Flag Byte13 (M0/M1 Shared- Per Device)</b> <b>Slave Dynamic Address Assigned Flag Byte14 (M0/M1 Shared- Per Device)</b> <b>Slave Dynamic Address Assigned Flag Byte15 (M0/M1 Shared- Per Device)</b>
<b>Bits</b>	<b>Attribute</b>	<b>Default</b>	<b>Description</b>
7:0	RW	0x0	<p>DA_Status Slave Address at (Byte# * 8 + Bit#) dynamic address assignment status. 0 = Hub Network treat this address as a I2C Static Address 1 = Hub Network treat this address as a I3C Dynamic Address Byte# Register mapping: 0 for REG64, 1 for REG 65...15 for REG79. Slave Address corresponding byte # = SlaveAddr[6:3] (Address Byte[7:4]) Slave Address corresponding bit # = SlaveAddr[2:0] (Address Byte[3:1]) Example: Address 0x13 corresponds byte # is 2, bit# is 3, and maps to REG66[3]. If the Address is a dynamic address, REG66[3] should be set.</p>

### 10.3.7 Slave Side Port Control Registers

<b>Address:</b>	80	<b>Register:</b>	<b>Slave Interfacing Port Master Agent Transaction Request (M0/M1 Shared - Per Device)</b>
<b>Bits</b>	<b>Attribute</b>	<b>Default</b>	<b>Description</b>
7:0	RW1S	0x0	<p>Agent_Transaction_Req Master Agent Transaction Request. Bit[n] maps to Slave Side Port n Control, i.e. Bit[0] Controls Port 0. For any Bit[n]: Writing 1 to this bit starts a Slave Side Port n Master Agent Transaction. The Master Agent transaction's descriptor and data should be programmed to the associated data buffer before this bit is set. This bit is self-cleared after the transaction is finished or aborted. If the Bus Agent is not enabled (Agent enable bit not set in REG#24, or GPIO is enabled in REG#30, or Port disabled in REG#18) the request will be ignored and no action taken; and the bit is cleared immediately.</p>

<b>Address:</b>	81	<b>Register:</b>	<b>Slave Side Port Hub Connection Control (M0/M1 Shared - Per Device)</b>
<b>Bits</b>	<b>Attribute</b>	<b>Default</b>	<b>Description</b>
7:0	RW	See table 26	<p>SSPorts_HubNetwork_Connection Slave Side Port Connection to Hub Network Setting. Bit[n] maps to Slave Side Port n Control, i.e. Bit[0] Controls Port 0. For any Bit[n]: 1 = Hub Connection to Slave Side Port[n] is enabled. 0 = Hub Connection to Slave Side Port[n] is disabled. User can only use the SMBus Agent to access the slaves behind this Slave Side Port. Note: If Slave Side Port IO is in OD_Only Mode and the selected Master is not in OD_Only mode, the configuration will not work and should be avoided. In this case, the OD_Only Slave Side Port connection setting is not be built.</p>

<b>Address:</b>	82	<b>Register:</b>	<b>Master/Slave Side Port Voltage Detection (M0/M1 Shared - Per Device)</b>
<b>Bits</b>	<b>Attribute</b>	<b>Default</b>	<b>Description</b>
7	R	0x0	<p>SDA_DET_CMP_Status Slave Side Port SDA voltage detection Result. 0 = The Bit[3:0] Selected SDA voltage is lower than the Bit[4] defined threshold. 1 = The Bit[3:0] Selected SDA voltage is higher than the Bit[4] defined threshold.</p>

6	RW	0x0	Reserved
5	RW	0x0	SDA_DET_CMP_EN Port SDA voltage detection comparator Enable. 0 = Disabled 1 = Enabled. The Bit[3:0] Selected Slave Side Port's SDA voltage is compared with the threshold set by Bit[3]. The result is readable at Bit[7]

Address:	82	Register:	Master/Slave Side Port Voltage Detection (M0/M1 Shared - Per Device)
Bits	Attribute	Default	Description
4	RW	0x0	SDA_DET_CMP_VTH Port SDA voltage detection threshold voltage reference selection. 0 = 2.1V 1 = 1.5V
3:0	RW	0x0	SDA_DET_CH_Sel Slave Interfacing Port SDA voltage detection Selection. Bit[3:0]: Channel table: 0: MSDA1 1: MSDA0 2: SSDA7 3: SSDA6 4: SSDA5 5: SSDA4 6: SSDA3 7: SSDA2 8: SSDA1 9: SSDA0 Other: GND

Address:	83	Register:	Slave Side Port Pull up Resistor Connection Control (M0/M1 shared - Per Device)
Bits	Attribute	Default	Description
7:0	RWP	0xFF	SSPorts_Pullup_EN Slave Side Port built-in Pull-Up resistor Enable. Pull up resistor connects to the VIO rail, and is only available when the Slave Side Port Operates in the OD/PP compatible mode, and connects to the VIN rail, when the Slave Side Port Operates in the OD Only mode (see Figure 18). Bit[n] maps to Slave Side Port n Control, i.e. Bit[0] Controls Port 0. For any Bit[n]: 0 = Pull up Resistor not connected to the Port. 1 = Pull up Resistor connected to this Slave Side Port. This bit field is set by SETBUSCON CCC with JEDEC Context. This bit field is independent of the GPIO mode control.

Address:	84	Register:	Slave Side Port GPIO Mode SCL Output Enable (M0/M1 shared - Per Device)
Bits	Attribute	Default	Description
7:0	RW	0x00	GPIO_SSCL_OUT_EN Slave Side Port GPIO Mode IO output Enable. Bit[n] maps to Slave Side Port n Control, i.e. Bit[0] Controls Port 0 SSCL pin(GPIO[n]0). For any Bit[n]: 0 = IO is not in driving mode (HiZ). 1 = IO is in driving mode.

Address:	85	Register:	Slave Side Port GPIO Mode SDA Output Enable (M0/M1 shared - Per Device)
Bits	Attribute	Default	Description
7:0	RW	0x00	GPIO_SSDA_OUT_EN Slave Side Port GPIO Mode IO output Enable. Bit[n] maps to Slave Side Port n Control, i.e. Bit[0] Controls Port 0 SDA pin (GPIO[n]1). For any Bit[n]: 0 = IO is not in driving mode (HiZ). 1 = IO is in driving mode.

Address:	86	Register:	GPIO SSCL Output Level (M0/M1 shared - Per Device)
Bits	Attribute	Default	Description
7:0	RW	0xFF	GPIO_SSCL_OUT_LEVEL For any Bit[n]: Maps to GPIO[n]0 pin. If port n is in GPIO mode, this register is the SDA[n] GPIO output value control, i.e. controls the logic value that is output to the IO PAD if the output enables of IO is set. 0: Logic Low 1: Logic High

Address:	87	Register:	GPIO SDA Output Level (M0/M1 shared - Per Device)
Bits	Attribute	Default	Description
7:0	RW	0xFF	GPIO_SSDA_OUT_LEVEL For any Bit[n]: Maps to GPIO[n]1 pin If port n is in GPIO mode, this register is the SDA[n] GPIO output value control, i.e. controls the logic value that is output to the IO PAD if the output enables of IO is set. 0: Logic Low 1: Logic High

Address:	88	Register:	Slave Side Ports Input Detection Mode Configuration (M0/M1 shared - Per Device)
Bits	Attribute	Default	Description
7:6	RW	0x0	SSDA2367_IO_IN_DET_CFG Slave Side Port SSDA[2/3/6/7] IOs Input Change Detection Configuration. 00 = Detection disabled 01 = Detection Flag is set when IO Input transitions from "0" to "1" 10 = Detection Flag is set when IO Input transitions from "1" to "0" 11 = Detection Flag is set when IO Input transitions from "0" to "1" or "1" to "0"
5:4	RW	0x0	SSCL2367_IO_IN_DET_CFG Slave Side Port SSCL[2/3/6/7] IOs Input Change Detection Configuration. 00 = Detection disabled 01 = Detection Flag is set when IO Input transitions from "0" to "1" 10 = Detection Flag is set when IO Input transitions from "1" to "0" 11 = Detection Flag is set when IO Input transitions from "0" to "1" or "1" to "0"
3:2	RW	0x0	SSDA0145_IO_IN_DET_CFG Slave Side Port SSDA[0/1/4/5] IOs Input Change Detection Configuration. 00 = Detection disabled 01 = Detection Flag is set when IO Input transitions from "0" to "1" 10 = Detection Flag is set when IO Input transitions from "1" to "0" 11 = Detection Flag is set when IO Input transitions from "0" to "1" or from "1" to "0"
1:0	RW	0x0	SSCL0145_IO_IN_DET_CFG Slave Side Port SSCL[0/1/4/5] IOs Input Change Detection Configuration. 00 = Detection disabled 01 = Detection Flag is set when IO Input transitions from "0" to "1" 10 = Detection Flag is set when IO Input transitions from "1" to "0" 11 = Detection Flag is set when IO Input transitions from "0" to "1" or from "1" to "0"

Address:	89	Register:	SSCL Input Detection IBI Enable (M0/M1 dedicated - Per Port)
Bits	Attribute	Default	Description
7:0	RW	0x0	SSCL_IO_IN_DET_IBI_EN For any Bit[n]: Maps to SSCL[n]/GPIO[n]0 pin The bit controls if the corresponding SSCL_IO_IN_DET_Flag generates IBI.

Address:	90	Register:	SSDA Input Detection IBI Enable (M0/M1 dedicated - Per Port)
Bits	Attribute	Default	Description
7:0	RW	0x00	SSDA_IO_IN_DET_IBI_EN For any Bit[n]: Maps to SSDA[n]/GPIO[n]1 pin The bit controls if the corresponding SSDA_IO_IN_DET_Flag generates IBI.

### 10.3.8 Slave Side Port Status

Address:	96	Register:	Slave Side Port Real Time SSCL IO Status (M0/M1 Shared- Per Device)
Bits	Attribute	Default	Description
7:0	R	0x00	SSCL_IO_IN Reflects the real-time status of Slave Interfacing Port SSCL IO Input. Bit[n] maps to Slave Side Port n Control, i.e. Bit[0] Controls the Port 0 SSCL pin. For any Bit[n]: 0 = IO Input is low 1 = IO Input is high.

Address:	97	Register:	Slave Side Port Real Time SSDA IO Status (M0/M1 Shared- Per Device)
Bits	Attribute	Default	Description
7:0	R	0x00	SSDA_IO_IN Reflects the real-time status of Slave Interfacing Port SSDA IO Input. Bit[n] maps to Slave Side Port n Control, i.e. Bit[0] Controls the Port 0 SSDA pin. For any Bit[n]: 0 = IO Input is low 1 = IO Input is high.

Address:	98	Register:	Slave Side Port SSCL IO Detection Flags (M0/M1 Shared- Per Device)
Bits	Attribute	Default	Description
7:0	RW1C	0x00	SSCL_IO_IN_DET_Flag SSCL IO Input detection captured flag. If an IO transition (as defined in REG#88) is detected at corresponding IO input, this bit is set. The bit is cleared after a "1" written to the bit field. Bit[n] maps to Slave Side Port n Control, i.e. Bit[0] Controls the Port 0 SSCL pin.

Address:	99	Register:	Slave Side Port SSDA IO Detection Flags (M0/M1 Shared- Per Device)
Bits	Attribute	Default	Description
7:0	RW1C	0x00	SSDA_IO_IN_DET_Flag SSDA IO Input detection captured flag. If an IO transition (as defined in REG#88) is detected at the corresponding IO input, this bit is set. The bit is cleared after a "1" is written to the bit field. Bit[n] maps to Slave Side Port n Control, i.e. Bit[0] Controls the Port 0 SSDA pin.

### 10.3.9 SMBus Agent Status and Configuration Registers

Address:	100 101 102 103 104 105 106 107	Register:	Slave Side Port 0 Status (M0/M1 Shared- Per Devices) Slave Side Port 1 Status (M0/M1 Shared- Per Devices) Slave Side Port 2 Status (M0/M1 Shared- Per Devices) Slave Side Port 3 Status (M0/M1 Shared- Per Devices) Slave Side Port 4 Status (M0/M1 Shared- Per Devices) Slave Side Port 5 Status (M0/M1 Shared- Per Devices) Slave Side Port 6 Status (M0/M1 Shared- Per Devices) Slave Side Port 7 Status (M0/M1 Shared- Per Devices)
Bits	Attribute	Default	Description
7:4	R	0x0	Master_Agent_Transaction_Return_Code This is the code that the Master Agent Returns after a transaction is executed. See Return Code Mapping table for details. This field is updated when the bit[0] of this register (Master_Agent_Trans_Finish_Flag) is set.
3	RW1C	0x0	Slave_Agent_Buf_Overflow_Flag Flag of Port Slave Agent new transaction detected when both buffers are full. NAK will be responded.
2	RW1C	0x0	Slave_Agent_Buf1_Received_Flag Flag of Port Slave Agent Transaction Received at buffer 1. Master should clear this register after retrieving data from the buffer to make the buffer available to the next message.
1	RW1C	0x0	Slave_Agent_Buf0_Received_Flag Flag of Port Slave Agent Transaction Received at buffer 0. Master should clear this register after retrieving data from the buffer to make the buffer available to the next message.
0	RW1C	0x0	Master_Agent_Trans_Finish_Flag Flag of Port Master Agent Transaction finished.

Address:	108	Register:	Slave Side Port Bus Agent Configuration (M0/M1 Shared- Per Device)
Bits	Attribute	Default	Description
7:6	B	0x0	Reserved
5	RW	0x0	Slave_Agent_Buf_Full_SDA_Low_Enable This bit enables a feature of holding SDA low when input packet buffers are full, as a flow-control measure for MCTP application to avoid the slave side MCTP transmitters time out when the bus agent buffers are not serviced in time. This bit cannot be set when bit[1] (I3C_HJ_Message_Receive_En) is set. 1: For any bus agent, if both of slave agent buffers are full (see REG#100-107, bit[3:2]), the SSSDA pin of that port will be hold low. The SSSDA holding-low will be released if either this bit is cleared, or any bit of bit[3:2] of that SMBus Agent (See REG#100-107) get cleared. 0: Function disabled
4	RW	0x0	I3C_HJ_ACK_STOP 0: When the HJ message is ACKed, No Stop issued - SDA go high before SCL go high 1: When the HJ message is ACKed, Stop will be issued.
3	RW	0x0	SMBus_Timeout_Disable 0: SMBus SCL timeout will be detected and responded 1: SMBus SCL timeout will be ignored- behave like I2C
2	RW	0x0	I3C_HJ_Message_NAK This bit controls how the SMBus Agent acknowledge the "02h" Hot Join address, if enabled in bit field 1 of this register below. 0: The HJ message will be ACKed. According to the configuration of bit[4], STOP is generated following the Ack. 1: The HJ message will be NAKed. Stop will be issued.

Address:	108	Register:	Slave Side Port Bus Agent Configuration (M0/M1 Shared- Per Device)
Bits	Attribute	Default	Description
1	RW	0x0	<b>I3C_HJ_Message_Receive_En</b> This bit enables the SMBus Slave Agent's reception of I3C Hot Join message. In this mode, after a Start condition, SMBus Agent generates a 100Khz Clock in OD mode and arbitrates with input SSCL in the address byte. If the Address is 02h (I3C Hot Join address), the Slave Agent will send ACK or NAK to terminate the message and put the first byte into the Slave Agent buffer, and update the status registers. If IBI is enabled, IBI will be sent to notify the master. 0: Disabled 1: Enabled
0	RW	0x0	<b>BUS_Agent_TX_RX_Loopback_En</b> This bit configures if the Slave Agent will receive the write message transmitted from the Master Agent. 0: Slave Agent only receive write message from the downstream port. 1: Slave Agent receive any write message on the bus, including messages sent from the Master Agent, provided the message address are not in the local address list (filtering list).

### 10.3.10 Special Registers

Address:	121	Register:	LDO Power Good and MSEL input Status
Bits	Attribute	Default	Description
7	R	0x0	<b>MSDA1_Level</b> MSDA1 Pin logic input real time status 0: Pin input logic low 1: Pin input logic high
6	R	0x0	<b>MSCL1_Level</b> MSCL1 Pin logic input real time status 0: Pin input logic low 1: Pin input logic high
5:4	R	0x0	<b>MSEL_Pin_Input_Code</b> 00: MSEL pin is driven logic low 11: MSEL pin is driven logic high 01: MSEL pin is floating / HiZ 10: Reserved Code - should never happen
3	R	0x0	<b>VIOS1_Pwr_Good</b> Power Good Status of VIOS1 0: Power is not good 1: Power is good
2	R	0x0	<b>VIOS0_Pwr_Good</b> Power Good Status of VIOS0 0: Power is not good 1: Power is good
1	R	0x0	<b>VIOM1_Pwr_Good</b> Power Good Status of VIOM1 0: Power is not good 1: Power is good
0	R	0x0	<b>VIOM0_Pwr_Good</b> Power Good Status of VIOM0 0: Power is not good 1: Power is good

Address:	122	Register:	Bus Reset Timeout Setting (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:3	RW	0x0	Reserved
2:0	RW	0x5	BusReset_Timeout Master Side Port 0 On-Chip Slave Bus Reset SCL Timeout setting 0: 1ms 1: 2ms 2: 4ms 3: 6ms 4: 9ms 5: 12ms 6: 18ms 7: 25ms

Address:	123	Register:	I3C On-Chip Slave Protocol Error Flags (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7	B	0x0	Reserved
6	B	0x0	Reserved
5	RW1C	0x0	S5_Flag
4	RW1C	0x0	S4_Flag
3	RW1C	0x0	S3_Flag
2	RW1C	0x0	S2_Flag
1	RW1C	0x0	S1_Flag
0	RW1C	0x0	S0_Flag

Address:	124	Register:	Device Command (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:0	WP	0x00	DEV_CMD Device or Interface Specific Command. The device level commands (e.g. Device Reset) are subject to the control of REG#19[1] bit control. The Security Lock Commands are Interface Specific - It applies the lock to the interface from which the commands were received. Device Command encoding: 0x96 = Device Reset. (applies to the whole device) 0x97 = I3C Interface Reset. (applies to the interface attached) 0xA1 = Driver Strength Calibration (applies to the whole device, after VIO applied). The calibration takes tIOCal to finish. Security Lock Command encoding: 0x51 = Lock register write access to other port's register via the paged window (Supported by M0 only — password to write M1 registers will be ignored. Lock remains until a pin reset and power cycle) 0x52 = Lock all critical register access from I2C or I3C private write access from the master side port. Critical Register List: <ul style="list-style-type: none"> <li>▪ REG#17</li> <li>▪ REG#19</li> <li>▪ REG#21</li> <li>▪ REG#22</li> <li>▪ REG#23</li> <li>▪ REG#25</li> </ul> 0x53 = Lock all protected register access from I2C or I3C private write access from this port (for M0, including access to M1) until a pin reset or power cycle 0x54 = Lock/Disable the 0x96 command until a pin reset or power cycle 0x55 = Lock/Disable the 0x97 command until a pin reset or power cycle 0x59 = Lock all vendor register access from I2C or I3C private write access from this port until a pin reset or power cycle

			Note: 0x59 command only locks vendor registers. Issuing both 0x59 command and 0x53 command in sequence if both vendor register and user protected configuration register need to be locked. Others = Reserved or Vendor usage - Do not issue
--	--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Address:	125	Register:	On-Chip Slave Reference Status (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:3	R	0x0	Reserved
2	R	0x1	HJ_Enabled On Chip Slave Hot Join Enabled Status. Enabled by ENEC. 0 = Disabled 1 = Enabled This bit is cleared with DISEC CCC and set by ENEC CCC
1	R	0x0	IBI_Enabled On Chip Slave IBI Enabled Status. Set / Reset by ENEC / DISEC CCC. 0 = Disabled 1 = Enabled In the JEDEC Bus Context this bit is Reset by BUS Reset event.
0	R	0x0	DA_Assigned On-Chip Slave Dynamic Address Assigned Status. Set by ENTDA/SETAASA/SETDASA, Reset by RSTDA. In the JEDEC Bus Context this bit Reset by BUS Reset event.

Address:	126	Register:	Master Interfacing Port On-Chip Slave Serial Address (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:1	RWP	0x00	SLAVE_ADDR Slave Address Re-configured. This field is programmed by OTP or by user to change the default Slave Static Address. If the register is programmed by the I2C interface, the new address takes effect after a STOP is received and the bus is idle.
0	RWP	0x0	SLAVE_ADDR_CFG_EN Slave Address Reconfigure Enable. 0 = Don't use the slave address defined in this register. 1 = Use the slave address defined in this register as the new Slave Address.

Address:	127	Register:	Page Pointer (M0/M1 Dedicated - Per Port)
Bits	Attribute	Default	Description
7:0	RW	0x0	PAGE_PTR Page Pointer. Specifies the Current Page Number of the Paged address spaces.

## 10.4 Part Number Specific Configurations

The different part numbers (see 2.1 Part Numbers) in the I3C Hub family powers up with a different configuration. The configuration setting below overwrites the defaults that listed in section 5.3 Register Definition Details.

**Table 26. Part Number Specific Configuration**

Register (decimal)	Description	PJ71M XX (XX =87/47/88/48/89/49)						Comment
		87	47	88	48	89	49	
0	Device Info	0x87	0x47	0x88	0x48	0x89	0x49	
1	Device Info	0x12	0x12	0x12	0x12	0x12	0x12	
2-7	PID Random number	random	random	random	random	random	random	
8	BCR	0x16	0x16	0x16	0x16	0x16	0x16	
10	Device Capability	0x0f	0x07	0x0f	0x07	0x0f	0x07	
17	Master Side Port Configuration	0x10	0x10	0x00	0x00	0x10	0x10	
18	Slave Side Port Enable	0x00	0x00	0xff	0x0f	0x00	0x00	
19	Device Configuration	0x0c	0x0c	0x0c	0x0c	0x0c	0x0c	
20	IO Drive Strength	0x05	0x05	0x00	0x00	0x05	0x05	
21	Hub Network Operation Mode	0x00	0x00	0x20	0x20	0x00	0x00	Analog Switch function default not enabled in A0 device
22	Interface Voltage Setting	0x00	0x00	0xff	0xff	0x05	0x5f	
23	Slave Side Port OD Only Setting	0xff	0xff	0x00	0x00	0xff	0xff	
24	Slave Side Port SMBus Agent Enable	0x00	0x00	0x00	0x00	0x00	0x00	
25	LDO Enable and Slave Side Pull Up Resistor Value	0x50	0x50	0xa0	0xa0	0x5f	0x5f	Pull up default to 1Kohm for PJ71M88BQW
30	Slave Side Port GPIO Mode Enable	0x00	0x00	0x00	0x00	0x00	0x00	
81	Slave Side Port Hub Connection Control	0x00	0x00	0xff	0x0f	0x00	0x00	

## 11 Supported CCC and Frame Mapping

### 11.1 I3C Basic 1.1 CCC

#### 11.1.1 SETBUSCON CCC

SETBUSCON CCC is an I3C Basic 1.1 CCC.

The PJ71MXX supports SETBUSCON CCC, offer a simple setup command to configure the device in a known bus context, like the JEDEC Module Bus Operation.

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 <sup>[1]</sup>	
	0x0C(Broadcast)								T	
	Context								T	
	Optional Bytes ... (Ignored)								T	Sr or P

- (1). The PJ71MXX NACKs if there is a parity or a PEC error in a previous transaction when the host performs consecutive transactions with Repeat Start.

If the Context byte matches the REG#29, a number of PJ71MXX configuration registers will be updated according to the JEDEC SPD Bus usage model.

### 11.2 JEDEC SPD Bus CCCs

#### 11.2.1 GETSTATUS CCC (JEDEC and I3C Basic 1.0)

##### 11.2.1.1 GETSTATUS CCC – Direct, Without PEC (MIPI Compatible)

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 <sup>[1]</sup>	
	0x90 (Direct)								T	
Sr	DevID[6:0]							R=1	A <sup>[1]</sup>	
	REG#32 PEC_Err	0	0	0	REG#125 HJ_ Enabled	0	0	REG#125 5 [5/1] IBI Enabled	T	
	0	0	Protocol Error <sup>[2]</sup>	0	Pending Interrupt number (IBI Source)				T	Sr or P

- (1). The PJ71MXX NACKs if there is a parity or PEC error in a previous transaction when the host performs consecutive transactions with Repeat Start.
- (2). Per I3C Spec, the Protocol Error is a read-and-clear log of S0-S5 errors detected between two GETSTATUS CCC cycles. Another REG#123 also logs the errors and it is cleared by write 1 to the bits.

### 11.2.1.2 GETSTATUS CCC – Direct, With PEC (JEDEC SPD Bus)

If the 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 the PEC calculation.

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 <sup>[1]</sup>	
	0x90 (Direct)								T	
	PEC								T	
Sr	DevID[6:0]							R=1	A <sup>[1]</sup>	
	REG#32 PEC_Err	REG#32 REG125[4/0] I3C_ mode	0	0	REG125[6/2] REG#12 5HJ_ Enabled	0	0	REG125[5/1] REG#12 5IBI Enabled	T	
	0	0	Protocol Error	0	Pending Interrupt number (IBI Source)				T	
	PEC								T	Sr or P

(1). The PJ71MXX NACKs if there is a parity or PEC error in a previous transaction when the host performs consecutive transactions with Repeat Start.

### 11.2.2 SETAASA CCC (JEDEC and I3C Basic 1.0)

SETAASA CCC does not support the PEC function, and the frame format is the same regardless PEC setting.

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	
	0x29 (Broadcast)								T	P

### 11.2.3 DEVCTRL CCC (JEDEC SPD Bus)

Per JEDEC SPD bus definition, this CCC may address a device ID in I2C Mode. I3C Hub device requires the Host to terminate DEVCTRL CCC with a STOP, not a Repeat Start.

#### 11.2.3.1 Without 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 <sup>[1]</sup>	
	0x62 (Broadcast)								T	
	AddrMask[2:0]		StartOffset[1:0]		PEC BL[1:0]		RegMod		T	
	DevID[6:0]							0	T	
	Byte 0 Data Payload								T	
	Byte 1 Data Payload								T	
	Byte 2 Data Payload								T	
	Byte 3 Data Payload								T	P

(1). The PJ71MXX NACKs if there is a parity or PEC error in a previous transaction when the host performs consecutive transactions with Repeat Start.

### 11.2.3.2 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 <sup>[1]</sup>	
	0x62 (Broadcast)								T	
	AddrMask[2:0]		StartOffset[1:0]		PEC BL[1:0]		RegMod		T	
	DevID[6:0]							0	T	
	Byte 0 Data Payload								T	
	Byte 1 Data Payload								T	
	Byte 2 Data Payload								T	
	Byte 3 Data Payload								T	
	PEC								T	P

- (1). The PJ71MXX NACKs if there is a parity or PEC error in a previous transaction when the host performs consecutive transactions with Repeat Start.

### 11.2.3.3 DEVCTRL CCC Command Definition

Parameter	Definition
AddrMask[2:0]	Broadcast, Unicast or Multicast Command Selection 000 = Unicast Command; PJ71MXX responds if the DevID[6:0] field matches with PJ71MXX own 7-bit address 011 = Multicast Command; PJ71MXX device and other devices respond if DevID[6:3] field matches with PJ71MXX device's own 4-bit MSB Slave address 111 = Broadcast Command; All devices responds 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. The Host can start at any Byte (from Byte 0 to Byte 3) and has continuous access to next byte until the STOP operation. If Byte 3 is reached, the host is responsible for applying the 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

Parameter	Definition
RegMod	Identifies if DEVCTRL is used for General Registers as identified in Byte 0 to Byte 3 or used for 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] & 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 the device to respond. DevID[2:0] is don't care. For any other codes for AddrMask[2:0], the device ignores the write transaction.

### 11.2.3.4 DEVCTRL CCC Data Payload Definition

Byte #	Bit #	Function	Definition	Comment
Byte 0	[7]	PEC Enable	0 = Disable 1 = Enable	REG#17 [0] is updated
	[6]	Parity Disable	0 = Enable 1 = Disable	REG#17[1] is updated
	[5:2]	RFU	RFU	
	[1]	RSVD	0 = RSVD 1 = RSVD	
	[0]	RFU	RFU	
Byte 1	[7:4]	RFU	RFU	
	[3]	Global & IBI Clear	0 = No Action 1 = Clear All Event and pending IBI	REG#32[3:0], REG#100-107[3:0], REG#98,REG#99 are cleared.
	[2:0]	RFU	RFU	
Byte 2	[7:0]	RFU	RFU	
Byte 3	[7:0]	RFU	RFU	

### 11.2.4 ENEC CCC (JEDEC and I3C Basic 1.0)

#### 11.2.4.1 ENEC Broadcast CCC Without PEC (Compatible with MIPI)

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 <sup>[1]</sup>	
	0x00 (Broadcast)								T	
	0x0				ENHJ	0x0	0x0	ENINT	T	Sr <sup>[2]</sup> or P

- (1). The PJ71MXX 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.

#### 11.2.4.2 ENEC Broadcast CCC With PEC (JEDEC SPD Bus)

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 <sup>[1]</sup>	
	0x00 (Broadcast)								T	
	0x0				ENHJ	0x0	0x0	ENINT	T	
	PEC								T	Sr <sup>[2]</sup> or P

- (1). The PJ71MXX NACKs if there is a parity or PEC error in a previous transaction when the host performs consecutive transactions with Repeat Start.
- (2). Repeat Start or Repeat Start with 7'h7E.

### 11.2.4.3 ENEC Direct CCC Without PEC (Compatible with MIPI)

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 <sup>[1]</sup>	
	0x80 (Direct)								T	
Sr	DevID[6:0]							W=0	A <sup>[1][2]</sup>	
	0x0			ENHJ	0x0	0x0	ENINT	T	Sr <sup>[3]</sup> or P	

- (1). The PJ71MXX NACKs if there is a parity or PEC error in a previous transaction when the host performs consecutive transactions with Repeat Start.
- (2). The PJ71MXX does not check for parity error in subsequent bytes when it determines the 7-bit device select code issued by the host does not match with its device code. The SPD5 Hub device ignores the entire packet until STOP or the next Repeat Start operation.
- (3). Repeat Start or Repeat Start with 7'h7E.

### 11.2.4.4 ENEC Direct CCC With PEC (JEDEC SPD Bus)

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 <sup>[1]</sup>	
	0x80 (Direct)								T	
	PEC								T	
Sr	DevID[6:0]							W=0	A <sup>[1][2]</sup>	
	0x0			ENHJ	0x0	0x0	ENINT	T		
	PEC								T	Sr <sup>[3]</sup> or P

- (1). The PJ71MXX NACKs if there is a parity or PEC error in a previous transaction when the host performs consecutive transactions with Repeat Start.
- (2). The PJ71MXX device does not check for parity or PEC error in subsequent bytes when it determines the 7-bit device select code issued by the host does not match with its device code. The SPD5 Hub device ignores the entire packet until STOP or the next Repeat Start operation.
- (3). Repeat Start or Repeat Start with 7'h7E.

### 11.2.4.5 ENEC CCC Data Encoding

Bit	Encoding
ENINT	0 = No Action 1 = Enable IBI Interrupt
ENHJ	0 = No Action 1 = Enable Hot Join

### 11.2.5 DISEC CCC (JEDEC and I3C Basic 1.0)

When DISEC CCC is registered by the I3C Hub, it updates IBI\_EN and HJ\_EN configuration in the chip, and it takes effect at the next Start operation (i.e after STOP operation).

For JEDEC SPD Bus Operation, if the 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.

**11.2.5.1 DISEC CCC – Broadcast, without PEC (MIPI Compatible)**

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 <sup>[1]</sup>	
	0x01 (Broadcast)								T	
	0x0				DIS HJ	0x0	0x0	DISINT	T	

- (1). The I3C Hub NACKs if there is a parity error in a previous transaction when the host performs consecutive transactions with Repeat Start.
- (2). Repeat Start or Repeat Start with 7'h7E.

**11.2.5.2 DISEC CCC – Broadcast, with PEC (JEDEC SPD Bus)**

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 <sup>[1]</sup>	
	0x01 (Broadcast)								T	
	0x00				DISHJ	0x0	0x0	DISINT	T	
	PEC								T	

- (1). The I3C Hub NACKs if there is a parity error in a previous transaction when the host performs consecutive transactions with Repeat Start.
- (2). Repeat Start or Repeat Start with 7'h7E.

**11.2.5.3 DISEC CCC – Direct, without PEC (MIPI Compatible)**

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 <sup>[1]</sup>		
	0x81 (Direct)								T		
Sr	DevID[6:0]								W=0		A <sup>[1][2]</sup>
	0x00				DIS HJ	0x0	0x0	DISINT	T		Sr <sup>[3]</sup> or P

- (1). The I3C Hub NACKs if there is a parity error in a previous transaction when the host performs consecutive transactions with Repeat Start.
- (2). The PJ71MXX does not check for parity error in subsequent bytes when it determines the 7-bit device select code issued by the host does not match with its device code. The SPD5 Hub device ignores the entire packet until STOP or the next Repeat Start operation.
- (3). Repeat Start or Repeat Start with 7'h7E.

**11.2.5.4 DISEC CCC – Direct, with PEC (JEDEC SPD Bus)**

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 <sup>[1]</sup>		
	0x81 (Direct)								T		
	PEC								T		
Sr	DevID[6:0]								W=0		A <sup>[1][2]</sup>
	0x00				DIS HJ	0x0	0x0	DISINT	T		
	PEC								T		Sr <sup>[3]</sup> or P

- (1). The PJ71MXX NACKs if there is a parity or PEC error in a previous transaction when the host performs consecutive transactions with Repeat Start.
- (2). The PJ71MXX device does not check for parity or PEC error in subsequent bytes when it determines the 7-bit device select code issued by the host does not match with its device code. The SPD5 Hub device ignores the entire packet until STOP or the next Repeat Start operation.
- (3). Repeat Start or Repeat Start with 7'h7E.

### 11.2.5.5 DISEC CCC Data Encoding

Bit	Encoding
DISINT	0 = No Action 1 = Disable IBI Interrupt
DISHJ	0 = No Action 1 = Disable Hot Join

### 11.2.6 RSTDAA CCC (JEDEC and I3C Basic 1.0)

When RSTDAA CCC is registered by the I3C Hub, it updates dynamic address assignment status and INF\_SEL bit in GETSTATUS CCC to '0'.

It takes effect at the next Start operation (i.e after STOP operation).

#### 11.2.6.1 RSTDAA CCC – Broadcast, without PEC (MIPI Compatible)

In JEDEC SPD Bus Operation, if the 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.

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 <sup>[1]</sup>	
	0x06 (Broadcast)								T	P

- (1). The I3C Hub NACKs if there is a parity or PEC error in a previous transaction when the host performs consecutive transactions with Repeat Start.

#### 11.2.6.2 RSTDAA CCC – Broadcast, with PEC (JEDEC SPD Bus)

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 <sup>[1]</sup>	
	0x06 (Broadcast)								T	
	PEC								T	P

- (1). The I3C Hub NACKs if there is a parity or PEC error in a previous transaction when the host performs consecutive transactions with Repeat Start.

#### 11.2.6.3 RSTDAA CCC – Direct, without PEC (MIPI Compatible)

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 <sup>[1]</sup>	
	0x86 (Direct)								T	
Sr	DevID[6:0]							W=0	A <sup>[1][2]</sup>	P

- (1). The I3C Hub NACKs if there is a parity error in a previous transaction when the host performs consecutive transactions with Repeat Start.
- (2). The I3C Hub device does not check for parity error in subsequent bytes when it determines the 7-bit device select code issued by the host does not match with its own device code. The I3C Hub device ignores the entire packet until STOP or the next Repeat Start operation.

#### 11.2.6.4 RSTDAA CCC - Direct with PEC (JEDEC SPD Bus)

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 <sup>[1]</sup>	
	0x86 (Direct)								T	
	PEC								T	

Sr	DevID[6:0]	W=0	A <sup>[1][2]</sup>	
	PEC		T	P

- (1). The I3C Hub NACKs if there is a parity or PEC error in a previous transaction when the host performs consecutive transactions with Repeat Start.
- (2). The I3C Hub device does not check for parity error in subsequent bytes when it determines the 7-bit device select code issued by the host does not match with its own device code. The I3C Hub device ignores the entire packet until STOP or the next Repeat Start operation.

### 11.2.7 DEVCAP CCC (JEDEC SPD Bus)

In I3C JEDEC SPD Bus Operation, 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.

#### 11.2.7.1 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 <sup>[1]</sup>	
	0xE0 (Direct)								T	
Sr	DevID[6:0]						R=1	A <sup>[1]</sup>		
	MSB (Each bit defines capability)								T	
	LSB (Each bit defines capability)								T	Sr <sup>[2]</sup> or P

- (1). The I3C Hub NACKs if there is a parity error in a previous transaction when the host performs consecutive transactions with Repeat Start.
- (2). Repeat Start or Repeat Start with 7'h7E.

#### 11.2.7.2 DEVCAP CCC - Direct w/ 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 <sup>[1]</sup>	
	0xE0 (Direct)								T	
	PEC								T	
Sr	DevID[6:0]						R=1	A <sup>[1]</sup>		
	MSB (Each bit defines capability)								T	
	LSB (Each bit defines capability)								T	
	PEC								T	Sr <sup>[2]</sup> or P

- (1). The I3C Hub NACKs if there is a parity or PEC error in a previous transaction when the host performs consecutive transactions with Repeat Start.
- (2). Repeat Start or Repeat Start with 7'h7E.

#### 11.2.7.3 DEVCAP CCC Byte Encoding

Bit	Encoding	Notes
MSB [7]	GETBCR CCC Support	Coded as '1'
MSB[6]	GETDCR CCC Support	Coded as '1'
MSB[5]	GETMRL CCC Support	Coded as '0'
MSB[4]	GETMWL CCC Support	Coded as '0'
MSB[3]	GETPID CCC Support	Coded as '1'
MSB[2]	0 = No Support for Timer based Reset 1 = Supports Timer based Reset	I3C Hub hard codes to '1'
MSB[1:0]	RFU	Coded as '000'
LSB[7:0]	RFU	Coded as '000'

### 11.2.8 SETHID CCC (JEDEC SPD Bus)

JEDEC definition states that SETHID CCC will be sent prior to Dynamic Address Assignment with SETAASA in JEDEC SPD Bus. The PJ71MXX I3C Hub device's on-chip slave will ignore this CCC. The PJ71MXX I3C Hub device's Hub Network propagates this CCC transparently as other CCCs that it supports.

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

### 11.3 I3C Basic 1.0 standard CCCs

#### 11.3.1 Supported I3C Basic 1.0 standard CCCs

I3C Hub Device supports the following standard I3C Basic 1.0 CCCs:

- SETNEWDA
- SETDASA
- GETPID (PID returns value as in REG#2-7 registers)
- GETBCR (BCR returns value as in REG#8 registers)
- GETDCR (DCR returns the value as in REG#9 registers)
- ENTDA

Refer to I3C Basic 1.0 Specification for the CCC framing details.

#### 11.3.2 PJ71MXX BCR

BIT	Name	MIPI I3C Basic 1.0 Description	PJ71MXX Device Setting
BCR [7]	Device Role <sup>[1]</sup>	2'b00 – I3C Slave 2'b01 – I3C Master <sup>[1]</sup> 2'b10 – Reserved for future definition by MIPI 2'b11 – Reserved for future definition by MIPI	0
BCR [6]	Device Role <sup>[0]</sup>		0
BCR [5]	MIPI Reserved	0 – Default	0
BCR [4]	Bridge Identifier <sup>[2]</sup>	0 – Not a Bridge Device 1 – Is a Bridge Device	1 <sup>[3]</sup>
BCR [3]	Offline Capable <sup>[4]</sup>	0 – Device will always respond to I3C Bus commands 1 – Device will not always respond to I3C Bus commands	0
BCR [2]	IBI Payload	0 – No data byte follows the accepted IBI 1 – Mandatory one or more data bytes follow the accepted IBI. Data byte continuation is indicated by T-bit	1
BCR [1]	IBI Request Capable	0 – Not Capable 1 – Capable	1
BCR [0]	Max Data Speed Limitation <sup>[5]</sup>	0 – No Limitation 1 – Limitation	0

(1). For an I3C Device acting as I3C Main Master, the BCR Device Role bits will contain the value 2'b01.

(2). Bridge Devices are required to comply with the MIPI Specification for I3C v 1.0 [MIPI02].

(3). The PJ71MXX is a Transparent Hub device, but also report a bridge device since it provide bridging to SMBUS with a Bus Agent.

(4). Offline Capable Devices retain the Dynamic Address.

(5). Master shall use the GETMXDS CCC to interrogate the Slave for specific limitation.

**11.3.3 PJ71MXX DCR**

Bit	Name	MIPI I3C Basic 1.0 Description	PJ71MXX Setting (as a "Hub" Device)
DCR [7]	Device ID [7]	255 available codes for describing the type of sensor, or Device. Examples: Accelerometer, gyroscope, composite Devices. The PJ71MXX will have the "Hub" Function.	1
DCR [6]	Device ID [6]		1
DCR [5]	Device ID [5]		0
DCR [4]	Device ID [4]		0
DCR [3]	Device ID [3]		0
DCR [2]	Device ID [2]		0
DCR [1]	Device ID [1]		1
DCR [0]	Device ID [0]		0

## 12 Theory of Operation

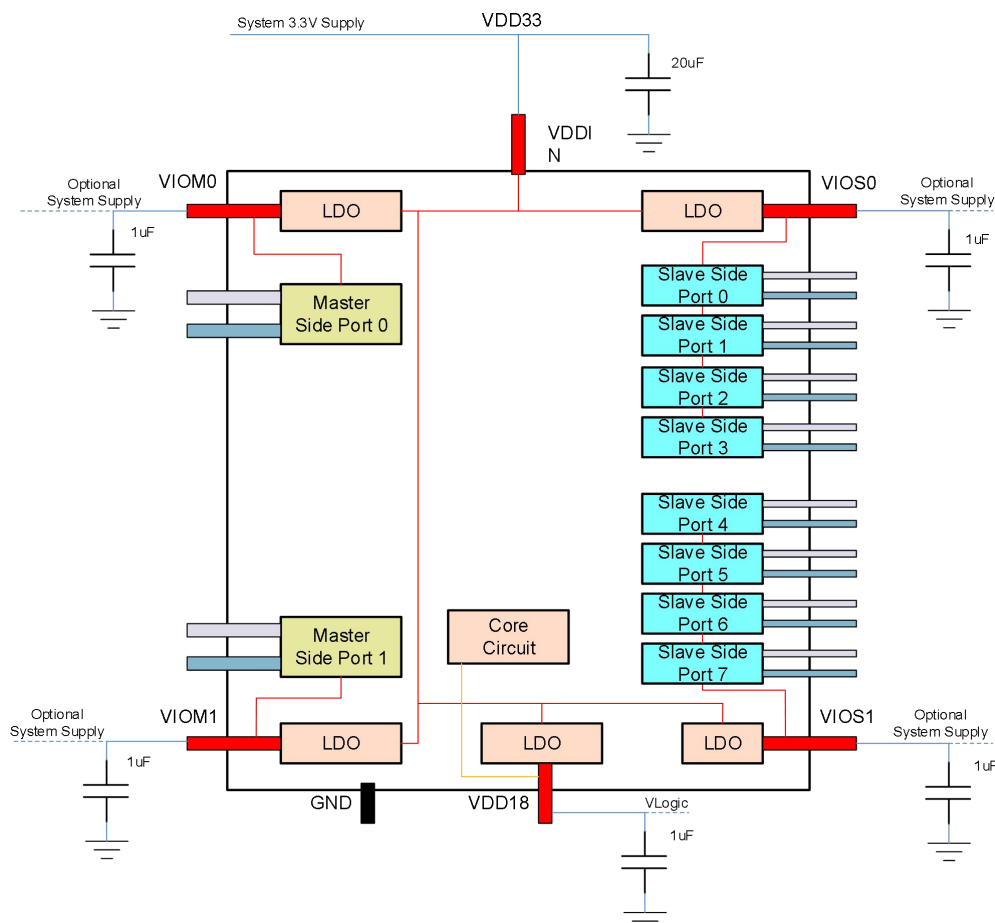
### 12.1 Power Supply

The PJ71MXX is supplied by a single power supply  $V_{DDIN}$ .  $V_{DDIN}$  needs to be decoupled to the ground with a 20uF MLCC capacitor.  $V_{DDIN}$  can range from 1.7V to 3.3V, and the nominal voltage is 3.3V.

The PJ71MXX on-chip Low-Dropout Regulators (LDO) generates a fixed 1.8V main Power ( $V_{DD18}$ ) and optionally generates IO power ( $V_{IOM}$ ,  $V_{IOS}$ ) at the different voltage level of 1.0V, 1.1V, 1.2V, or 1.8V.  $V_{DD18}$  and VIO need to be decoupled to the ground with a 1uF MLCC capacitor. When the  $V_{DDIN}$  is 1.8V, there will be no headroom for the  $V_{DD18}$  Regulator to regulate the  $V_{DD18}$  Rail. In this case, it is suggested user short  $V_{DDIN}$  and  $V_{DD18}$  directly.

- $V_{DD18}$  regulator is fixed to 1.8V and enabled.
- IO regulators are programmable (see REG#22) with the options of 1.0V, 1.1V, 1.2V, or 1.8V, by default they are disabled. They can be enabled by programming the LDO Enable configuration register (see REG#25) setting via an I2C or I3C command.
- Each VIO pin is associated a power good detector. Those power detectors detects if the VIO voltage level falls into +/- 12% of the nominal voltage setting defined in REG#22. The power good status is accessible at REG#121.
- The IO supplies ( $V_{IOM0/1}$ ,  $V_{IOS0/1}$ ) should be set to the same voltage of IO operation. In Open-Drain Only operation, if the IO voltage is greater than 1.8V, the IO supply should be set to 1.8V.  $V_{IOM0/1}$  are not required to allow an OD\_Only port to access the on-chip slave in I2C Mode.

The PJ71MXX device does not have a constraint on the power-up sequence. All power-ups can be started in any order.



**Figure 24. Power Supply Diagram**

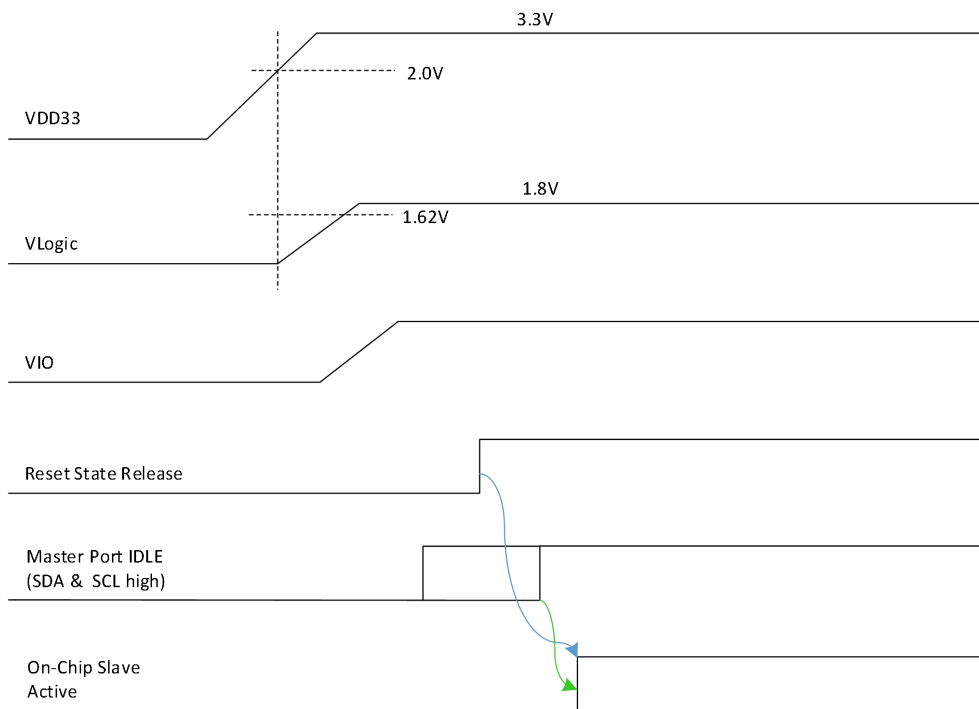
## 12.2 Initialization

After the VDDIN power ramping above 2V, the VDD18 regulator is enabled.

When the VDD18 Regulator voltages reach 90% of the target value, the internal logic goes out of the reset state and start to initialize the device.

The Key Configuration Registers are initialized with default values. If the VIO LDOs are enabled as per device definition in REG#25 (see table 26 Part Number Specific Configuration), the VIO LDOs are turned on at this moment.

After the configuration registers are initialized, the device logic is enabled. The two on-chip I2C/I3C slave interfaces are active after both SCL and SDA lines are detected as high.



**Figure25.Device Initialization**

## 12.3 Device Reset

There are multiple ways to reset the I3C Hub Device. A reset will clear all configuration registers to their default value.

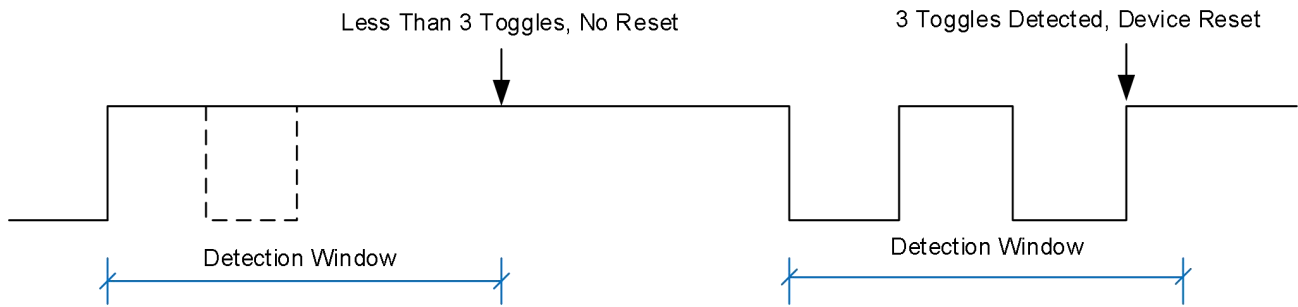
The I3C Hub device reset itself after power up with a built in Power-On Reset circuit.

Master on any Master Side Ports can write to the DEV\_CMD (Reg#124) register with a reset command to trigger a device reset.

A device reset clears all the configuration registers and repeats the initialization process. This is the same as the way device gets out of the reset state. Masters should allow for  $t_{RESET}$  for the Device Reset to finish.

If enabled by REG#19, MSEL pin toggling can be used to reset the I3C Hub device:

- If it is not already in an on-going detection window, a MSEL pin level change (i.e. low to high, or high to low) starts a MSEL toggle detection window, along with the master side port selection.
- If three or more following triggering edges are detected in the detection window (defined by  $t_{MSEL\_DET}$ ), the I3C Hub will reset itself.
- If there are less than three following toggles detected in the detection window, the toggle detector will wait for a new toggle to start a new detection window.



## 12.4 I3C Interface Reset

Master on any of the Master Side Ports writes to the DEV\_CMD register with a reset command to trigger an interface reset on the I3C slave attached to that specific master interfacing port. The master waits for at least 100us to allow for the interface reset to finish before new commands issue to the interface.

An interface reset restore the default value to all the interface-dedicated configuration registers. After the interface is reset, the master on that interface re-assigns the dynamic address to that slave interface before issuing new direct I3C commands or CCCs.

## 12.5 Bus Reset

The Bus Reset feature is defined in the JEDEC DIMM Management Bus operation specification.

The JEDEC DIMM bus reset mechanism uses a timeout mechanism on MSCL as shown below to force an I2C/I3C bus reset. JEDEC SPD bus supposes all devices on the bus are reset simultaneously, including the slaves that are hooked to the Slave Side Ports. The Bus reset operation works the same way regardless of whether the device is operating in I2C or I3C Basic mode.

When the Bus Reset Condition met, the On-Chip Slave of the PJ71MXX takes the following actions:

1. The associated bus's On-Chip Slave Interface and any pending commands or transactions are cleared
2. Dynamic address assignment is cleared; the Slave works in I2C mode
3. The device treats the bus reset as a STOP operation.

If the Master Side Port that received the Bus Reset is the selected Master Side Port, the Bus Reset condition will be propagated to the Slave Side Ports by definition. Bus Reset does not change the Master Side Port selection. The Bus Reset does not reset the other I3C slave that is attached to the other master interfacing port.

The Bus Reset Procedure is defined as SCL low detected for an period that equals to a timeout setting. The timeout setting is programmable via REG#122. To guarantee that the device resets the I2C bus or JEDEC SPD / I3C Basic bus, the MSCL clock input Low time has to be greater than or equal to  $t_{\text{TIMEOUT}}(\text{Max})$ .

The PJ71MXX Hub device does not reset the I2C bus or JEDEC SPD/ I3C Basic bus if the MSCL clock input Low time is less than  $t_{\text{TIMEOUT}}(\text{Min})$ .

If the MSCL clock input Low time is between  $t_{\text{TIMEOUT}}(\text{Min})$  and  $t_{\text{TIMEOUT}}(\text{Max})$ , the PJ71MXX Hub device does not guarantee that it will reset the I2C bus or I3C Basic bus.

## 12.6 Bus Clear

The PJ71MXX device supports the following bus clear scheme in I2C mode.

There may be abnormal circumstances when the host and I3C hub device's on chip slave interface are out of sync in a read operation. For these type of events, the MSDA data line may appear as stuck low as the device is expecting to receive more clock pulses from the host. The host may optionally clear the device that looks stuck low on the MSDA data line by sending 9 MSCL pulses without driving the MSDA data line. The I3C Hub device will release the MSDA within these nine clocks. Host may send additional 0 to 9 MSCL pulses, then send a STOP to terminate the transaction. The PJ71MXX device will return to the Idle state. The device is ready for normal new transaction with Start condition.

## 12.7 Bus Context

The I3C Hub Device operates in two bus contexts in the application:

- JEDEC Module Management Bus Protocols (As defined in SPD/Hub, TS, RCD and PMIC devices' JEDEC Spec)
- Generic I3C Basic 1.0 (As defined in MIPI I3C Basic Specification 1.0)

The PJ71MXX supports a superset of features that covers both contexts and offers a set of configuration registers to set up the device to operate the devices under a specific bus context. The device powers up with a default setting.

The user sets PJ71MXX Device' s Bus Context by a SETBUSCON CCC, or manually program the configuration registers as per the operation context in fine detail.

### 12.7.1 Context Setting Command

When a SETBUSCON CCC is received, it checks the context byte against the JEDEC context defining byte, REG#29 and a byte pattern of "00x1\_000x" as the code for I3C Basic 1.0/1.1.

- If it matches the JEDEC context defining byte, the device assumes that the master wants to set the bus context as "JEDEC". A set of configuration registers are set accordingly. Check the register definition for further details.
- If it matches the I3C Basic 1.0/1.1 Code, some of the configuration registers are updated accordingly as per the I3C Basic 1.0 setting. Check the register definition section for further details.
- Other context code is ignored.

Please note that the SETBUSCON only set or reset the registers once, upon received the CCC. Those bits are still programmable by I2C/I3C private register access after the SETBUSCON CCC is received.

### 12.7.2 JEDEC SPD/Module Management Bus Operation

#### 12.7.2.1 On-Chip I3C Slave

##### 12.7.2.1.1 JEDEC Bus Context CCCs

JEDEC SPD Bus has defined three JEDEC specific CCCs:

- SETHID
- DEVCTRL
- DEVCAP

The PJ71MXX I3C Hub Device on-chip slaves support DEVCTRL and DEVCAP CCC. The Hub Network supports all JEDEC CCCs.

The PJ71MXX I3C Hub Device supports a list of JEDEC supported I3C Basic CCCs with PEC, plus JEDEC SPD Bus specific payload bits.

The CCC framing of JEDEC defined CCCs are found in section JEDEC SPD Bus CCCs.

##### 12.7.2.1.2 Dynamic Address Assignment

JEDEC Bus operates in homogeneous mode – all devices have a static address and are assigned a dynamic address from its static address with a SETAASA broadcast CCC. JEDEC SPD Bus refers to the mode that a dynamic address is assigned as "I3C Mode" .

##### 12.7.2.1.3 Packet Error Checking Feature

The PJ71MXX I3C Hub Device supports all JEDEC defined PEC Features. Only I3C mode (dynamic address assigned) supports packet error checking.

The I3C Hub device implements an 8-bit Packet Error Code (PEC) which is appended at the end of all transactions if PEC is enabled through DEVCTRL CCC, or by directly writing '1' to the PEC\_Fun\_Enable bit. The PEC is a CRC-

8 value calculated on all the messages bytes except for START, REPEATED START, STOP conditions, T-bits, ACK, NACK, and IBI header (7'h7E followed by W=0) bits.

The polynomial for CRC-8 calculations is:

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

The seed value for the PEC function is zero.

When Host calculates PEC for the I3C Hub device, it includes slave address bits followed by R/W bit.

#### **12.7.2.1.4 Parity Disable Feature**

JEDEC SPD Bus allows parity check to be disabled.

The PJ71MXX I3C Hub Device supports-the I3C\_Parity\_Check\_Disable Feature.

#### **12.7.2.1.5 Bus Reset**

Bus Reset is defined by JEDEC SPD Bus. PJ71MXX I3C Hub Device supports Bus Reset.

#### **12.7.2.2 Hub Network**

In the JEDEC SPD bus, the I3C transactions passing through the Hub Network have with the following assumptions:

- No IBI Address arbitration optimization – the first byte of the I3C frame is Open Drain. Users can explicitly configure the address arbitration optimization configuration register, or use SETBUSCON CCC to set the context.
- No mixing of device protocols – All devices are in either I2C mode or I3C basic mode. Mode transition with SETAASA broadcast CCC.
- PEC of the CCC and private R/W transactions are supported and enabled throughout the hierarchy with DEVCTRL CCC. The on-chip slave interface and the Hub Control logic transitions the PEC enable status at the same time.

Refer to JEDEC SPD Bus Operation for further information.

### **12.7.3 Generic I3C Basic 1.0 Bus Operation**

#### **12.7.3.1 On-Chip I3C Slave**

PEC and Bus-Reset are not defined in MIPI I3C Basic 1.0. A SETBUSCON with MIPI 1.0 Context clears these settings. However, the user can elect to enable these settings. Since the PEC enabled mode CCC framing is not defined in I3C basic for most CCCs, the host should avoid using the PEC setting or use these CCCs when the PEC is enabled.

Parity checking will be enabled. It could be still disabled via the configuration register but discouraged for I3C basic compatibility.

The JEDEC CCCs – SETHID is ignored. The DEVCTRL is supported but it is assumed that the Host will not issue this CCC.

The dynamic address assignment is done by SETAASA, SETDASA, SETNEWDA, or ENTDAAs CCC.

### 12.7.3.2 Hub Network

All I3C transactions passing through the Hub Network have the following assumptions:

- No HDR operation throughout the I3C Hierarchy.
- I2C, I3C SDR, IBI, and CCCs listed in section Bus Context are supported
- IBI Address arbitration optimization is supported if enabled by configuration register (see REG#21). If the feature is enabled, it is the responsibility of the Host to make sure that the first bit of the Dynamic Address of the I3C slave devices that reside behind the Slave Side Ports are always “0”;
- Mixing of I2c/I3C device protocols is supported and enabled by configuration register, if masters configure the Slave Dynamic Address Flag table in REG#64. However, the master only operates at the full speed when the I2C devices are I3C compatible – i.e. the bus is a mixed fast bus. If I2C slaves cannot tolerate I3C speed transactions, the master has to lower the speed. Another option is to group the low-speed I2C devices with a dedicated port and disconnect that port when a high-speed operation is needed for I3C slaves.
- IBI may or may not have the MDB byte. The Hub Control logic properly handles the hand-off by always hold the ACK bit on behalf of the Slave device during the SCL high phase at the first ACK location.

#### 12.7.3.2.1 Slave Dynamic Address Assignment Flag table

The Dynamic Address Assignment Flag table in Slave Dynamic Address Assignment Flag Registers tracks if a slave address was assigned with dynamic address.

- Each entry of the DA Flag table indicates if a slave address maps to a static address or dynamic address. It is the responsibility of the I3C master to configure this table along with (or right after) the device discovery/enumeration process. The PJ71MXX device does not snoop the Dynamic Address Assignment process on the bus.
- For instance, if dynamic address “0100\_011” was assigned to a slave, the byte #4 (0100), bit 3 (011) of the Slave Dynamic Address Assignment Flag Registers should be set.
- Transaction address matching a static address is treated as an I2C transaction;
- Transaction address matching a dynamic address is treated as an I3C transaction;
- Transaction address matching CCC frames is treated as CCC

Refer to Section Generic I3C Basic 1.0 Bus Operation for further information.

### 12.7.4 Comparison of the two Bus Context's Operation

The On-Chip Slave and Hub Network of Slave Side Port are independent operating function blocks. They may support different features. For instance, the on-chip Slave ignores some CCCs but the Hub Network may support CCCs passing through the Hub Network.

**12.7.4.1 I3C Bus Related Features**
**Table27.Feature Comparison of JEDEC SPD Bus and Generic I3C Bus Context**

Feature	JEDEC I3C SPD Bus			Generic I3C Basic 1.0 Bus		
	On-chip Slave in I2C Mode	On-Chip Slave in I3C Mode	On-Chip Hub Network	On-Chip Slave DA not assigned	On-Chip Slave DA assigned	On-Chip Hub Network
PEC checking (see PEC_Fun_Enable)	Not supported	Defined	Transparent <sup>[1]</sup>	Not Defined	Not Defined	Transparent
Bus Reset (see Bus_Reset_Enable)	Supported	Defined	Transparent	Not Defined	Not Defined	Transparent
Address Arbitration Optimization (see HubNetwork_IBI_Arbit_Opt_En)	N/A	N/A	Not defined	N/A	N/A	Supported
Mixed I2C/I3C Devices (see HubNetwork_Frame_Protocol)	N/A	N/A	Not defined	N/A	N/A	Supported
Slave Side Ports OD only Mode	No	No	N/A	Yes	No	N/A
Slave Side Ports connection to Hub Network	Yes	Yes	Yes			
ENEC CCC	JEDEC illegal — I3C Hub Device will respond as I3C generic device	Supported	Transparent	Direct CCC Ignored; Broadcast CCC supported	Supported	Transparent
DISEC CCC	JEDEC illegal — I3C Hub Device will respond as I3C generic device	Supported	Transparent	Direct CCC Ignored; Broadcast CCC supported	Supported	Transparent
RSTDAA CCC	Ignored	Supported (Broadcast CCC only)	Transparent	Ignored	Supported (both Direct and Broadcast)	Transparent
SETDASA CCC	JEDEC Not Defined / I3C Hub Device Supported	JEDEC Not Defined / Ignored	Transparent	Supported	Ignored	Transparent
SETAASA CCC	Supported	Ignored	Transparent	Supported	Ignored	Transparent
GETSTATUS CCC	Ignored	Supported	Transparent	Ignored	Supported	Transparent
DEVCAP CCC	JEDEC illegal — Device will handle as I3C generic device	Supported	Transparent	Ignored	Supported	Transparent
SETHID CCC	Ignored	Ignored	Transparent	Ignored	Ignored	Transparent

**Table 27. Feature Comparison of JEDEC SPD Bus and Generic I3C Bus Context (Cont.)**

Feature	JEDEC I3C SPD Bus			Generic I3C Basic 1.0 Bus		
	On-chip Slave in I2C Mode	On-Chip Slave in I3C Mode	On-Chip Hub Network	On-Chip Slave DA not assigned	On-Chip Slave DA assigned	On-Chip Hub Network
DEVCTRL CCC	Supported	Supported	supported	I3C Not Defined / I3CHub device supported	I3C Not Defined / I3CHub device supported	supported
ENTDAA CCC	JEDEC Not Defined / I3CHub Device Supported	N/A	Supported	Supported	N/A	Supported
SETMWL CCC	Ignored	Ignored	Transparent	Ignored	Ignored	Transparent
SETMRL CCC	Ignored	Ignored	Transparent	Ignored	Ignored	Transparent
SETNEWDA CCC	Ignored	JEDEC Not Defined / I3C Hub Device Supported	Transparent	Ignored	Supported	Transparent
GETPID CCC	Ignored	JEDEC Not Defined / 3C Hub Device Supported	Transparent	Ignored	Supported	Transparent
GETBCR CCC	Ignored	JEDEC Not Defined / 3C Hub Device Supported	Transparent	Ignored	Supported	Transparent
GETDCR CCC	Ignored	JEDEC Not Defined / 3C Hub Device Supported	Transparent	Ignored	Supported	Transparent
ENTAS[0-3] CCC	Ignored	Ignored	Transparent	Ignored	Ignored	Transparent
I3C Basic 1.1 CCC						
RSTACT CCC	Ignored	Ignored	Transparent	Ignored	Ignored	Transparent
SETBUSCON CCC	Not Defined / 3C Hub supported	JEDEC Not Defined / 3C Hub Device Supported	Transparent	Supported	Supported	Transparent

(1). Transparent means that the Hub Network does not differentiate it from regular transaction and transparently propagate it to the downstream device ports.

### 12.7.4.2 IO Related Features

**Table28.IO Operation Comparison between JEDEC SPD Bus and Generic I3C Bus**

Feature	JEDEC Context Behavior			Generic I3C Context Behavior		
	Master Side Port	Slave Side Port	On-Chip Hub Network	Master Side Port	Slave Side Port	On-Chip Hub Network
Master / Slave Side Port IO Default in OD only Mode	Yes	No	N/A	Yes	Yes	N/A
Slave Side Port default connection to Hub Network (see SSPorts_HubNetwork_Connection)	N/A	Yes	N/A	N/A	No	N/A
Slave Side Port Pull up Resistor (see SSPorts_Pullup_EN)	N/A	Yes	N/A	N/A	Yes	N/A
Slave Side Port Default Enabled	N/A	Yes	N/A	N/A	No	N/A

### 12.7.4.3 Error Handling Features

**Table29.Error Handling Feature Comparison Between JEDEC SPD Bus and Generic I3C Bus**

Feature	JEDEC Context	Generic I3C Context
	On-Chip Slave	On-Chip Slave
S0	Not defined	Defined
S1	Defined	Defined
S2	Defined	Defined
S3	Not defined	Defined
S4	Not defined	Defined
S5	Not defined	Defined
PEC	Defined	Not defined

## 12.8 Two Masters Operation

### 12.8.1 I3C Network Management

When two masters are connected to the I3C Hub device, only one master connects to the Hub Network at any time. One master initializes the I3C Network, including assigning dynamic addresses, configures the Hub devices, etc.

The other master can take over to the Hub Network in a few different ways:

- The other master gets all the I3C network information, including the dynamic addresses, Hub configurations, and device configurations-via application-specific protocols or settings. For instance, a setting is communicated via the master-master communication channel provided by the PJ71MXX, or other communication channels.
- The other master resets the bus and re-initializes the I3C Network. In this model, each master has its own settings. No configuration exchange is needed, but it will be unable to do dynamic switching between the two masters.

### 12.8.2 Master Selection

Master Side Port selection is made by MSEL pin (higher priority) or Control Registers. Refer to section Multiplexer Selection for details of the control scheme.

The two masters may negotiate before switching.

- In the pin-strapping mode, the Hub Network will select M0 regardless the MSEL Pin state and Register Selection settings with REG#56. Refers to REG#19 definition for details.

### 12.8.3 Shared Device Level Register Access

Two masters can access on-chip Bus Agent and GPIO at the same time. If Register Device\_Reg\_Always\_Selected (see REG#19) is set, only the selected master can control the them (i.e. have write access to the registers).

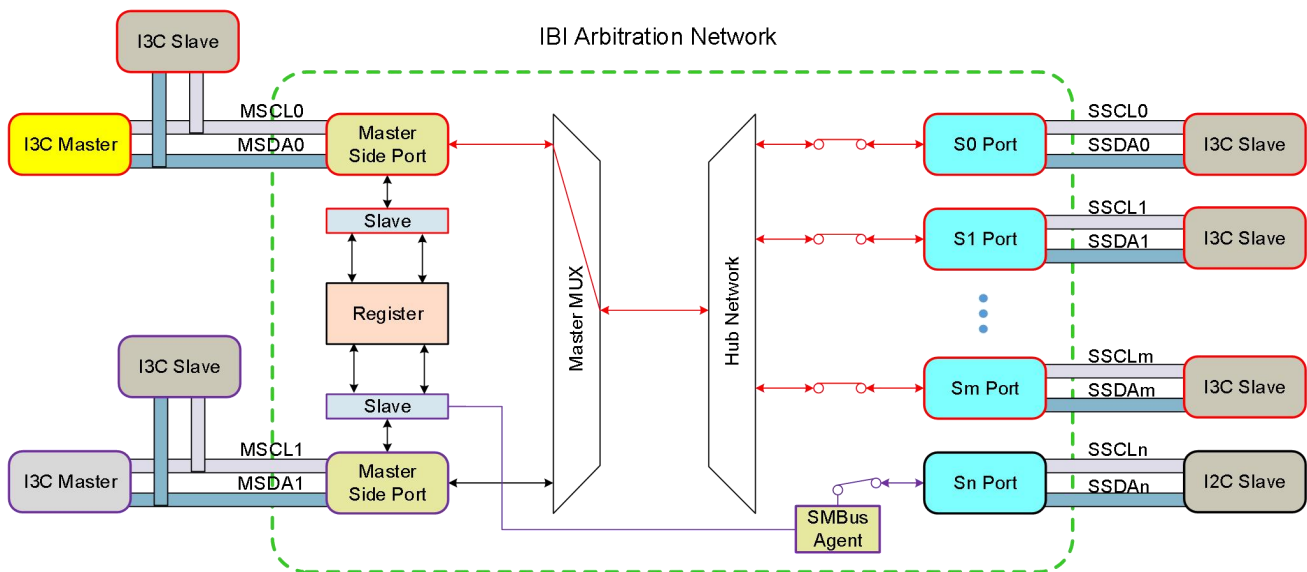
### 12.8.4 Dedicated Port Level Register Access

Each master has its own register space and master side port dedicated registers.

One masters can read the other port's register map via the page 4 reference window. M0 has the capability to write to the M1 registers (see Paged Register Map).

## 12.9 IBI Generation and Arbitration

IBI is supported in the I3C bus hierarchy visible to each of the masters that hooked to any master interfacing ports. The selected master that hooked to the selected master interfacing port will see all slaves that hooked to the slave interfacing port in the transparently connected hub network.



**Figure 26. Chip-Level IBI Domain and Generation**

When an IBI occurs, all the devices and virtual devices (the on-chip I3C slave that attached to the master interfacing port) arbitrates together as per I3C IBI arbitration rules.

- For the selected master, the arbitration participant includes any I3C Slaves that hooked to the bus connecting to the master interfacing port, the attached on-chip slave, and any downstream I3C slaves that hooked to the Slave Interfacing Port that connected to the Hub Network.
- For the non-selected master, the arbitration participant includes I3C Slaves that hooked to the bus connecting to the master interfacing port, the attached on-chip slave, and any on-chip Bus Agent configured to have IBI routed to that master interfacing port.
- See the example below. The Red framed blocks are selected master's arbitration scope, and the Purple framed blocks are the non-selected master's IBI arbitration scope. The two IBI arbitration scopes operate independently. In this example, one SMBus Agent generates IBI via M1 port as a function of the PJ71MXX I3C Hub device.

## 12.10 Master-Master Communication

The PJ71MXX device offers multiple ways to help the masters to communicate through this device.

### 12.10.1 IBI Message Buffer

The PJ71MXX offers an IBI buffer for each on-chip slave that attached to a Master Side Port. That buffer allows one master to send up to 4 bytes with each triggered inter-master IBI. The IBI sequence is as follows, for example, assume M0 sending data to M1:

- Master 0 program the 4 bytes data into the Master 0 to Master 1 message buffer registers (see Out\_Msg\_Byte0)
- Master 0 program the IBI request bit (see Master\_Msg\_Req) to trigger an IBI in M1 bus.
- M1 bus arbitrates the IBI with other slaves, if any, that are hooked up on the M1 port. If the arbitration loses out, it tries the next time with a START or bus idle condition.
- Master 1 received the IBI. It allows the four data payload bytes directly sent through the IBI, or terminates the IBI transaction and launch a read transaction to read the IBI data.
- M1 analyzes the data bytes, optionally fetching more bytes from the shared buffers, and follow any negotiation protocol that pre-defined between M1 with M0. It then clears the Master\_Msg\_Pending\_Flag. It may send some data bytes back to M0 via the similar IBI channel. Alternatively, it may fetch additional data from the directly mapped smaller scratch-board registers, or a paged packet buffer, which is up to 256 bytes.

Please note that IBI is an I3C feature and it is not available if the masters operate in I2C mode, or if the IBI is disabled. If IBI is not available, the user can use the scratch-board buffers or paged buffers to transfer data in an application-specific protocol with register read/write, along with other channels outside the PJ71MXX. The other master may use polling, or other application-specific protocol to utilize the IBI buffers.

### 12.10.2 Shared Scratch Board Buffers

The PJ71MXX offers 4 bytes directly mapped shared register (see Scratch\_Byte0) buffer between M0 and M1. The shared buffer can be Read and Write by any master. The two masters can use this buffer to define application-specific communication protocols.

The shared buffer and IBI message buffer can be combined and provide up to 8 data bytes data transfer from one master to the other.

The PJ71MXX offers 256 bytes paged buffer (see Master Side Port Scratch Board buffer 0, 128 bytes, R/W) between M0 and M1. The paged buffer could be Read and Write by any master. The two masters could use this buffer to define application-specific communication protocols.

## 12.11 Transparent Hub Network Operation

When the Hub Network is connecting to one or more Slave Side Ports, the connected Slave Side Ports operate in a transparent bridging mode.

To configure the Slave Side Port to the Hub Network, the following configuration need to be set accordingly:

- Slave Side Ports need to be enabled
- Slave Side Ports' pull-up resistor is configured
- GPIO and Bus Agent are not enabled
- Port Connection to the Hub Network is set

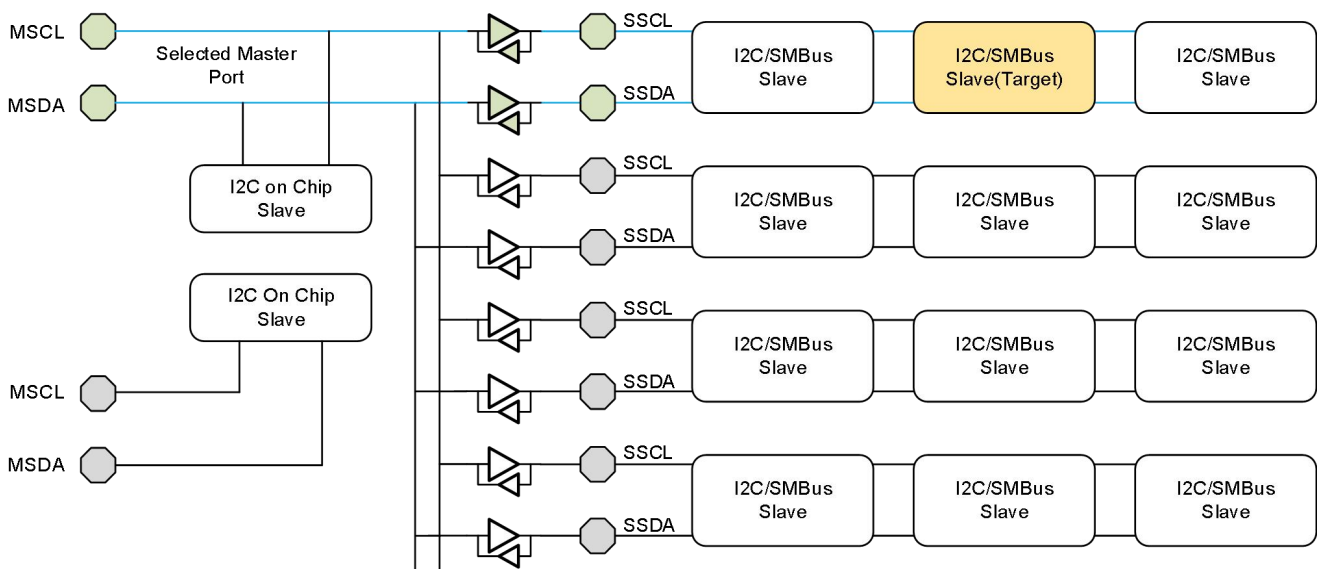
In transparently connected Hub Network, SCL stretching is only supported when the IO in both the Master Side Port and Slave Side Port are in the OD Only Signaling mode.

**Table30.Transparent Hub Network Operation**

Master Side Port		Slave Side Port		Transparent Bridging Support	Slave Side Port SCL driving mode
OD_Only bit	IO Voltage	OD_Only bit	IO Voltage		
0	1.0V-1.8V	0	1.0V-1.8V	Yes	Push/Pull
0	1.0V-1.8V	1	1.0V-3.3	No (Slave Port will not be connected if connection is configured)	Output Disabled if Hub connection is configured
1	1.0V-3.3V	0	1.0V-1.8V	Yes	Push/Pull
1	1.0V-3.3V	1	1.0V-3.3V	Yes	Open-Drain / SCL stretching supported

### 12.11.1 I2C/SMBus Open-Drain Only Operation

In this mode, the I3C Hub device operates when the selected master is configured as “OD Only” (see MPort\_OD\_Only). The Slave Side Port should be in “OD Only” configuration as well. The Selected Master SCL and SDA bus connects to Slave Side Ports' SDA channel with a bi-directional Open-Drain bridge. The Hub device supports SCL stretching from Slave hooked to Slave Side Ports.


**Figure27.Hub Network Transparent Operation in OD Only Mode**

### 12.11.2 I2C/I3C Compatible Mode Operation

#### 12.11.2.1 I2C Operation

In this mode, all Slave Side SSCL ports that connected to the Hub Network works in re-driver mode and re-drives signal received from selected Master Side M\_SCL port input. SSCL stretching is not supported.

The selected Master Side M\_SDA port is bridged to all connected Slave Side SSDA ports with bidirectional open drain circuit.

The SSCL re-driver and SSDA pull-up are powered by port associated VIOS supply.

The selected Master Side Port's MPort\_OD\_Only register is set after power up. It can be cleared manually, or if configured, get cleared if received any I3C transactions with “7E” address header. The input port input signaling level is defined by the associated VIOM supply.

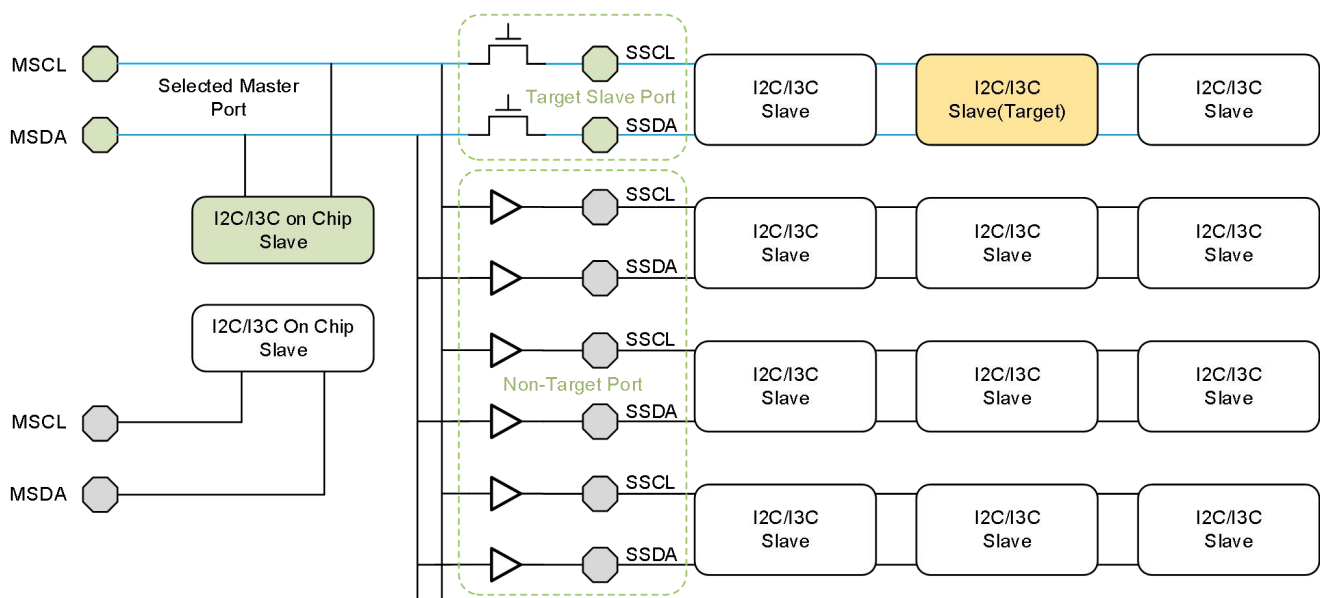
### 12.11.2.2 I3C Operation

In I3C Operation, the transaction's target Slave Side Port is connected to the selected Master Side Port with a high-performance analog switch, or low latency re-driver. The non-target ports are re-driven with the sensed signal at the selected Master Side Port. All the transparently connected ports see the exactly same traffic. The "HubNetwork\_PP\_Analog\_Switch\_En" register (see REG#21) defines if a redriver analog switch will be placed between the selected Master Side Port and target Slave Side Port.

#### 12.11.2.2.1 Dynamic Analog Switching Mode

In this mode, the target slave interfacing port connects to the selected master interfacing port with a high-performance analog switch. The enabling between the analog switch and the re-driving circuit is automatic and only when a read transaction is acknowledged. To use this feature, the master and the slave has to use the same or very close IO supply voltage.

The advantage of this mode is very low pass-through latency, if the Master has a strong driver.

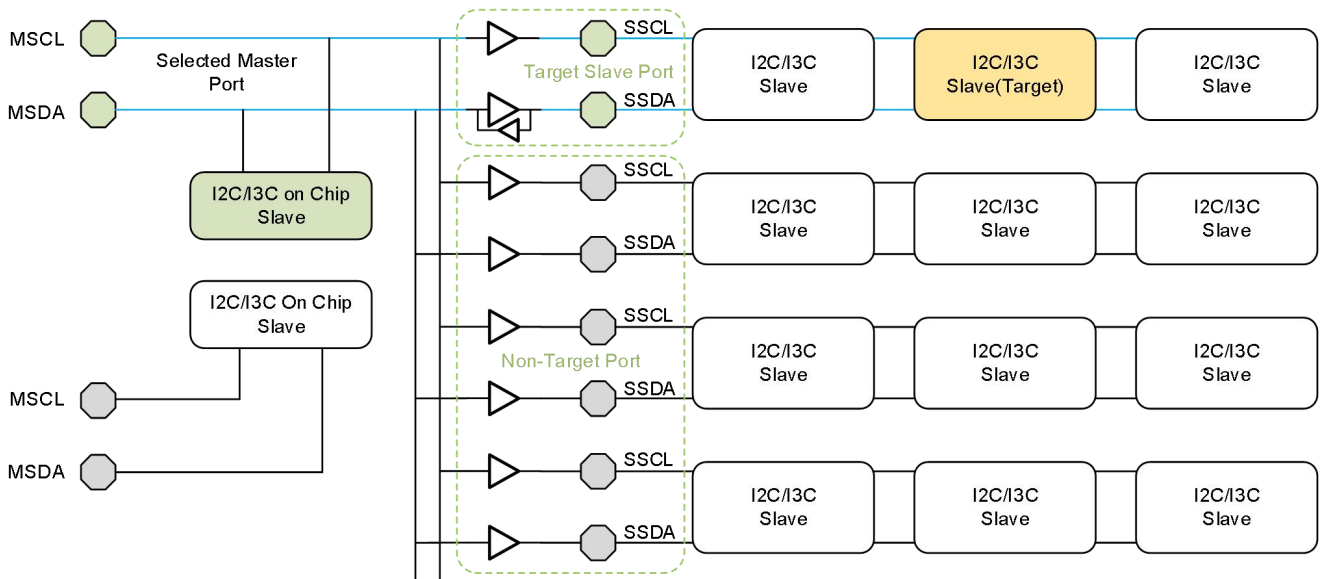


**Figure28.Hub Network Transparent Operation in I3C Auto-analog-switch Mode**

#### 12.11.2.2.2 Re-Driving Mode

In this mode, the target slave interfacing port connects to the selected master interfacing port with high-performance re-drivers. The master and the slave can have different IO supply voltage, which is good for level shifting.

The advantage of this mode is an isolated network, well-controlled drive strength and allow for different master and slave operating voltages.



**Figure 29. Transparent Hub Network in Re-Driving Mode**

### 12.11.2.3 Dynamic Address Assignment Procedure

#### 12.11.2.3.1 JEDEC Module Management Bus Operation

It is assumed that all devices on the JEDEC Module Management bus are both I2C and I3C capable, and all devices are uniquely identifiable by a known, predefined static address.

The assumed Startup and Dynamic Address Assignment procedure will be as follows:

- The host issues zero or several I2C-mode commands.
- The host issues SETHID CCC to enable DIMM's SPD/Hub device's HID feature.
- The host issues zero or several I2C-mode commands.
- The host will issue SETAASA CCC to bring all devices into I3C mode, with the dynamic address assigned the same as the static address.
- The host will issue I3C transactions with all devices got dynamic addresses (in I3C mode).
- The host may issue RSTDAA CCC to bring all devices into I2C operations.
- The host may issue Bus Reset to bring all devices into I2C operations.

#### 12.11.2.3.2 Generic I3C Basic 1.0 Bus Operation

The masters that initially control the I3C bus elects to use a SETAASA, SETDASA, ENTDAAs to assign a dynamic address to the slaves in the I3C network. There are two ways the Hub Network understands the framing of the transactions passing through the hierarchy:

- A: All down-stream devices the same as the Hub's on-chip slave settings
  - The master sets all I3C Hub devices' HubNetwork\_Frame\_Protocol Register in I2C mode first.
  - The master assigns the dynamic address using SETAASA, or SETDASA, or ENTDAAs to assign the dynamic address to all slaves at the same time, or one by one. Before all devices have a dynamic address assigned, no private Read/Write transactions will go through the I3C hierarchy.
  - After all the devices get a dynamic address (including all the I3C Hub devices), the master can issue I3C framed private Read and Write Transactions.
- B: Any down-stream devices have unique settings
  - The master sets all I3C Hub devices' HubNetwork\_Frame\_Protocol Register in I2C mode first.
  - The master assigns a dynamic address to a slave at its discretion. Before accessing that dynamic address,

the master is required to set the dynamic address assigned flag for that address in the dynamic address assignment table (see REG#64) for all the I3C Hub devices. It is suggested that the master update the table immediately after the dynamic address assignment.

- It is suggested that the master assign the dynamic address of all the I3C hub devices in the network first.

### **12.11.3 Mixed I2C/I3C Slave support**

#### **12.11.3.1 Push-Pull Operation after Repeat Start**

The I3C protocol specifies that after the address arbitration header, the bits following Repeat Start will be transferred in a Push/Pull manner.

In generic I3C Basic 1.0 Bus operation, the hub network will operate in push/pull mode for this address byte after a Repeat Start, even if it's an I2C transaction. This is because it may be a push/pull transaction targeting an I3C slave at full speed. The ACK/NAK bit is Open Drain. After the address byte is received and Acknowledged, the hub control logic determines the transaction protocol and follow up accordingly.

#### **12.11.3.2 Transaction Type Determination**

To support the I2C / I3C mixed bus, the Hub control logic needs to understand if an address is assigned as a dynamic address or remain as a static address. To allow this, the master needs to program the dynamic address assignment table REG#64 after the dynamic address is assigned. The PJ71MXX does not snoop on the CCCs that assigns the dynamic addresses.

If the I2C/I3C mixed bus support defined in Register HubNetwork\_Frame\_Protocol is enabled, The Hub control logic parses the slave address and determine if an I2C or I3C transaction frame is expected on the bus.

If the I2C/I3C mixed bus support is not enabled, the Hub control logic in the PJ71MXX assumes all the devices remain in either I2C mode (all static address) or all in I3C mode (all assigned dynamic address).

If the Dynamic Address Assignment table is not used for address lookup, it could be used as a generic data buffer for master-master communication.

### **12.11.4 Bus Partition**

The PJ71MXX device supports port temporary port disconnection in the transparently connected Hub Network. That means devices behind the disconnected port does not receive the transaction that the connected device sees. This brings two advantages:

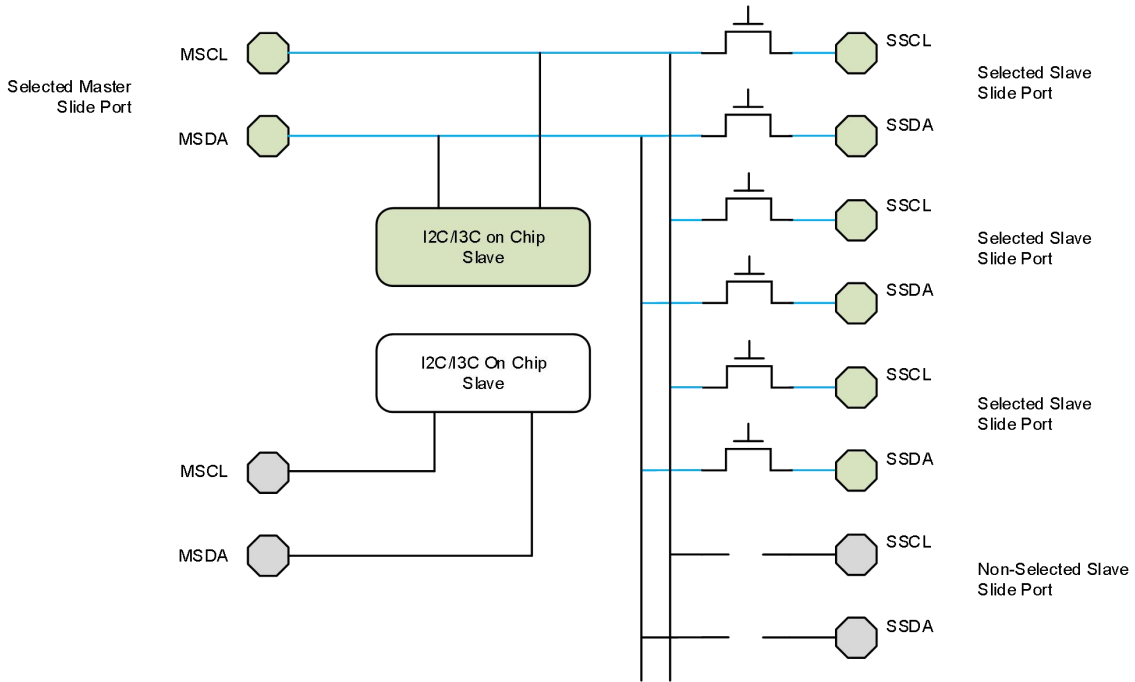
- The master can put slow I2C devices behind an I2C only port and disconnect that port when communicating with I3C traffic. The I3C transactions could be transmitted at the full speed without concerning the compatibility of the I2C devices.
- The master can expand the manageable devices and slave address space by accessing the partitioned ports in a time-division manner.

### **12.11.5 Fixed Analog Switch Mode**

The Hub Network can be set to a fixed analog switch mode (see HubNetwork\_Fixed\_Analog\_Switch\_Mode). In this mode, all connected ports can be connected with a fixed-on analog switch. User uses the Slave Side Port Connection Register (REG#81) to control the ports that connect to the Hub Network. Once the connection is configured, the analog switch is enabled regardless the traffic mode or frame format, i.e. there is no difference between Open Drain operation and a push/pull operation.

Multiple Slave Side Ports can be connected to the Hub Network via the Analog Switch at the same time. However, in this case the Fixed Analog Switch Mode will be unable to isolate the electrical load of connected ports.

This mode is different from the dynamic analog switch mode operation. In the dynamic analog switch mode, only the target port is in the analog switch mode during the push/pull operation, where other connected ports connect to the re-drivers; the open-drain operation is not in analog switch mode so load isolation is achieved.



**Figure 30. Hub Network in Fixed Analog Switch Mode**

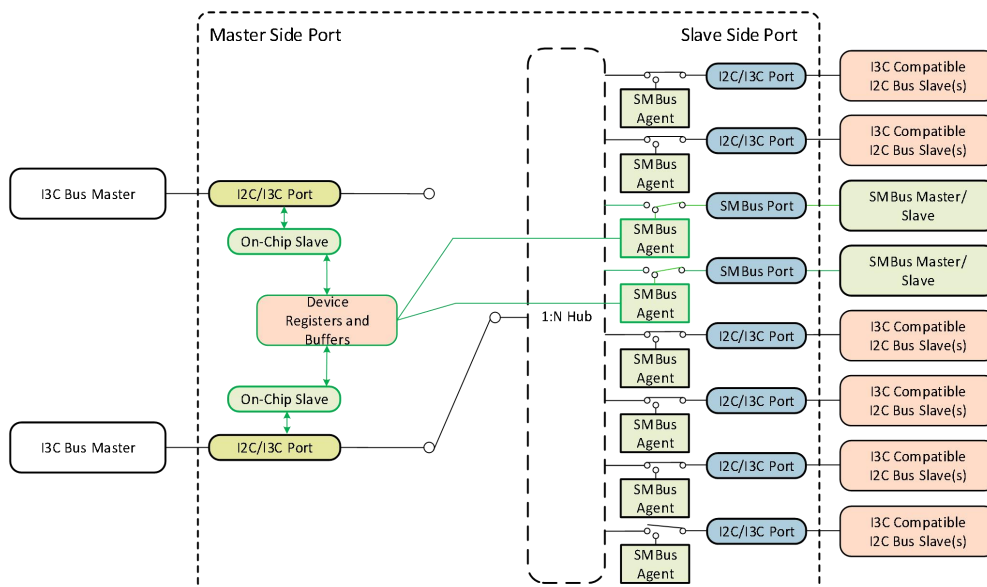
### 12.12 Bus Agent Operation

The PJ71MXX offers a dedicated SMBus Agent that is attached to each Slave Interface Port. In contrast to the transparent Hub Network bridging operation, the Bus Agent works differently. The Agent independently works with downstream devices according to the configuration and request from the masters and operates at the transaction level.

The SMBus Agent is used to interface with the legacy SMBus device or I2C devices as a protocol and speed adapter. The SMBus Agent initiates and receive transactions but doesn't interpret them.

#### 12.12.1 Topology

In Agenting mode, the Slave Interfacing Port that the Agent is enabled is disconnected from the Hub Network. The Agents interfaces to the downstream devices like a bus master plus bus slave. The transaction that is transmitted or received is stored in the Bus Agent buffers.

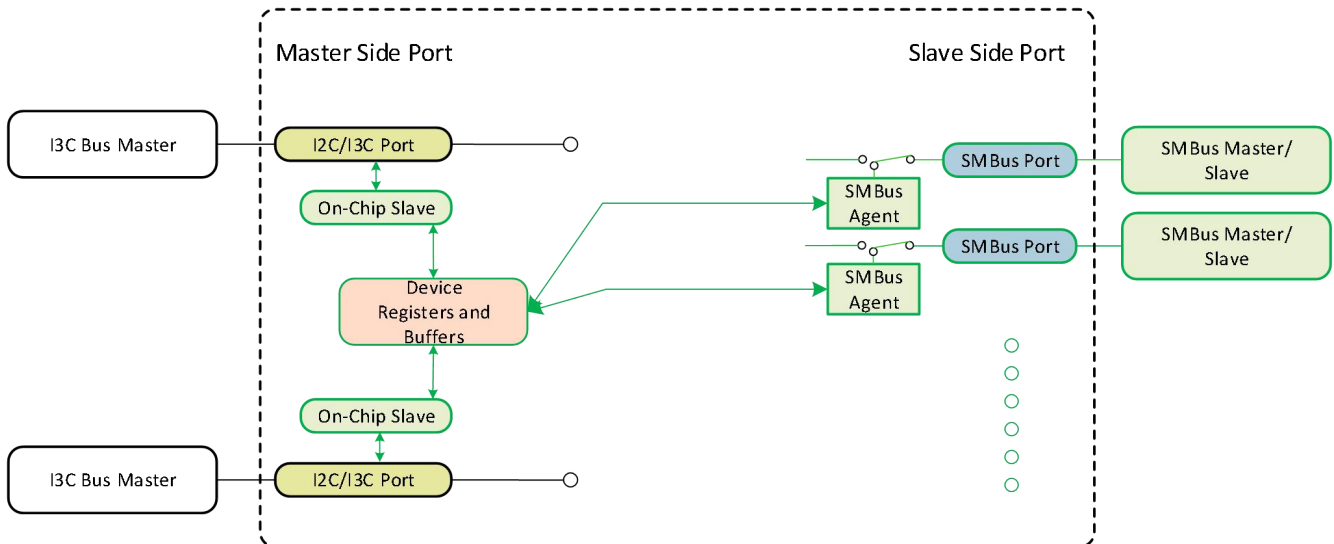


**Figure 31. SMBus Agent Operation**

### 12.12.2 Bus Agent Communication to the I3C Masters

The SMBus Agent communicates with the bus masters via configuration registers and transaction buffers, which are mapped to the address space of the Master Interface Port attached on-chip slave. The Agent is accessed from any master interfacing port.

The related registers are SMBus Agent Status and Configuration Registers, SSPorts\_Agent\_IBI\_EN, and SMBus Agent Data Buffer.



**Figure32.SMBus Agent Access Model**

### 12.12.3 Bus Agent Communication to the Downstream Devices

The SMBus master agent and slave agent operate independently as if two devices hooked up at the same bus at the Slave Interfacing Port.

- The master agent supports SCL stretching bit by bit. Therefore, a master initiated transaction may lose arbitration. The same transaction that the master agent transmitted may be captured by the slave agent.
- The slave agent receives all bus write transactions. The local transactions, which are determined by comparing the device address of the transaction, are discarded.

### 12.12.4 Signal Integrity

By partitioning the I3C network with multiple ports, topology is improved, the load is reduced. The system has better Signal Integrity (SI). The system designer should treat all PCB traces as transmission lines since the master and slave devices transfer data in a fast transition edge.

### 12.12.5 IO Drive Strength Control in Re-Driving Mode

The PJ71MXX offers programmable IO driver impedance (see REG#20). All IO drivers in the PJ71MXX are calibrated for 1.0V Rail and room temperature Push/Pull Operation in the factory testing.

If IO rail voltage is not 1.0V or the operating temperature differs significantly, the driver strength might deviate from the programmed value. PJ71MXX devices offer a driver calibration command (see DEV\_CMD) after all the IO voltage is supplied in the application. This command will re-calibrate the IO driver strength with the IO voltage and ambient temperature. User should wait 2ms to allow the calibration finish.

### 12.12.6 Dynamic Analog Switch Mode

The analog switch path is automatically turned on, if configured. It allows a direct connection between the master interfacing ports and slave interfacing ports.

Since the analog switches introduce a passing resistance and pin capacitance, when using the analog switch mode, it is suggested that the system designer limits the number of cascading stages of the Hub stages in analog switch mode and run SI simulations. When the Hub is configured to work in analog switch mode, the transmitting device (either master or slave) in the I3C Push/Pull operation should be tuned to the strong drive strength setting.

## 12.13 Hot Join

### 12.13.1 Hot Join on Slave Side Ports

#### 12.13.1.1 Slave Side Port connects to the Hub Network

To support I3C Hot Join propagation from downstream port to upstream port, the Slave Interfacing Ports should be connected to the Hub Network and operate in transparent bridging mode.

The hub network supports IBI, Hot Join, and ENTDA CCC framing. Therefore, Hot Join of Slaves on Slave Side Ports is supported in the Generic I3C Basic 1.0 Bus operation.

#### 12.13.1.2 Slave Side Port connects to the Bus Agent

If configured, the SMBus Agent can receive the Hot Join message and report it through the IBI channel.

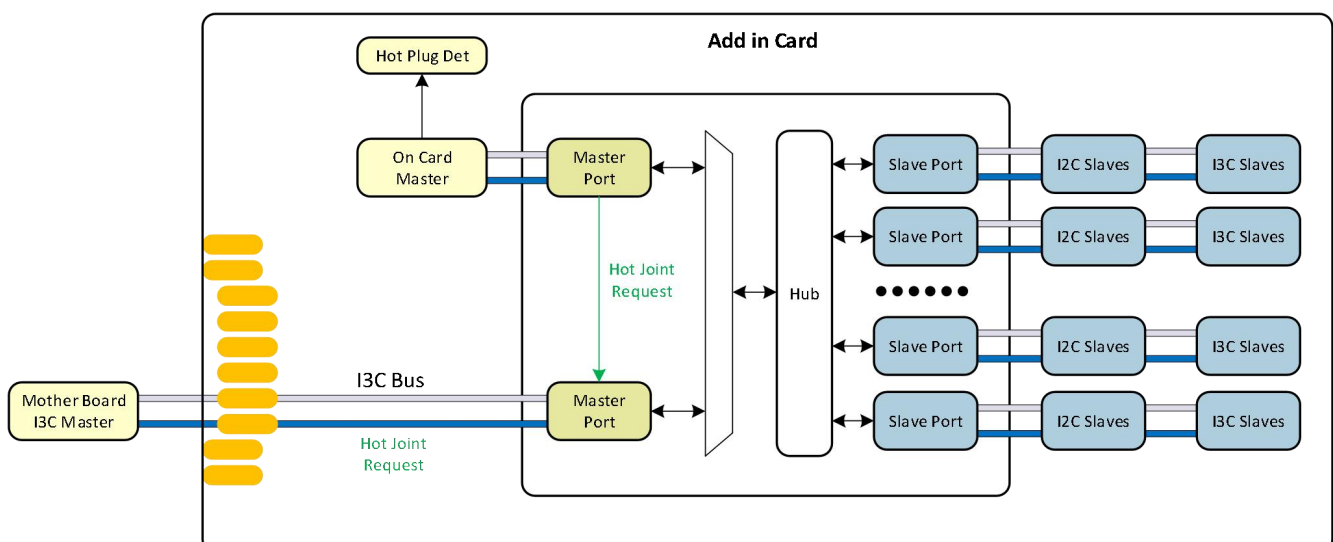
### 12.13.2 Hot Join on Master Side Ports

The PJ71MXX supports hot-join on Master Side Ports.

- Assume that M1 is the port to hot join an I3C bus.
- The master at M0 configures the device configuration registers and enables the Hot Join feature.
  - M0 may choose to configure the M1 as the selected Master Side Port to connect to the Hub Network to Slave Side Ports.
- The master at M0 sends Hot Join request (see Other\_Master\_HJ\_Req) to M1.
- M1 sends Hot Join to the Master Side Port.
- Master at M1 goes through the Hot Join flow.

Please note that a Master Side Port does not initiate the Hot Join procedure by itself after power-up. It is triggered by a request from another Master Side Port. The intended application example is add-in cards where an on-card master triggers the Hot Join to enumerate the card to the motherboard I3C master after plugged in.

The example below is of the PJ71MXX as the edge hub device on an add-in card that connects to the motherboard. The on-card master and motherboard master share access to the slaves. The on-card master initiates a hot-join procedure after the card is plugged in and properly booted up. The on-card master does not need to be an I3C capable device.



**Figure33.Hot Join on an Add-in Card**

## 12.14 Transaction Rules

### 12.14.1 Device Configuration Transaction Rules

Any transactions that targeting the on-chip slave interface and programming the configuration registers, i.e. the registers with bit fields with the “P” attribute, should be ended a “STOP” rather than a “Repeat Start” symbol.

When programming to the following registers, if REG#18 is not with “0x00”, the I2C/I3C transaction of the register write access should finish at the address of those registers with a STOP:

REG#16, REG#17, REG#18, REG#19, REG#21,REG#23,REG#24, REG#30,REG#81, REG#124,

The exception is when REG#18=00, the master could use a write-through transaction to configure those registers in a long burst. If REG18 is included in this burst, the value of REG#18 should be set to “0x00”. REG#18 needs to be set after other configuration registers were properly programmed.

To avoid side effects, a Slave Side Port configuration change from Bus Agent mode or GPIO mode to transparent mode needs to respect the following constraints: The IO should be either in logic high / SMBus idle state, or in IO disabled state, prior to connect to the transparent Hub Network.

### 12.14.2 CCC Transaction Rules

A CCC should end with a STOP if the following transaction is an on-chip slave register access (the I3C “Private Read/Write”).

A Broadcast CCC should end with a STOP if the following transaction is a Direct CCC to this device. SETBUSCON CCC should end with a STOP.

## 12.15 Device Addressing and Identification

### 12.15.1 I2C/SMBus Static Addresses

The Hub device powers up with a default I2C static address, as defined in 9.2.3.1 Slave Addressing. This address can be updated by programming REG#126.

If multiple Hub devices connect to the same I2C bus and Master Side Port 1 is not used, the Master Side Port 1 pins could be strapped to differentiate the I2C static addresses. Refer to REG#19 for details.

If the number of Hub devices exceeds the addresses that the pin strapping can support, the master can control the strapping pins dynamically.

The Hub device does not support SMBus ARP protocol.

### 12.15.2 I3C Dynamic Addresses

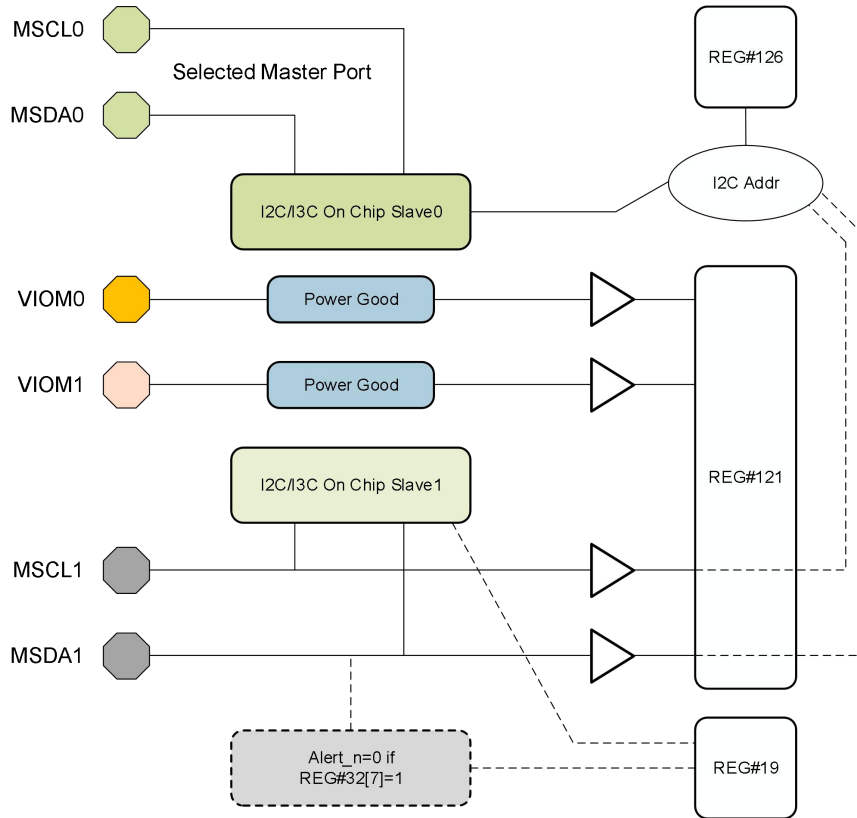
I3C ENTDA protocol assigns unique I3C dynamic address to each I3C device on the bus. In the case that multiple Hub devices reside on the same I3C bus, the master need to assign unique addresses to each device and identify each I3C dynamic address on the board. The I3C Hub device supports ENTDA with an unique PID for each device, so multiple I3C Hub device always get assigned with a different Dynamic Address.

The application software may need to uniquely identify each I3C Hub devices on the same bus. This can be done by the following ways:

- Using a downstream port: The master can sense a downstream port’s logic level that selectively pulled down or pulled up in GPIO mode. This can be done once after each after power cycle. If the downstream port that used in the identification are used during the operation, the identification selection signals need to be turned off into high impedance state after the identification phase.
- Using the unused Maser Side Port 1: M1, If not enabled, can be used for device identification via REG#121.
  - The MSDA1, MSCL1 pins of the not-enabled port can be pulled high or low to identify the device.
  - The VIOM1 supply can be used to identify the device by setting the VIO1M target level and check the VIO1M

power good status.

- Using the MSEL pin: The I3C master use a dedicated MSEL pin to control each Hub device. Device can be identified by reading the MSEL pin state of each device.



**Figure34.Using Master Side Port 1 for Device Identification**

## 12.16 SMBus Only Operation

SMBus only operation through the transparent Hub Network is supported when both the Master Side Port and Slave Side Port is in OD\_Only Mode. SCL clock stretch is support through the Hub Network in this mode.

If the Maser Side Port 1 is not used, with configuration in REG#19, the MSDA1 pin changes to “Alert\_n” pin and outputs the IBI Pending status bit in an inverted logic in an Open Drain manner. MSEL pin should be fixed to select Master Side Port 0.

The Alert\_n pin’s function is illustrated as below:

### IBI\_pending Status (REG#32[7]) Alert\_n pin State

- 0 = Float (need external pull-up resistor, pull-up level up to 3.3V)
- 1 = 0

Below is a table that illustrates different REG#19 settings and associated M1 pins usage. Refer to Figure 34 for M1 pin mapping details.

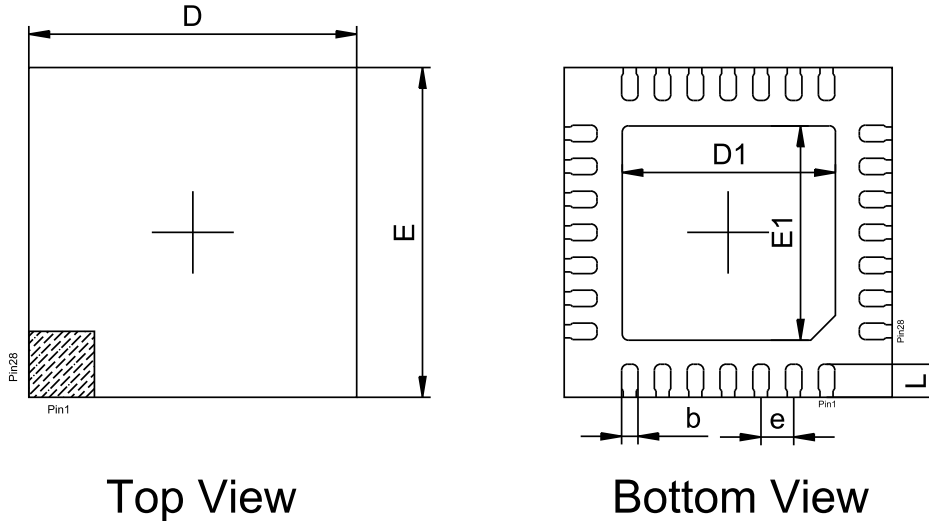
**Table31.M1 Pins Usage**

REG#19[6:4]	MSCL1 Pin	MSDA1 Pin	On-Chip Slave @M1	On-Chip Slave @M0
<b>000</b>	As MSCL1 of M1	As MSDA1 of M1	Working	Working
<b>001</b>	For M0 identification	For M0 identification	Disabled	Working
<b>011</b>	For M0 identification	As Alert_n	Disabled	Working
<b>111</b>	As M0_I2C_Addr[0]	As Alert_n	Disabled	Working / I2C Address[0] Changed
<b>101</b>	As M0_I2C_Addr[0]	As M0_I2C_Addr[1]	Disabled	Working / I2C Address[1:0] Changed

# 13 Package Outline Dimensions and Recommend Land Pattern Layout

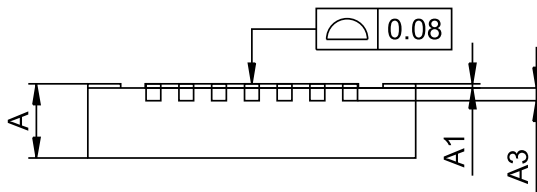
## 13.1 Package Outline Dimensions

QFN4x4-28Unit (mm)



Top View

Bottom View



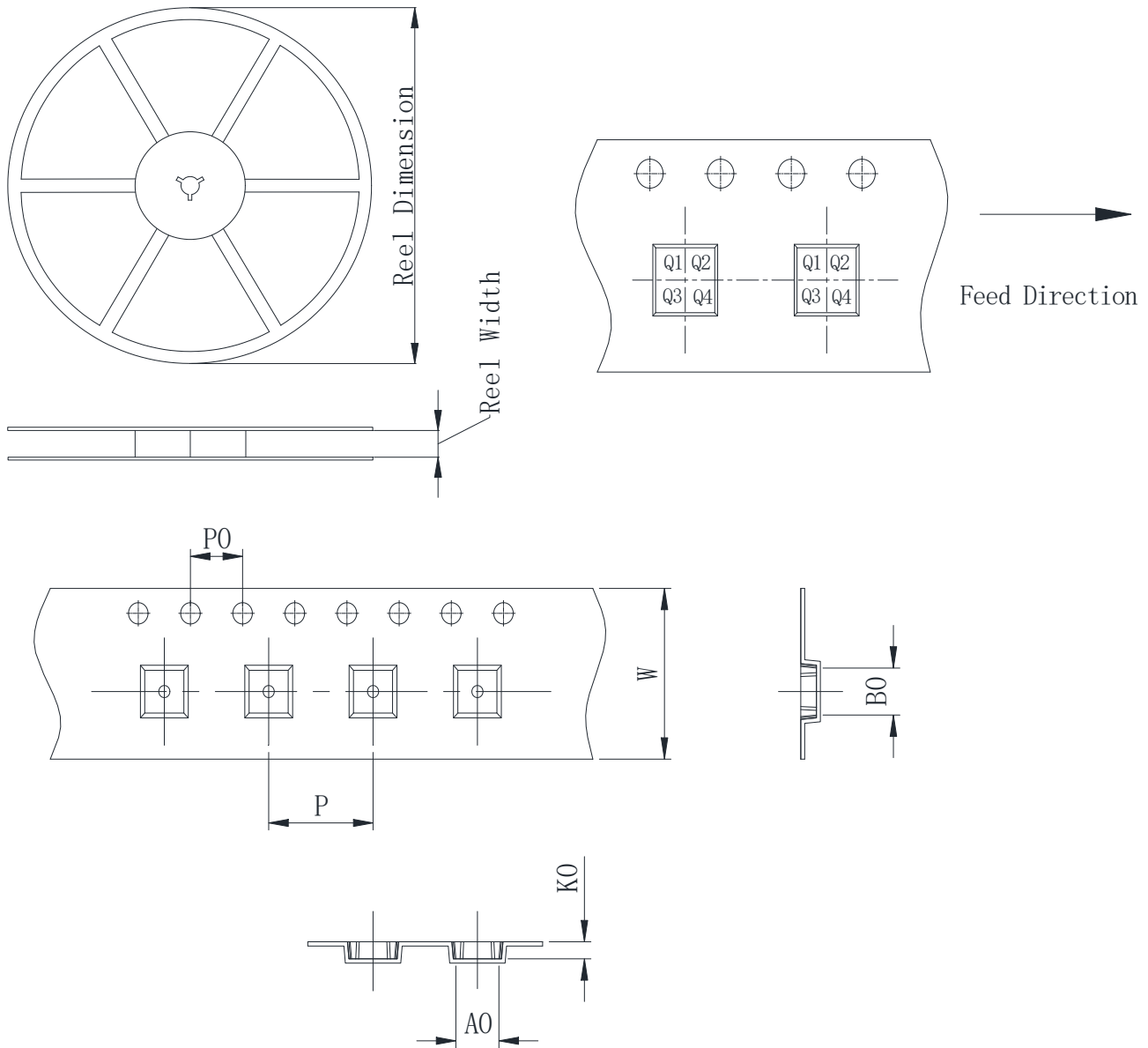
Side View

Symbol	Dimensions in millimeters		Dimensions in Inches	
	Min.	Max.	Min.	Max.
A	0.800	0.900	0.031	0.035
A1	0.000	0.050	0.000	0.002
A3	0.203REF		0.008REF	
D	3.900	4.100	0.154	0.161
E	3.900	4.100	0.154	0.161
D1	2.500	2.700	0.098	0.106
E1	2.500	2.700	0.098	0.106
b	0.150	0.250	0.006	0.010
e	0.400BSC		0.016BSC	
L	0.350	0.450	0.014	0.018

(1). Pin1 shape on backside is not limited to bevel, it can be a notch or arch.



## 14 Packing information



Package type	Reel size	Reel dimension (±3.0mm)	Reel width (±1.0mm)	A0 (±0.1mm)	B0 (±0.1mm)	K0 (±0.1mm)	P (±0.1mm)	P0 (±0.1mm)	W (±0.3mm)	Pin1
QFN4x4-28	13'	330	12.4	4.3	4.3	1.1	8.0	4.0	12.0	Q2

## 15 Version History

Version	Date	Changes
Rev.1.0	2026-02-26	Initial release

## Disclaimer

MetaWells Co., Ltd. and all other MetaWells Co., Ltd. trademarks are the exclusive property of MetaWells Co., Ltd.. Any mention of additional trademarks or registered trademarks in this document is made in acknowledgment of their respective owners.

Without the consent of MetaWells Co., Ltd. , any modification to any part or subsection of the company's product specifications and datasheets is strictly prohibited in any form or manner. MetaWells Co., Ltd. reserves the rights to make changes of the content herein the document follow PCN procedure. Please refer to our website for latest document. It is recommended that customers obtain the latest version of relevant information and verify its timeliness and completeness before placing an order. All products are sold in accordance with the terms and conditions provided in the sales contract at the time of order confirmation, which includes warranty scope, intellectual property rights, and liability limitations.

During the sales period, MetaWells Co., Ltd. guarantees that the performance of the products will be in accordance with the company's standard warranty. The company utilizes testing and other quality control techniques to ensure the necessity to maintain this warranty. Except for government-mandated requirements, there is no need for special testing of measurement tables or other instruments.

Customers must acknowledge that our products are not intended or manufactured for use in life support systems or other high-risk activities or environments. The utilization of our products in such applications may lead to severe consequences, including personal injury, loss of life, property damage, or environmental harm. Our products are explicitly not warranted for deployment in high-risk activities, and we shall bear no responsibility towards customers or third parties for such usage.

MetaWells Co., Ltd. commits to providing technical support, assistance, advice, and information, in line with our current capabilities, including guidance related to the design, development, or debugging of purchased boards or other applications. However, we hereby disclaim all warranties concerning technical support, merchantability, or fitness for a particular purpose. We do not guarantee the error-free operation of boards or applications in conjunction with our support services, and we shall not be held liable for any consequences arising from the utilization of our support services.

**Copyright© MetaWells Co., Ltd.**

**Website : [www.metawells.com](http://www.metawells.com)**